

# **General Mathematics UDF Guide**

Version 1.5

LazyMath

## **Acknowledgements**

A huge thank you to everyone who helped test the UDFs 😊

## Contents

|                             |          |
|-----------------------------|----------|
| <b>Data Analysis .....</b>  | <b>5</b> |
| Summary Statistics.....     | 5        |
| Dot Plot .....              | 6        |
| Histogram.....              | 7        |
| Frequency Table .....       | 8        |
| Inverse Normal .....        | 9        |
| Normal Bound.....           | 10       |
| Normal Solve .....          | 11       |
| Line Solve .....            | 12       |
| Linear Regression .....     | 13       |
| Linear Transformations..... | 14       |
| Residuals.....              | 15       |
| Mean Smoothing.....         | 16       |
| Median Smoothing.....       | 17       |
| Seasonal Data.....          | 18       |
| Significant Figures .....   | 21       |

|                                               |           |
|-----------------------------------------------|-----------|
| <b>Recursion and Financial Modelling.....</b> | <b>22</b> |
| Recurrence Relation.....                      | 22        |
| Recurrence Relation Step.....                 | 23        |
| Amortisation Table .....                      | 24        |
| Number of Payments.....                       | 25        |
| Finance Solver .....                          | 26        |
| <b>Matrices .....</b>                         | <b>27</b> |
| Communication Matrix.....                     | 27        |
| Analyse Dominance .....                       | 29        |
| Dominance Solve .....                         | 30        |
| Square Root Matrix.....                       | 31        |
| Outcomes .....                                | 32        |
| Leslie Matrix.....                            | 34        |
| Leslie Table.....                             | 35        |
| Steady State.....                             | 36        |
| Transition Table .....                        | 37        |
| Inverse Transition .....                      | 39        |

|                                                |           |
|------------------------------------------------|-----------|
| <b>Networks and Decision Mathematics .....</b> | <b>41</b> |
| Introduction.....                              | 41        |
| Vertex Labelled Graphs (VLG) .....             | 41        |
| Edge Labelled Graphs (ELG) .....               | 44        |
| Dijkstra’s Algorithm .....                     | 47        |
| Prim’s Algorithm .....                         | 48        |
| Hamiltonian .....                              | 49        |
| Traverse Vertices .....                        | 50        |
| Traverse Edges.....                            | 51        |
| Float Time.....                                | 52        |
| Dummy Activity .....                           | 53        |
| Project Crashing.....                          | 54        |
| Flow .....                                     | 59        |
| Path Flow .....                                | 61        |
| Allocate.....                                  | 62        |
| Hungarian Algorithm.....                       | 63        |

## Data Analysis

### Summary Statistics

Determines the quartiles, fences, mean, and standard deviation of the input data.

#### Syntax

`sum_stats(Data)`

Where, *Data* represents a list containing the data to be analysed.

#### Example

The number of points a pro gamer scores on Flappy Bird over 10 games is shown in the table below.

| Game  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  |
|-------|----|----|----|----|----|----|----|----|----|-----|
| Score | 12 | 47 | 58 | 73 | 20 | 31 | 10 | 22 | 17 | 250 |

Determine the quartiles, fences, and outliers (if any).

```
score_data {12.,47.,58.,73.,20.,31.,10.,22.,17.,250.}
summary_stats(score_data)
```

Total = 10

►Data Summary:

|                |            |
|----------------|------------|
| "Minimum"      | 10.        |
| "Q1"           | 17.        |
| "Q2"           | 26.5       |
| "Q3"           | 58.        |
| "Maximum"      | 250.       |
| "IQR"          | 41.        |
| "Lower Fence"  | -44.5      |
| "Upper Fence"  | 119.5      |
| "Range"        | 240.       |
| "Mean"         | 54.        |
| "Standard Dev" | 71.972     |
| "Skew"         | "Positive" |

Warning: Skew may be inaccurate

►Possible outliers:

{ 250. }

Sorted data saved as data.summary\_stats

Done

**Warning:** Skew may be inaccurate

Contact  
lazymath2024@gmail.com

## Dot Plot

Determines the summary statistics of an input dot plot.

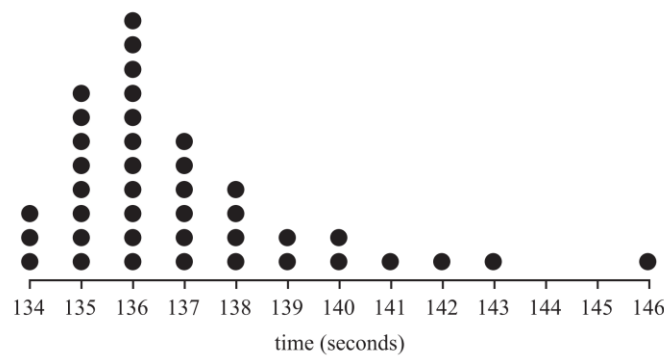
### Syntax

`dot_plot(Data)`

Where, *Data* represents a matrix with the *x*-values in the top row, and the *y*-values in the bottom row.

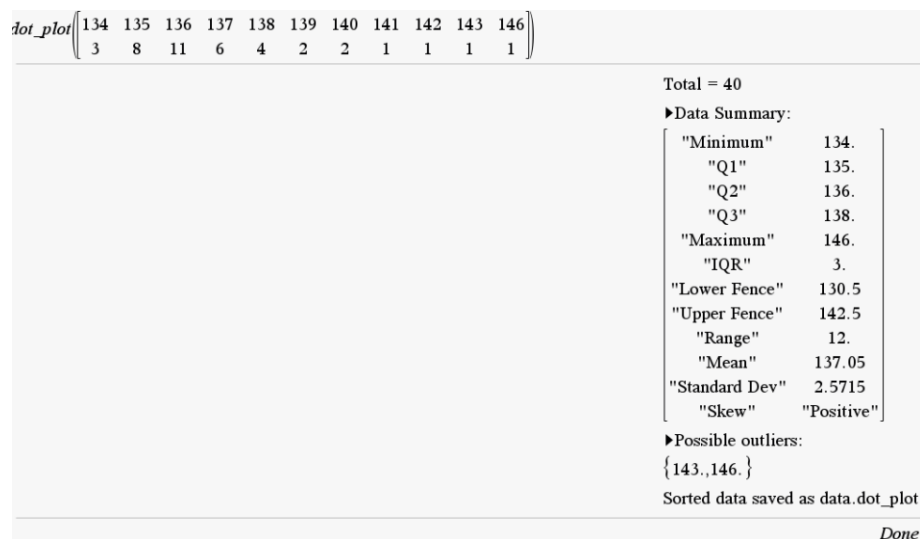
### Example

The dot plot shows the times, in seconds, of 40 runners in the qualifying heats of their 800 m club championship.



Source: VCAA 2023 General Mathematics Written Examination 1 Question 1

Determine the median and skew of the data.



**Warning:** Skew may be inaccurate.

Contact

lazymath2024@gmail.com

## Histogram

Determines the summary statistics of an input histogram.

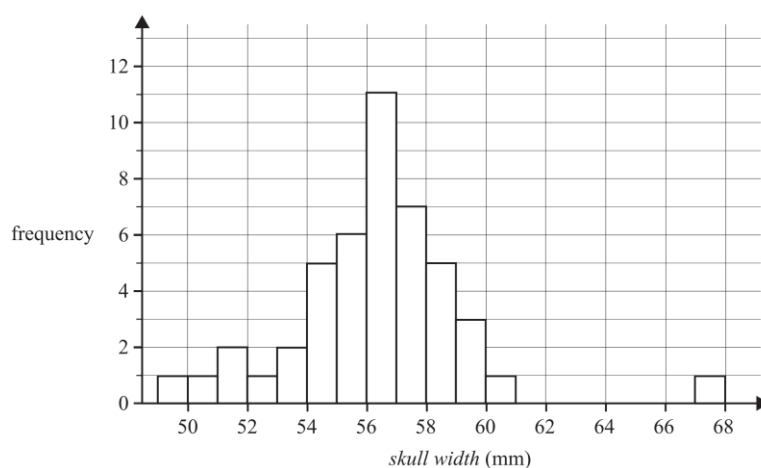
### Syntax

*histogram(Data)*

Where, *Data* represents a matrix with the *x*-values in the top row, and the *y*-values in the bottom row.

### Example

The histogram below displays the distribution of *skull width*, in millimeters, for 46 female possums.



Source: VCAA 2022 Further Mathematics Written Examination 1 Question 1

```
histogram([49 50 51 52 53 54 55 56 57 58 59 60 67],1)
```

Total = 46

►Data Summary:

|               |         |
|---------------|---------|
| "Minimum"     | "49-50" |
| "Q1"          | "54-55" |
| "Q2"          | "56-57" |
| "Q3"          | "57-58" |
| "Maximum"     | "67-68" |
| "IQR"         | "2-4"   |
| "Lower Fence" | "48-52" |
| "Upper Fence" | "60-64" |
| "Range"       | "17-19" |

►Approximate values:

|                |            |
|----------------|------------|
| "Mean"         | 56.326     |
| "Standard Dev" | 2.9235     |
| "Skew"         | "Negative" |

Warning: Skew may be inaccurate

►Possible outliers:

{ "49-50", "67-68" }

Sample data saved as data.histogram

Done

**Warning:** Skew may be inaccurate.

Contact

lazymath2024@gmail.com

## Frequency Table

Determines the frequency table of the input data list.

### Syntax

*freq\_table(Data, Minimum, Bin Size)*

Where,

*Data* represents a list containing the data

*Minimum* represents the starting point of the frequency table

*Bin Size* represents the size of each bin in the frequency table

### Example

Determine the frequency table of the following data.

{35, 48, 45, 43, 38.2, 50, 39.8, 40.7, 40, 50, 35.4, 38.8, 40.2, 45, 45, 40, 43.3, 53.1, 35.6, 44.1, 34.8}

Start your table from 30 and use a bin size of 5.

*freq\_table({35,48,45,43,38.2,50,39.8,40.7,40,50,35.4,38.8,40.2,45,45,40,43.3,53.1,35.6,44.1,34.8},30,5)*

►Frequency Table:

| "Interval " | "Frequency" | "Percentage" |
|-------------|-------------|--------------|
| "30-<35"    | 1.          | 4.7619       |
| "35-<40"    | 6.          | 28.571       |
| "40-<45"    | 7.          | 33.333       |
| "45-<50"    | 4.          | 19.048       |
| "50-<55"    | 3.          | 14.286       |
| "Total "    | 21.         | 100.         |

Done

## Inverse Normal

Uses the 68-95-99.7% rule alongside the given mean and standard deviation to determine the values for which  $Pr(X > x) = \%p$  and  $Pr(X < x) = \%p$ .

### Syntax

*norm\_inverse(Mean, Standard Deviation, Percentage Probability)*

Where,

*Mean* represents the mean of the normal distribution

*Standard Deviation* represents the standard deviation of the normal distribution

*Percentage Probability* represents the percentage probability of being less than or greater than a value

### Example

The weight of dogs is normally distributed with a mean of 30 kg with a standard deviation of 3.4 kg.

Using the 68-95-99.7% rule, determine the weight which 16% of dogs are less than.

*norm\_inverse(30,3.4,16)*

► Given:

$\bar{x} = 30$  and  $s_x = 3.4$

► Answer:

16% of values are less than 26.6

16% of values are greater than 33.4

*Done*

## Normal Bound

Uses the 68-95-99.7% rule to determine the cumulative percentage probability between two bounds, that is,  $Pr(x_1 < X < x_2)$  %.

### Syntax

*norm\_bound(Mean, Standard Deviation, Lower Bound, Upper Bound)*

Where,

*Mean* represents the mean of the normal distribution

*Standard Deviation* represents the standard deviation of the normal distribution

*Lower Bound* represents the lower bound in the probability expression

*Upper Bound* represents the upper bound in the probability expression

### Example

The weight of dogs is normally distributed with a mean of 30 kg with a standard deviation of 3.4 kg.

Using the 68-95-99.7% rule, determine the percentage of dogs which weigh between 26.6 kg and 36.8 kg.

*norm\_bound(30,3.4,26.6,36.8)*

► Given:

$\bar{x} = 30$  and  $s_x = 3.4$

► Answer:

81.5% of the values are between 26.6 and 36.8

*Done*

## Normal Solve

Uses the 68-95-99.7% rule to determine the mean and standard deviation of a normal distribution, given two probabilities,  $Pr(X < x_1) = p_1\%$  and  $Pr(X > x_2) = p_2\%$ .

### Syntax

`normsolve(Lower, % Pr(Lower), Upper, %Pr(Upper))`

Where,

*Lower* represents the value,  $x_1$

*% Pr(Lower)* represents the percentage probability of  $X < x_1$ , in other words,  $p_1\%$

*Upper* represents the value,  $x_2$

*% Pr(Upper)* represents the percentage probability of  $X > x_2$ , in other words,  $p_2\%$

### Example

The mean and standard deviation for the average weight of dogs is unknown.

After conducting some measurements, scientists determined that:

- 2.5% of dogs weigh more than 36.8 kg
- 16% of dogs weigh less than 26.6 kg

Use the 68-95-99.7% rule to determine, in kilograms, the mean and standard deviation.

```
norm_solve(26.6,16,36.8,2.5)
```

►Given:

16% of values are less than 26.6

2.5% of values are greater than 36.8

►Determine the number of  $sx$  from  $\bar{x}$  using 68–95–99.7% rule:

26.6 is -1  $sx$  from  $\bar{x}$

36.8 is 2  $sx$  from  $\bar{x}$

►Determine the equations:

$$26.6 = \bar{x} - sx$$

$$36.8 = \bar{x} + 2sx$$

►Solve equations simultaneously for  $\bar{x}$  and  $sx$ :

$$sx=3.4 \text{ and } \bar{x}=30.$$

Done

## Line Solve

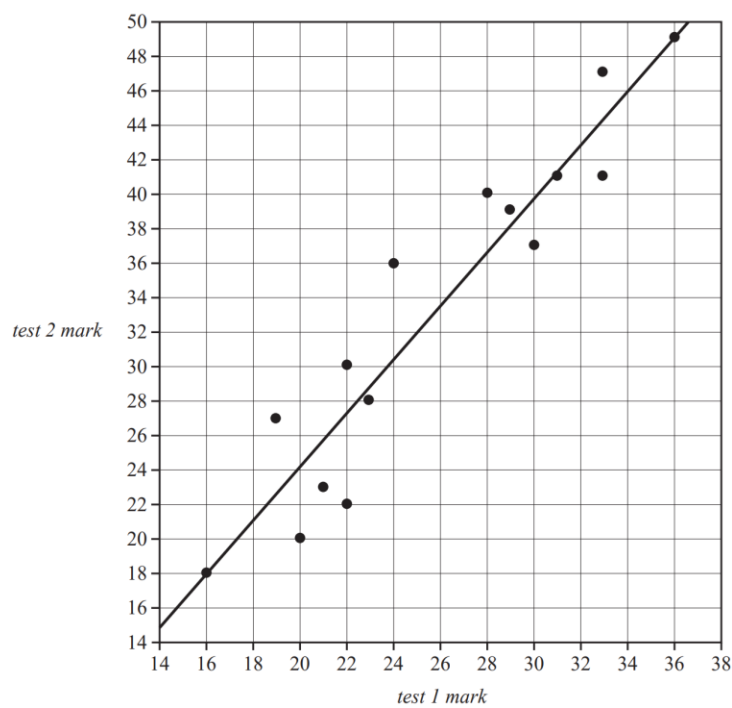
Determines the equation of the line passing through two input points.

### Syntax

$\text{lin\_solve}(x1, y1, x2, y2)$

Where,  $x1, y1, x2, y2$  represent the  $x$  and  $y$  coordinates of the two points respectively

### Example



Source: VCAA 2023 General Mathematics Examination 2 Question 7 & Question 8

Determine the equation for the least squares line.

$$\text{lin\_solve}(16, 18, 34, 46) \quad y = 1.5556 \cdot x - 6.8889$$

## Linear Regression

Determines the least squares line,  $R$ ,  $R^2$ , and association between the explanatory variable and the response variable.

### Syntax

$lin\_reg(EV, RV)$

Where,

$EV$  represents a list containing the values of the explanatory variable

$RV$  represents a list containing the values of the response variable

### Example

The amount of money a student earns from their stocks each year is shown in the table below.

| Year           | 1    | 2    | 3    | 4     | 5     | 6     | 7     | 8     |
|----------------|------|------|------|-------|-------|-------|-------|-------|
| Amount<br>(\$) | 2.50 | 6.70 | 8.90 | 10.50 | 11.70 | 16.20 | 17.50 | 19.20 |

Determine the equation for the line of best fit of the data.

$ev$   $\{1,2,3,4,5,6,7,8\}$

$rv$   $\{2.5,6.7,8.9,10.5,11.7,16.2,17.5,19.2\}$

$lin\_reg(ev,rv)$

Length = 8

►Linear Regression:

|                   |                           |
|-------------------|---------------------------|
| "Equation "       | $y=2.3095 \cdot x+1.2571$ |
| "R"               | 0.98899                   |
| "R <sup>2</sup> " | 0.9781                    |
| "Association "    | "strong positive "        |
| "Interpolation "  | "1≤x≤8"                   |

Done

Determines the least squares line and  $R^2$  of various transformations of the explanatory and response variables. These include squaring, reciprocal, and  $\log_{10}$ .

$$\textit{lin trans}(EV, RV)$$

$EV$  represents a list containing the values of the explanatory variable

$RV$  represents a list containing the values of the response variable

The amount of money a student earns from their stocks each year is shown in the table below.

| Year        | 1    | 2    | 3    | 4     | 5     | 6     | 7     | 8     |
|-------------|------|------|------|-------|-------|-------|-------|-------|
| Amount (\$) | 2.50 | 6.70 | 8.90 | 10.50 | 11.70 | 16.20 | 17.50 | 19.20 |

Determine the least squares line with  $\log_{10}(\text{amount})$  as the explanatory variable.

|      |                                                   |
|------|---------------------------------------------------|
| $rv$ | $\{2.5, 6.7, 8.9, 10.5, 11.7, 16.2, 17.5, 19.2\}$ |
|------|---------------------------------------------------|

$$lin\_trans(ev,rv)$$

►Transforms:

| "Trans "           | "R <sup>2</sup> " | "Equation "                         |
|--------------------|-------------------|-------------------------------------|
| "x"                | "0.9781"          | "y=2.3095x+1.2571"                  |
| "x <sup>2</sup> "  | "0.9104"          | "y=0.24168x <sup>2</sup> +5.4871"   |
| "y <sup>2</sup> "  | "0.9556"          | "y <sup>2</sup> =52.617x-72.426"    |
| "log(x) "          | "0.9375"          | "y=18.13log(x)+1.2126"              |
| "log(y) "          | "0.8464"          | "log(y)=0.10845x+0.51181"           |
| "x <sup>-1</sup> " | "0.7640"          | "y=-17.02x <sup>-1</sup> +17.432"   |
| "y <sup>-1</sup> " | "0.6002"          | "y <sup>-1</sup> =-0.0364x+0.29046" |

*Done*

## Residuals

Determines the least squares line fit and the differences between the true values and predicted values.

### Syntax

$residual(EV, RV)$

Where,

$EV$  represents a list containing the values of the explanatory variable

$RV$  represents a list containing the values of the response variable

### Example

The amount of money a student earns from their stocks each year is shown in the table below.

| Year           | 1    | 2    | 3    | 4     | 5     | 6     | 7     | 8     |
|----------------|------|------|------|-------|-------|-------|-------|-------|
| Amount<br>(\$) | 2.50 | 6.70 | 8.90 | 10.50 | 11.70 | 16.20 | 17.50 | 19.20 |

Determine the residual value for each year.

$ev$   $\{1.,2.,3.,4.,5.,6.,7.,8.\}$

$rv$   $\{2.5,6.7,8.9,10.5,11.7,16.2,17.5,19.2\}$

$residual(ev,rv)$

| "y"  | "ŷ"    | "Residual" |
|------|--------|------------|
| 2.5  | 3.5667 | -1.0667    |
| 6.7  | 5.8762 | 0.82381    |
| 8.9  | 8.1857 | 0.71429    |
| 10.5 | 10.495 | 0.00476    |
| 11.7 | 12.805 | -1.1048    |
| 16.2 | 15.114 | 1.0857     |
| 17.5 | 17.424 | 0.07619    |
| 19.2 | 19.733 | -0.53333   |

Done

## Mean Smoothing

Performs mean smoothing on the provided dataset and returns the result. Points which are marked with a blank string indicate they are invalid.

### Syntax

*mean\_smooth(Data, Size)*

Where,

*Data* represents a list containing the data to be mean smoothed

*Size* represents the group size which is used in smoothing

### Example

The number of sales made by a company for the first eight months of 2025 is shown in the table below.

| Month | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| Sales | 200 | 250 | 100 | 350 | 450 | 500 | 890 | 320 |

Determine the four-mean smoothed data, with centering.

*data* { 200.,250.,100.,350.,450.,500.,890.,320. }

*mean\_smooth(data,4)*

| "x" | "y"    |
|-----|--------|
| 1.  | "0"    |
| 2.  | "0"    |
| 3.  | 256.25 |
| 4.  | 318.75 |
| 5.  | 448.75 |
| 6.  | 543.75 |
| 7.  | "0"    |
| 8.  | "0"    |

Done

## Median Smoothing

Performs median smoothing on the provided dataset and returns the result. Points which are marked with a blank string indicate they are invalid.

### Syntax

$med\_smooth(Data, Size)$

Where,

*Data* represents a list containing the data to be median smoothed

*Size* represents the group size which is used in smoothing

### Example

The number of sales made by a company for the first eight months of 2025 is shown in the table below.

| Month | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| Sales | 200 | 250 | 100 | 350 | 450 | 500 | 890 | 320 |

Determine the three-median smoothed data.

$data$   $\{200., 250., 100., 350., 450., 500., 890., 320.\}$

$med\_smooth(data, 3)$

| "x" | "y"  |
|-----|------|
| 1.  | " "  |
| 2.  | 200. |
| 3.  | 250. |
| 4.  | 350. |
| 5.  | 450. |
| 6.  | 500. |
| 7.  | 500. |
| 8.  | " "  |

Done

## Seasonal Data

Determines the seasonal averages, seasonal indices, deseasonalised data, and the least square line fit of the deasonalised data. Rounding for each calculation step can be specified using the appropriate syntax.

### Syntax

#### Case 1: Exact values

*season(Data)*

Where, *Data* represents the matrix containing the data, with each row representing one cycle and each column representing one period.

### Note

In SACs and exams, you will have to round your answers at each stage. This case would be useful for checking your answers rather than obtaining the answers.

#### Case 2: Rounded values

*season({"Data", Round\_1, Round\_2, Round\_3, Round\_4})*

Where,

*"Data"* represents a string containing the name of the variable used to store the data

*Round\_1* represents the number of decimal places to round the average of each cycle to

*Round\_2* represents the number of decimal places to round the seasonal indices to

*Round\_3* represents the number of decimal places to round the average of the seasonal indices to

*Round\_4* represents the number of decimal places to round the deseasonilised data to

### Note

All of the above must be inputted in sequence as a list

### Example

The sales data for a clothing store was tracked quarterly for three years.

| Year    | 2025 |    |    |    | 2026 |    |    |    | 2027 |    |    |    |
|---------|------|----|----|----|------|----|----|----|------|----|----|----|
| Quarter | 1    | 2  | 3  | 4  | 1    | 2  | 3  | 4  | 1    | 2  | 3  | 4  |
| Sales   | 82   | 57 | 42 | 43 | 88   | 59 | 48 | 50 | 97   | 65 | 52 | 55 |

- Calculate the sales average for each quarter. Give your answer correct to two decimal places.
- Calculate the seasonal indices for each sale. Give your answer correct to three decimal places.
- Calculate the average of the seasonal indices for each sale. Give your answer correct to two decimal places.
- Deseasonalise the data. Give your answer correct to the nearest whole number.
- Determine the least squares line fit for the deseasonalised data.

### Case 1

```
data
[82. 57. 42. 43.
88. 59. 48. 50.
97. 65. 52. 55.]

season(data)

►Season UDF:
►Find averages for each cycle:
["Season" 1. 2. 3. 4. "Avg"]
"Cycle 1." 82. 57. 42. 43. 56.
"Cycle 2." 88. 59. 48. 50. 61.25
"Cycle 3." 97. 65. 52. 55. 67.25]

►Find indices and take their average:
["Season" 1. 2. 3. 4.
"Cycle 1." 1.4643 1.0179 0.75 0.76786
"Cycle 2." 1.4367 0.96327 0.78367 0.81633
"Cycle 3." 1.4424 0.96654 0.77323 0.81784
"Avg" 1.4478 0.98256 0.76897 0.80068]

►Deseasonalise the data:
["Season" 1. 2. 3. 4.
"Cycle 1." 56.638 58.012 54.619 53.705
"Cycle 2." 60.782 60.048 62.421 62.447
"Cycle 3." 66.998 66.154 67.623 68.692]

►Find LSR fit of deseasonalised data:
y=1.3066x+53.019
```

**Note:** This may **not** provide the answers the marker will be looking for since it uses the exact value at each stage rather than the rounded values.

Contact  
lazymath2024@gmail.com

## Case 2

|                                    |                                                                                                         |
|------------------------------------|---------------------------------------------------------------------------------------------------------|
| <i>data</i>                        | $\begin{bmatrix} 82. & 57. & 42. & 43. \\ 88. & 59. & 48. & 50. \\ 97. & 65. & 52. & 55. \end{bmatrix}$ |
| <i>season</i> ({ "data",2,3,2,0 }) |                                                                                                         |

►Season UDF:

►Find averages for each cycle:

|            |     |     |     |     |       |
|------------|-----|-----|-----|-----|-------|
| "Season"   | 1.  | 2.  | 3.  | 4.  | "Avg" |
| "Cycle 1." | 82. | 57. | 42. | 43. | 56.   |
| "Cycle 2." | 88. | 59. | 48. | 50. | 61.25 |
| "Cycle 3." | 97. | 65. | 52. | 55. | 67.25 |

►Find indices and take their average:

|            |       |       |       |       |
|------------|-------|-------|-------|-------|
| "Season"   | 1.    | 2.    | 3.    | 4.    |
| "Cycle 1." | 1.464 | 1.018 | 0.75  | 0.768 |
| "Cycle 2." | 1.437 | 0.963 | 0.784 | 0.816 |
| "Cycle 3." | 1.442 | 0.967 | 0.773 | 0.818 |
| "Avg"      | 1.45  | 0.98  | 0.77  | 0.8   |

►Deseasonalise the data:

|            |     |     |     |     |
|------------|-----|-----|-----|-----|
| "Season"   | 1.  | 2.  | 3.  | 4.  |
| "Cycle 1." | 57. | 58. | 55. | 54. |
| "Cycle 2." | 61. | 60. | 62. | 63. |
| "Cycle 3." | 67. | 66. | 68. | 69. |

►Find LSR fit of deseasonalised data:

$$y=1.3007x+53.212$$

## Significant Figures

Rounds an input number to a specific number of significant figures.

### Syntax

$\text{sig\_fig}(\text{Number}, SF)$

Where,

$\text{Number}$  represents the number to round

$SF$  represents the number of significant figures to round the number to

### Example

Round the number 14.520010 to five significant figures.

$\text{sig\_fig}(14.52001, 5)$  "14.520"

## Recursion and Financial Modelling

### Recurrence Relation

Determines the compound interest per annum, annuity payment, and perpetuity payment of the input recurrence relation in the form

$$V_{n+1} = aV_n + b$$

#### Syntax

*recur\_rel(R, Pmt, V<sub>0</sub>, CpY)*

*R* represents the coefficient in front of *V<sub>n</sub>*, that is *a* in the equation above

*Pmt* represents the amount being added to *V<sub>n</sub>*, that is *b* in the equation above

*V<sub>0</sub>* represents the starting balance of the loan

*CpY* represents the number of periods per annum

#### Example

Let  $E_0 = \$300\,000$  and  $E_{n+1} = 1.003E_n - 2159.41$

- Determine the compound interest per annum.
- Determine the monthly payment, in dollars, the investor would receive if they wanted the annuity to act as a perpetuity.

*Source: VCAA 2024 General Mathematics Written Examination 2 Question 7*

*recur\_rel(1.003,-2159.41,3·10<sup>5</sup>,12)*

Reducing Balance Loan

►Recurrence Relation:

$V_{n+1} = 1.003V_n - 2159.41$ ,  $V_0 = 300000$

►Interest:

$I = (1.003 - 1) \times 12 \times 100 = 3.6$

3.6% per annum, compounding monthly

►Payment: \$2159.41 per month

►Interest Only: \$900.00 per month

*Done*

## Recurrence Relation Step

Displays the lines of working required to work out  $V_n$  given a recurrence relation in the form.

$$V_{n+1} = aV_n + b$$

### Syntax

$\text{recur\_rel\_step}(R, \text{Pmt}, V_0, \text{Iter})$

$R$  represents the coefficient in front of  $V_n$ , that is  $a$  in the equation above

$\text{Pmt}$  represents the amount being added to  $V_n$ , that is  $b$  in the equation above

$V_0$  represents the starting balance of the loan

$\text{Iter}$  represents which term in the sequence we wish to determine

### Example

Let  $E_0 = \$300\,000$  and  $E_{n+1} = 1.003E_n - 2159.41$

Showing recursive calculations, determine the balance of the annuity after two months. Round your answer to the nearest cent.

Source: VCAA 2024 General Mathematics Written Examination 2 Question 7

$\text{recur\_rel\_step}(1.003, -2159.41, 3 \cdot 10^5, 2)$

► Recurrence Relation Step:

► Working:

$$V_0 = 300000$$
$$V_1 = 1.003 \times 300000 - 2159.41 = 298740.59$$
$$V_2 = 1.003 \times 298740.59 - 2159.41 = 297477.4$$

► Solution:

$$= \$297477.40$$

Done

## Amortisation Table

Generates the amortisation table based on the input payment amount, frequency of payments, interest rate, and starting balance.

### Syntax

$amor\_tbl(\%I, Pmt, V_0, CpY, Iter)$

Where,

$\%I$  represents the percentage compound interest per annum

$Pmt$  represents the payment per period

$V_0$  represents the starting balance of the loan

$CpY$  represents the number of periods per annum

$Iter$  represents the number of rows of the table you wish to generate

### Example

Arthur invests \$600 000 in an annuity that provides him with a monthly payment of \$3973.00.

Interest is calculated monthly at a rate of 0.42% per month.

Complete the first four lines of the amortisation table. Round all values to the nearest cent.

Source: VCAA 2023 General Mathematics Written Examination 2 Question 6

$amor\_tbl(0.42 \cdot 12, -3973, 6 \cdot 10^5, 12, 3)$

---

► Recurrence Relation:

$R = 1 + 5.04 / (12 \times 100) = 1.0042$

$V(n+1) = 1.0042V_n - 3973, V_0 = 600000$

► Amortisation Table:

| "No." | "Pmt"     | "I"       | "PR"      | "Bal"       |
|-------|-----------|-----------|-----------|-------------|
| 0     | "0.00"    | "0.00"    | "0.00"    | "600000.00" |
| 1     | "3973.00" | "2520.00" | "1453.00" | "598547.00" |
| 2     | "3973.00" | "2513.90" | "1459.10" | "597087.90" |
| 3     | "3973.00" | "2507.77" | "1465.23" | "595622.67" |

Reducing Balance Loan

---

*Done*

## Number of Payments

Determines the number of payments which could be made to pay off the loan and the final payment amounts.

### Syntax

$final\_pmt(\%I, Pmt, V_0, CpY)$

Where,

$\%I$  represents the percentage interest per period

$Pmt$  represents the payment per period

$V_0$  represents the starting balance of the loan

$CpY$  represents the number of periods per annum

### Example

Arthur borrowed \$30 000 to buy a new motorcycle.

Interest on this loan is charged at a rate of 6.4% per annum, compounding quarterly.

Arthur will repay the loan in full using quarterly repayments of \$1515.18. The final payment will differ slightly from the previous repayments.

Determine the total cost of repaying the loan, the final payment, and the number of payments required to pay off the loan.

Source: VCAA 2023 General Mathematics Written Examination 2 Question 5

|                                              |             |            |
|----------------------------------------------|-------------|------------|
| $final\_pmt(6.4, -1515.18, 3 \cdot 10^4, 4)$ |             |            |
| Reducing Balance Loan                        |             |            |
| I = 6.40%, Regular Pmt = \$ 1515.18          |             |            |
| PV = \$ 30000.00                             |             |            |
| "No."                                        | "Final Pmt" | "Total"    |
| 23                                           | "3006.36"   | "36340.32" |
| 24                                           | "1515.04"   | "36364.18" |
| Done                                         |             |            |

## Finance Solver

Solves for a particular parameter based on the input values provided

### Syntax

*finance\_solve(N, %I, PV, Pmt, FV, CpY)*

*N* represents the number of payment periods

*%I* represents the interest rate per annum

*PV* represents the present value

*Pmt* represents the payment per period

*FV* represents the final value

*CpY* represents the number of periods per annum

### Example

Bob has a student loan is \$50,000 with an interest rate of 5.00% per annum, compounding monthly. Bob makes a payment of \$500 every month. After one year, what is the final balance of Bob's student loan?

```
finance_solve(12,5,-50000,500,x,12)
46418.6671483
```

## Matrices

### Communication Matrix

Determines the paths which the sender could use to communicate to the receiver. The rows represent the senders, while columns represent receivers.

#### Syntax

$com(Matrix, Start, End)$

Where,

$Matrix$  represents the communication matrix

$Start$  represents the sender

$End$  represents the receiver

#### Example

The communication matrix below shows the direct paths by which messages can be sent between two people in a group of six people, U to Z.

|               |          | <i>receiver</i> |          |          |          |          |          |
|---------------|----------|-----------------|----------|----------|----------|----------|----------|
|               |          | <i>U</i>        | <i>V</i> | <i>W</i> | <i>X</i> | <i>Y</i> | <i>Z</i> |
| <i>sender</i> | <i>U</i> | 0               | 1        | 1        | 0        | 1        | 1        |
|               | <i>V</i> | 1               | 0        | 1        | 0        | 1        | 0        |
|               | <i>W</i> | 1               | 1        | 0        | 1        | 0        | 1        |
|               | <i>X</i> | 0               | 1        | 0        | 0        | 1        | 1        |
|               | <i>Y</i> | 0               | 0        | 1        | 1        | 0        | 1        |
|               | <i>Z</i> | 1               | 1        | 0        | 1        | 1        | 0        |

Source: VCAA 2019 Further Mathematics Written Examination 1 Section B Module 1 Question 7

In how many ways can  $Y$  get a message to  $W$ ?

**Example – Continued**

|                                                                                                                                                                                                             |  |               |             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|---------------|-------------|
| $com \left( \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}, 5, 3 \right)$ |  |               |             |
|                                                                                                                                                                                                             |  | "Path"        | "Receivers" |
|                                                                                                                                                                                                             |  | "EC"          | 1.          |
|                                                                                                                                                                                                             |  | "EDBC"        | 3.          |
|                                                                                                                                                                                                             |  | "EFBC"        | 3.          |
|                                                                                                                                                                                                             |  | "EFAC"        | 3.          |
|                                                                                                                                                                                                             |  | "EFDBC"       | 4.          |
|                                                                                                                                                                                                             |  | "EDFBC"       | 4.          |
|                                                                                                                                                                                                             |  | "EDFAC"       | 4.          |
|                                                                                                                                                                                                             |  | "EFABC"       | 4.          |
|                                                                                                                                                                                                             |  | "EFBAC"       | 4.          |
|                                                                                                                                                                                                             |  | "EDBAC"       | 4.          |
|                                                                                                                                                                                                             |  | "EFDBAC"      | 5.          |
|                                                                                                                                                                                                             |  | "EDFBAC"      | 5.          |
|                                                                                                                                                                                                             |  | "EDFABC"      | 5.          |
|                                                                                                                                                                                                             |  | "Total: 13. " | "□"         |

## Analyse Dominance

Determines the two-step, and total dominance from the one-step dominance matrix.

### Syntax

$dom\_analyse(Matrix)$

Where,  $Matrix$  represents the one-step dominance matrix.

### Example

Five friends, Bhavi ( $B$ ), Kai ( $K$ ), Oscar ( $O$ ), Sian ( $S$ ) and Xavier ( $X$ ), played a round-robin table tennis tournament. Each friend played each of the others once. Every game had a winner and a loser.

The one-step dominance matrix constructed from the tournament's results is shown below.

|        |     | loser |     |     |     |     |
|--------|-----|-------|-----|-----|-----|-----|
|        |     | $B$   | $K$ | $O$ | $S$ | $X$ |
| winner | $B$ | 0     | 1   | 0   | 1   | 0   |
|        | $K$ | 0     | 0   | 0   | 0   | 1   |
|        | $O$ | 1     | 1   | 0   | 1   | 0   |
|        | $S$ | 0     | 1   | 0   | 0   | 0   |
|        | $X$ | 1     | 0   | 1   | 1   | 0   |

A '1' in this matrix shows that the player named in that row defeated the player named in that column.

For example, the '1' in row 3, column 4 shows that Oscar defeated Sian.

Which one of the following tables shows the number of one-step and two-step dominances accumulated by each player in the tournament?

Source: VCAA 2022 Further Mathematics-NHT Written Examination 1 Section B Module 1 Question 6

$dom\_analyse\left(\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}\right)$

►Results:

| "Player" | "Won"   | "Lost"  |
|----------|---------|---------|
| "A"      | "B,D"   | "C,E"   |
| "B"      | "E"     | "A,C,D" |
| "C"      | "A,B,D" | "E"     |
| "D"      | "B"     | "A,C,E" |
| "E"      | "A,C,D" | "B"     |

►Dominance:

| "Player" | "One" | "Two" | "Total" |
|----------|-------|-------|---------|
| "A"      | 2     | 2     | 4       |
| "B"      | 1     | 3     | 4       |
| "C"      | 3     | 4     | 7       |
| "D"      | 1     | 1     | 2       |
| "E"      | 3     | 6     | 9       |

## Dominance Solve

Determines the results matrix based off the one-step and two-step dominances of competitors. The winners are the rows, and the losers are the columns.

### Syntax

$dom\_solve(Step1, Step2)$

Where,  $Step1$  and  $Step2$  are the one-step and two-step dominances respectively.

### Example

Five staff members in Elena's office played a round-robin video game tournament, where each employee played each of the other employees once. In each game there was a winner and a loser.

A table of their one-step and two-step dominances was prepared to summarise the results.

| Staff member    | One-step dominance | Two-step dominance |
|-----------------|--------------------|--------------------|
| Ike ( $I$ )     | 3                  | 5                  |
| Joelene ( $J$ ) | 3                  | 4                  |
| Katie ( $K$ )   | 1                  | 1                  |
| Leslie ( $L$ )  | 1                  | 2                  |
| Mikki ( $M$ )   | 2                  | 4                  |

Consider the results matrix shown below.

A '1' in this matrix shows that the player named in that row defeated the player named in that column.

A '0' in this matrix shows that the player named in that row lost to the player named in that column.

Use all of the information provided to complete the results matrix.

Source: VCAA 2021 Further Mathematics Written Examination 2 Section B Module 1 Question 4

$dom\_solve\left(\begin{pmatrix} 3 \\ 3 \\ 1 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \\ 1 \\ 2 \\ 4 \end{pmatrix}\right)$

►Results Matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

►Results:

| "Player" | "Won"   | "Lost"  |
|----------|---------|---------|
| "A"      | "B,C,D" | "E"     |
| "B"      | "C,D,E" | "A"     |
| "C"      | "D"     | "A,B,E" |
| "D"      | "E"     | "A,B,C" |
| "E"      | "A,C"   | "B,D"   |

## Square Root Matrix

Determines the square root of a matrix which was produced by multiplying two **binary** matrices together.

This can be used for finding the one-step dominance from the two-step dominance.

### Syntax

$\text{sqrt\_mat}(\text{Matrix})$

Where,  $\text{Matrix}$  represents squared matrix.

### Example

A badminton competition is held between four players, Amanda ( $A$ ), Ben ( $B$ ), Carlos ( $C$ ) and Darius ( $D$ ).

In the competition, each player competes in one game with each of the other three players.

The matrix  $S^2$  below shows the two-step dominance that each player has over the other players.

$$S^2 = \begin{matrix} & \begin{matrix} \text{loser} \\ A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

Source: VCAA 2017 Further Mathematics-NHT Written Examination I Section B Module 1 Question 7

Determine the one-step dominance matrix.

$$\text{sqrt\_mat} \left( \begin{bmatrix} 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right)$$

►Results Matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

►Results:

$$\begin{bmatrix} \text{"Player"} & \text{"Won"} & \text{"Lost"} \\ \text{"A"} & \text{"B,C,D"} & \text{"None"} \\ \text{"B"} & \text{"None"} & \text{"A,C,D"} \\ \text{"C"} & \text{"B"} & \text{"A,D"} \\ \text{"D"} & \text{"B,C"} & \text{"A"} \end{bmatrix}$$

## Outcomes

Determines the possible outcomes and final player rankings based off a given matrix.

### Syntax

*outcomes(Matrix)*

Where, *Matrix* represents the current one-step dominance matrix with variables in place of blank spaces.

**Note:** You may enter the entire matrix, or just the upper triangle as a string. Both cases are shown below.

### Example 1

Five competitors, Andy (*A*), Brie (*B*), Cleo (*C*), Della (*D*) and Eddie (*E*), participate in a darts tournament. Each competitor plays each of the other competitors once only, and each match results in a winner and a loser.

The matrix below shows the results of this darts tournament.

There are still two matches that need to be played.

|               |          | <i>loser</i> |          |          |          |          |
|---------------|----------|--------------|----------|----------|----------|----------|
|               |          | <i>A</i>     | <i>B</i> | <i>C</i> | <i>D</i> | <i>E</i> |
| <i>winner</i> | <i>A</i> | 0            | ...      | 0        | 1        | 0        |
|               | <i>B</i> | ...          | 0        | 1        | 0        | 1        |
|               | <i>C</i> | 1            | 0        | 0        | ...      | 1        |
|               | <i>D</i> | 0            | 1        | ...      | 0        | 0        |
|               | <i>E</i> | 1            | 0        | 0        | 1        | 0        |

A '1' in the matrix shows that the competitor named in that row defeated the competitor named in that column.

For example, the '1' in row 2, column 3 shows that Brie defeated Cleo.

A '...' in the matrix shows that the competitor named in that row has not yet played the competitor named in that column.

The winner of this darts tournament is the competitor with the highest sum of their one-step and two-step dominances.

Which player, by winning their remaining match, will ensure that they are ranked first by the sum of their one-step and two-step dominances?

*Source: VCAA 2020 Further Mathematics Written Examination 1 Section B Module 1 Question 9*

|          |                                                                                                                                       |       |      |      |      |      |
|----------|---------------------------------------------------------------------------------------------------------------------------------------|-------|------|------|------|------|
| outcomes | $\begin{pmatrix} 0 & x & 0 & 1 & 0 \\ x & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & x & 1 \\ 0 & 1 & x & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$ |       |      |      |      |      |
|          |                                                                                                                                       |       |      |      |      |      |
|          |                                                                                                                                       |       |      |      |      |      |
|          |                                                                                                                                       |       |      |      |      |      |
|          |                                                                                                                                       |       |      |      |      |      |
|          |                                                                                                                                       |       |      |      |      |      |
|          |                                                                                                                                       | "Win" | "BD" | "BC" | "AD" | "AC" |
|          |                                                                                                                                       | "A"   | 5    | 5    | 1    | 3    |
|          |                                                                                                                                       | "B"   | 1    | 1    | 1    | 2    |
|          |                                                                                                                                       | "C"   | 3    | 2    | 1    | 1    |
|          |                                                                                                                                       | "D"   | 2    | 3    | 1    | 5    |
|          |                                                                                                                                       | "E"   | 3    | 3    | 1    | 3    |

## Example 2

Instead of entering the entire matrix, we can enter just the upper triangle in the same fashion inside a string. You can obtain a string by pressing “Ctrl + ×”.

Using the same question as in **Example 1**, we see that the upper triangle going from top to bottom and left to right is given by “x010101x10”.

*outcomes*(“x010101x10”)

---

| "Win " | "BD" | "BC" | "AD" | "AC" |
|--------|------|------|------|------|
| "A"    | 5    | 5    | 1    | 3    |
| "B"    | 1    | 1    | 1    | 2    |
| "C"    | 3    | 2    | 1    | 1    |
| "D"    | 2    | 3    | 1    | 5    |
| "E"    | 3    | 3    | 1    | 3    |

---

*Done*

## Leslie Matrix

Determines the Leslie matrix based off the given fertility and survival rates.

### Syntax

*leslie\_mat(fertility\_rates, survival\_rates)*

Where,

*fertility\_rates* is a list containing the fertility rates of all age groups.

*survival\_rates* is a list containing the survival rates of all age groups except the final age group.

### Example

#### Question 30

Data has been collected on the female population of a species of mammal located on a remote island.

The female population has been divided into three age groups, with the initial population (at the time of data collection), the birth rate, and the survival rate of each age group shown in the table below.

|                    | Age group (years) |      |      |
|--------------------|-------------------|------|------|
|                    | 0–2               | 2–4  | 4–6  |
| Initial population | 2100              | 6400 | 4260 |
| Birth rate         | 0                 | 1.8  | 1.2  |
| Survival rate      | 0.7               | 0.6  | 0    |

The Leslie matrix ( $L$ ) that may be used to model this particular population is

*Source: VCAA 2024 General Mathematics Written Examination 1 Question 30*

$$\text{leslie\_mat}(\{0, 1.8, 1.2\}, \{0.7, 0.6\})$$


---


$$\begin{bmatrix} 0. & 1.8 & 1.2 \\ 0.7 & 0. & 0. \\ 0. & 0.6 & 0. \end{bmatrix}$$


---

*Done*

## Leslie Table

Determines the state table up to a specified number of generations.

### Syntax

$leslie(L, R0, Iter)$

Where,

$L$  represents the Leslie matrix which models the population.

$R0$  represents the initial state matrix.

$Iter$  represents the number of rows in the table you wish to generate.

### Example

#### Question 11 (3 marks)

A population of a native animal species lives near the construction site.

To ensure that the species is protected, information about the initial female population was collected at the beginning of 2023. The birth rates and the survival rates of the females in this population were also recorded.

This species has a life span of 4 years and the information collected has been categorised into four age groups: 0–1 year, 1–2 years, 2–3 years, and 3–4 years.

This information is displayed in the initial population matrix,  $R_0$ , and the Leslie matrix,  $L$ , below.

$$R_0 = \begin{bmatrix} 70 \\ 80 \\ 90 \\ 40 \end{bmatrix} \quad L = \begin{bmatrix} 0.4 & 0.75 & 0.4 & 0 \\ 0.4 & 0 & 0 & 0 \\ 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{bmatrix}$$

- ii. complete the following table, showing the initial female population, and the predicted female population after one year, for each of the age groups.

1 mark

Source: VCAA 2024 General Mathematics Written Examination 2 Question 11

$$leslie \left( \begin{bmatrix} 0.4 & 0.75 & 0.4 & 0 \\ 0.4 & 0 & 0 & 0 \\ 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{bmatrix}, \begin{bmatrix} 70 \\ 80 \\ 90 \\ 40 \end{bmatrix}, 1 \right)$$

► State Table:

| "Age"     | "1"  | "2" | "3" | "4" | "Total" |
|-----------|------|-----|-----|-----|---------|
| "Initial" | 70.  | 80. | 90. | 40. | 280.    |
| "Gen 1"   | 124. | 28. | 56. | 45. | 253.    |

► Long Term Decay:

Common ratio: 0.8832326847

11.67673153% decrease

► Long Term Proportions:

| "Age"   | "1"    | "2"    | "3"    | "4"    |
|---------|--------|--------|--------|--------|
| "Prop." | 0.4963 | 0.2248 | 0.1781 | 0.1008 |

## Steady State

Determines the steady state matrix using the transition matrix and the addition matrix.

### Syntax

$steady\_state(T, B)$

Where,

$T$  represents the transition matrix.

$B$  represents the addition matrix.

### Example

#### Question 27

The following transition matrix,  $T$ , models the movement of a species of bird around three different locations,  $M$ ,  $N$  and  $O$  from one day to the next.

$$T = \begin{array}{c} \begin{array}{ccc} & \text{this day} & \\ & M & N & O \\ \begin{bmatrix} \frac{1}{3} & 0 & \frac{9}{10} \\ \frac{1}{3} & 1 & \frac{1}{10} \\ \frac{1}{3} & 0 & 0 \end{bmatrix} & \begin{array}{l} M \\ N \\ O \end{array} & \text{next day} \end{array} \end{array}$$

Which one of the following statements best represents what will occur in the long term?

- A. No birds will remain at location  $M$ .
- B. No birds will remain at location  $N$ .
- C. All of the birds will end up at location  $M$ .
- D. All of the birds will end up at location  $O$ .
- E. An equal number of birds will be at all three locations.

Source: VCAA 2023 General Mathematics Written Examination 1 Question 27

$$steady\_state \left( \begin{bmatrix} \frac{1}{3} & 0 & \frac{9}{10} \\ \frac{1}{3} & 1 & \frac{1}{10} \\ \frac{1}{3} & 0 & 0 \end{bmatrix}, 0 \right) \begin{bmatrix} 0. \\ 1. \\ 0. \end{bmatrix}$$

Determines the state table up to the specific number of generations using the initial state matrix.

$$transition(T, R0, B, Iter)$$

$T$  represents the transition matrix.

$R0$  represents the initial state matrix.

$B$  represents the addition matrix.

*Iter* represents the number of rows in the table you wish to generate.

To encourage more construction workers ( $C$ ) to stay, the construction company has given workers an incentive to move into the job of foreman ( $F$ ).

Matrix  $R$  below shows the ways in which staff are expected to change their jobs from year to year with this new incentive in place.

$$R = \begin{matrix} & \begin{matrix} \text{this year} \\ C & F & M & L \end{matrix} \\ \begin{matrix} C \\ F \\ M \\ L \end{matrix} & \begin{bmatrix} 0.4 & 0.2 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0.2 & 0.3 & 0 \\ 0.2 & 0.4 & 0.3 & 1 \end{bmatrix} \end{matrix} \begin{matrix} C \\ F \\ M \\ L \end{matrix} \begin{matrix} \\ \\ \text{next year} \\ \end{matrix}$$

The site always requires at least 330 construction workers.

To ensure that this happens, the company hires an additional 190 construction workers ( $C$ ) at the beginning of 2024 and each year thereafter.

The matrix  $V_{n+1}$  will then be given by

$$V_{n+1} = R V_n + Z, \text{ where}$$

$$V_0 = \begin{bmatrix} 330 \\ 50 \\ 10 \\ 0 \end{bmatrix} \begin{matrix} C \\ F \\ M \\ L \end{matrix} \quad Z = \begin{bmatrix} 190 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} C \\ F \\ M \\ L \end{matrix} \text{ and } n \text{ is the number of years after 2023.}$$

- c. How many more staff are there on the site in 2024 than there were in 2023? 1 mark
- d. Based on this new model, the company has realised that in the long term there will be more than 200 foremen on site.
- In which year will the number of foremen first be above 200? 1 mark

Source: VCAA 2024 General Mathematics Written Examination 2 Question 12

**Example – Continued**

$$transition \left( \begin{pmatrix} 0.4 & 0.2 & 0 & 0 \\ 0.4 & 0.2 & 0.4 & 0 \\ 0 & 0.2 & 0.3 & 0 \\ 0.2 & 0.4 & 0.3 & 1 \end{pmatrix} \begin{bmatrix} 330 \\ 50 \\ 10 \\ 0 \end{bmatrix} \begin{bmatrix} 190 \\ 0 \\ 0 \\ 0 \end{bmatrix}, 4 \right)$$

►State Table:

| "Group"    | "A"    | "B"    | "C"   | "D"    | "Total " |
|------------|--------|--------|-------|--------|----------|
| "Initial " | 330.   | 50.    | 10.   | 0.     | 390.     |
| "Iter 1 "  | 332.   | 146.   | 13.   | 89.    | 491.     |
| "Iter 2 "  | 352.   | 167.2  | 33.1  | 217.7  | 552.3    |
| "Iter 3 "  | 364.24 | 187.48 | 43.37 | 364.91 | 595.09   |
| "Iter 4 "  | 373.19 | 200.54 | 50.51 | 525.76 | 624.24   |

►Death/Absent Group Detected

Total does not include Group D

►System's stability:

System does not stabilise

## Inverse Transition

Determines the state table from the given current generation, current state matrix, transition matrix, and addition matrix.

### Syntax

$transition\_inv(T, Rn, B, Iter)$

Where,

$T$  represents the transition matrix.

$Rn$  represents the current state matrix.

$B$  represents the addition matrix.

$Iter$  represents the current generation.

### Example

#### Question 32

A large sporting event is held over a period of four consecutive days: Thursday, Friday, Saturday and Sunday.

People can watch the event at four different sites throughout the city: Botanical Gardens ( $G$ ), City Square ( $C$ ), Riverbank ( $R$ ) or Main Beach ( $M$ ).

Let  $S_n$  be the state matrix that shows the number of people at each location  $n$  days after Thursday.

The expected number of people at each location can be determined by the matrix recurrence rule

$$S_{n+1} = TS_n + A$$

$$\text{where } T = \begin{array}{c} \begin{array}{ccccc} & \text{this day} & & & \\ & G & C & R & M \\ \begin{bmatrix} 0.4 & 0.2 & 0.4 & 0 \\ 0.4 & 0.1 & 0.3 & 0.3 \\ 0.1 & 0.4 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.2 & 0.5 \end{bmatrix} & \begin{array}{l} G \\ C \\ R \\ M \end{array} \end{array} & \begin{array}{c} \text{next day} \end{array} & \text{and} & A = \begin{array}{c} \begin{bmatrix} 300 \\ 200 \\ 100 \\ 300 \end{bmatrix} \begin{array}{l} G \\ C \\ R \\ M \end{array} \end{array}$$

$$\text{Given the state matrix } S_3 = \begin{array}{c} \begin{bmatrix} 5620 \\ 6386 \\ 4892 \\ 6902 \end{bmatrix} \begin{array}{l} G \\ C \\ R \\ M \end{array} \end{array}$$

the number of people watching the event at the Botanical Gardens ( $G$ ) from Thursday to Sunday has

Source: VCAA 2024 General Mathematics Written Examination 1 Question 32

$$transition\_inv \left( \begin{bmatrix} 0.4 & 0.2 & 0.4 & 0 \\ 0.4 & 0.1 & 0.3 & 0.3 \\ 0.1 & 0.4 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.2 & 0.5 \end{bmatrix}, \begin{bmatrix} 5620 \\ 6386 \\ 4892 \\ 6902 \end{bmatrix}, \begin{bmatrix} 300 \\ 200 \\ 100 \\ 300 \end{bmatrix}, 3 \right)$$

---

►State Table:

| "Group"    | "A"   | "B"   | "C"   | "D"   | "Total" |
|------------|-------|-------|-------|-------|---------|
| "Initial " | 4924. | 4732. | 6540. | 4904. | 21100.  |
| "Iter 1 "  | 5832. | 6076. | 4120. | 5972. | 22000.  |
| "Iter 2 "  | 5496. | 6168. | 4720. | 6516. | 22900.  |
| "Iter 3 "  | 5620. | 6386. | 4892. | 6902. | 23800.  |

---

Done

## Networks and Decision Mathematics

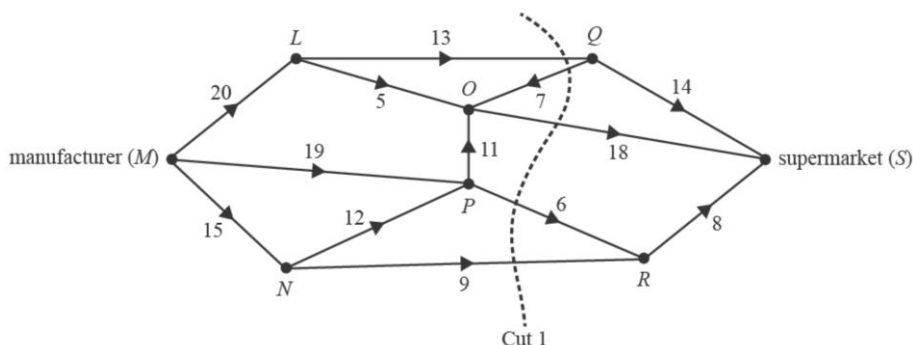
### Introduction

This section will discuss how to input networks for the UDFs to use. It is important to know the difference between the two types of graphs (vertex-labelled & edge-labelled) to be able to properly use the UDFs. Furthermore, it is highly recommended that you practice inputting these on your handheld CAS as this is what you will be doing in the end of year VCE exam.

### Vertex Labelled Graphs (VLG)

As the name suggests, these are graphs where the vertices are assigned labels (such as A, B, C, etc) rather than the edges. These are common for Dijkstra's algorithm, and flow questions.

An example of a vertex labelled graph is shown below.



Source: VCAA 2024 VCE General Mathematics Examination 2 Question 14

As you can see from the graph above, the vertices of the graph have been labeled with the letters (O, M, N, etc), while the edges are unlabelled.

### Syntax

*label transitions*

Where,

*label* represents the letter label of the current vertex

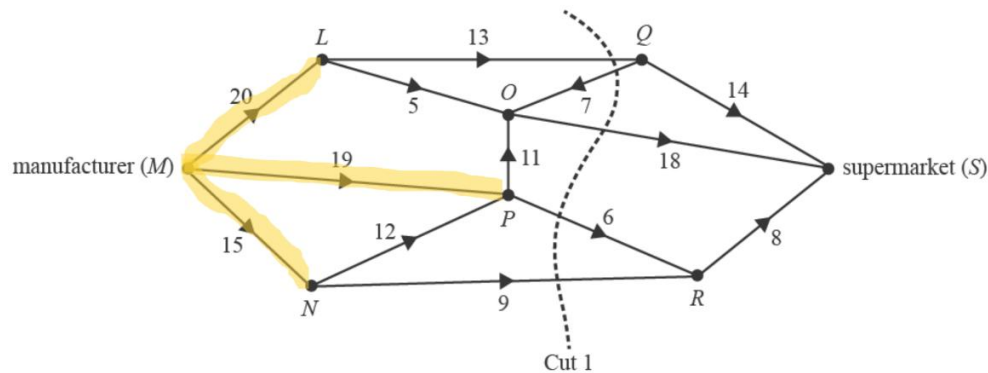
*transitions* represent the vertices which can be reached from the current vertex, and the associated weight

Contact

lazymath2024@gmail.com

### Example 1

Using the previous graph, suppose we start off at vertex M. We have three possible transitions as shown in the diagram below.

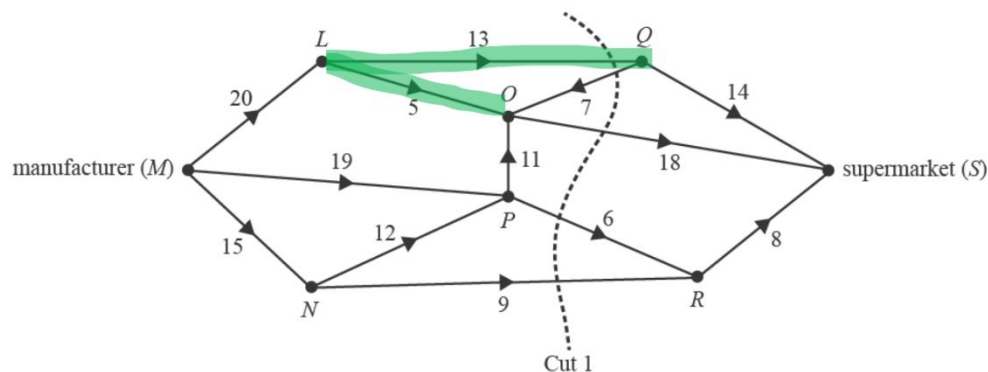


After identifying our transitions, we can use our *label transitions* syntax to write our vertex. Here, we would use

$$m\ l20\ p19\ n15$$

This is because ‘*m*’ is the label of vertex M, and from vertex M we can transition to either ‘*l*’ with a cost of 20, ‘*p*’ with a cost of 19, and finally ‘*n*’ with a cost of 15.

Now suppose we are at vertex L. We have two possible transitions as shown in the diagram below.



Using the same syntax as before, we would obtain

$$l\ q13\ o5$$

This is because ‘*l*’ is the label of the vertex L, and from vertex L we can transition to either ‘*q*’ with a cost of 13, and ‘*o*’ with a cost of 5.

### Example 1 – Continued

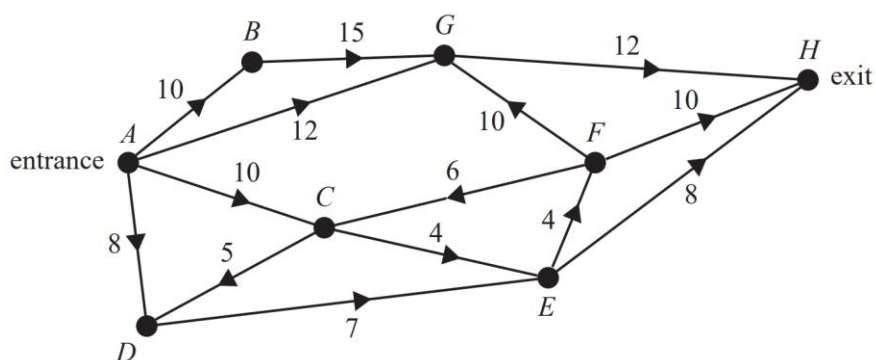
Using the same syntax for the rest of the vertices in the graph, we obtain.

$$\{ml20p19n15, lq13o5, np12r9, os18, pol1r6, qo7s14, rs8, s\}$$

**Note:** Spaces have been omitted to show what should be entered into the CAS.

After completing the list, you can save this into the CAS under a variable for later use. This will then allow you to use this graph for the UDFs.

### Example 2



Source: VCAA 2023 General Mathematics Examination 1 Question 39 & Question 40

To practice the syntax, have a go at this one yourself! The solution is shown at the bottom of the page.

### Solution

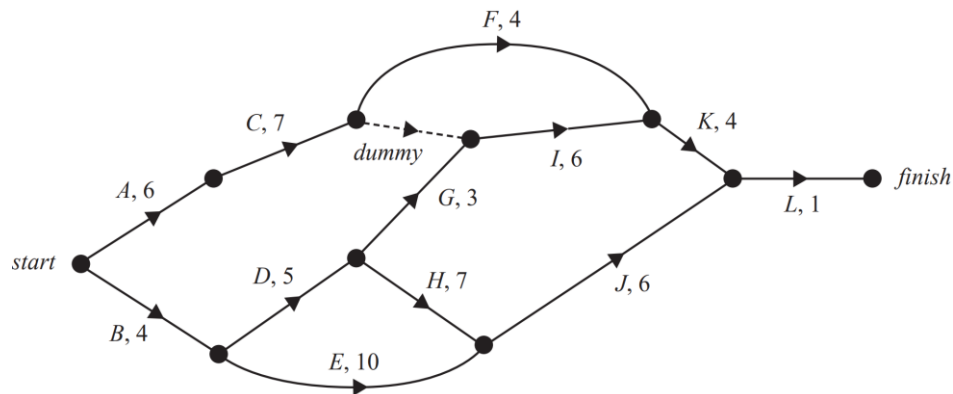
$$\{ab10g12c10d8, bg15, cd5e4, de7, ef4h8, fg10c6h10, gh12, h\}$$

Contact  
lazymath2024@gmail.com

## Edge Labelled Graphs (ELG)

These are graphs where the edges are labelled, while the vertices are unlabelled. These are common for crashing and float time questions.

An example of an edge labelled graph is shown below.



*Source: VCAA 2023 General Mathematics Examination 2 Question 14*

As you can see from the graph above, the edges are labelled with letters (A, B, C, etc), while the vertices are unlabelled.

Since the edges of these graphs are used to represent activities in general mathematics, they will be referred to as activities in this guide.

### Syntax

*label duration predecessors*

Where,

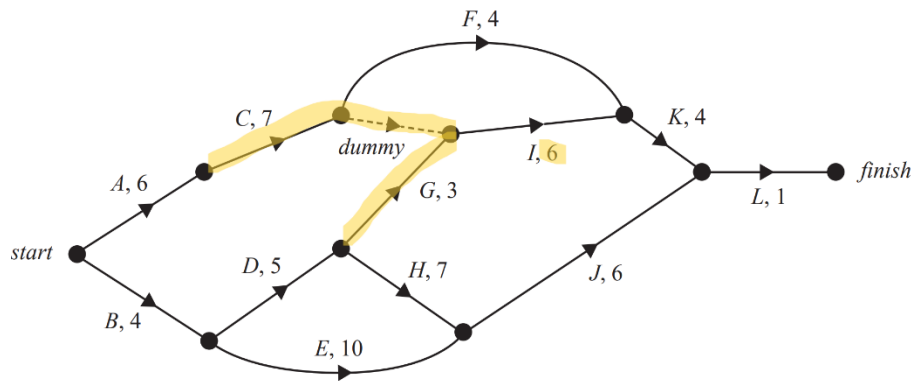
*label* represents the letter used to represent the activity

*duration* represents the value of the activity

*predecessors* represent the predecessors of the activity

### Example 1

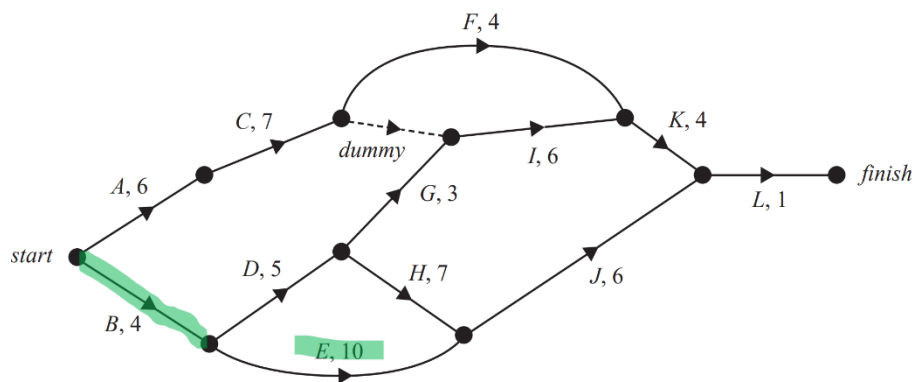
Using the graph, suppose we start at activity I, we have two predecessor activities (C and G), and a duration of 6 as shown below.



After identifying the duration and predecessors, we can use our *label duration predecessors* syntax to write the activity. Here, we would have

*i 6 c g*

This is because the label for our activity is 'i', it has a duration of '6', and its predecessors are 'c' and 'g'. Now suppose we are at activity E. We have one predecessor and a duration of 10 as shown in the diagram below.



Using the same syntax as before, we would obtain

*e 10 b*

This is because 'e' is the label of the activity E, activity E has a duration of 10, and the only predecessor of activity E is activity B.

### Example 1 – Continued

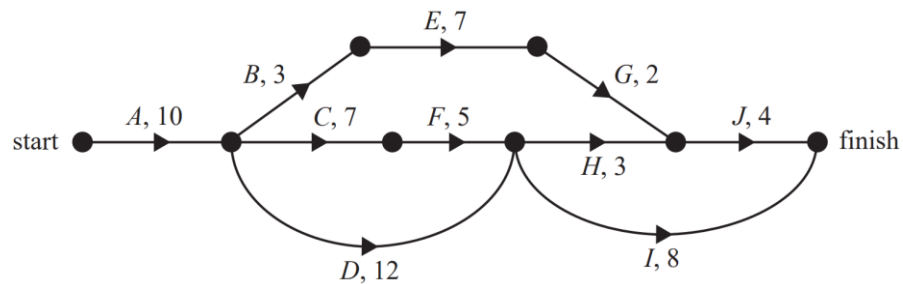
Using the same syntax for the rest of the activities in the graph, we obtain.

$$\{a6, b4, c7a, d5b, e10b, f4c, g3d, h7d, i6gc, j6hk, k4if, llkj\}$$

**Note:** Spaces have been omitted to show what should be entered into the CAS.

After completing the list, you can save this into the CAS under a variable for later use. This will then allow you to use this graph for the UDFs.

### Example 2



Source: VCAA 2022 General Mathematics Examination 2 Question 2

To practice the syntax, have a go at this one yourself! The solution is shown at the bottom of the page.

### Solution

$$\{a10, b3a, c7a, d12a, e7b, f5c, g2e, h3df, i8df, j4hg\}$$

## Dijkstra's Algorithm

Uses Dijkstra's algorithm to determine the shortest pathway in a graph.

### Syntax

$dijkstra(VLG, Start, End)$

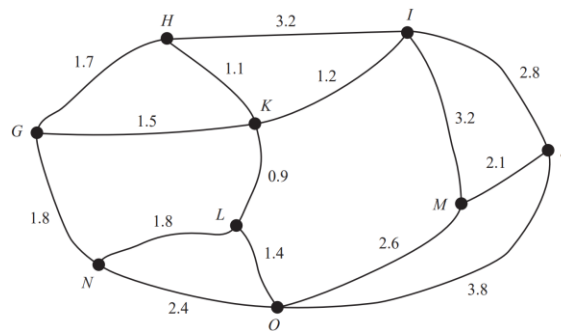
Where,

$VLG$  represents a vertex labelled graph

$Start$  represents the label of the start vertex

$End$  represents the label of the end vertex

### Example



Source: VCAA 2023 General Mathematics Examination 2 Question 13

```
graph {gh17n18k15,hg17k11i32,ih32k12j28m32,ji28m21o38,kh11g15i12l9,ln18k9o14,mj21i32o26,ng18l18o24,on24l14m26j38}
dijkstra(graph,g,m)
```

►Dijkstra:

►Working:

| "Iter" | "Visited" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" |
|--------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0.     | "G"       | 0.  | ∞   | ∞   | ∞   | ∞   | ∞   | ∞   | ∞   | ∞   |
| 1.     | "K"       | 0.  | 17. | ∞   | ∞   | 15. | ∞   | ∞   | 18. | ∞   |
| 2.     | "H"       | 0.  | 17. | 27. | ∞   | 15. | 24. | ∞   | 18. | ∞   |
| 3.     | "N"       | 0.  | 17. | 27. | ∞   | 15. | 24. | ∞   | 18. | ∞   |
| 4.     | "L"       | 0.  | 17. | 27. | ∞   | 15. | 24. | ∞   | 18. | 42. |
| 5.     | "I"       | 0.  | 17. | 27. | ∞   | 15. | 24. | ∞   | 18. | 38. |
| 6.     | "O"       | 0.  | 17. | 27. | 55. | 15. | 24. | 59. | 18. | 38. |
| 7.     | "J"       | 0.  | 17. | 27. | 55. | 15. | 24. | 59. | 18. | 38. |
| 8.     | "M"       | 0.  | 17. | 27. | 55. | 15. | 24. | 59. | 18. | 38. |

►Result:

Minimum Distance: 59.

Optimal Path: GKIM

Done

```
graph:={gh17n18k15,hg17k11i32,ih32k12j28m32,ji28m21o38,kh11g15i12l9,ln18k9o14,mj21i32o26,ng18l18o24,on24l14m26j38}
```

Contact

lazymath2024@gmail.com

## Prim's Algorithm

Uses Prim's Algorithm to find a minimum spanning tree of the graph.

### Syntax

$\text{prim}(VLG)$

Where,

$VLG$  represents a vertex labelled graph

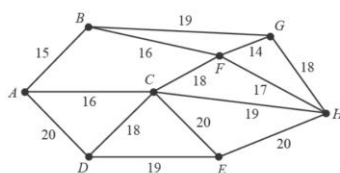
#### Question 36

Eight houses in an estate are to be connected to the internet via underground cables.

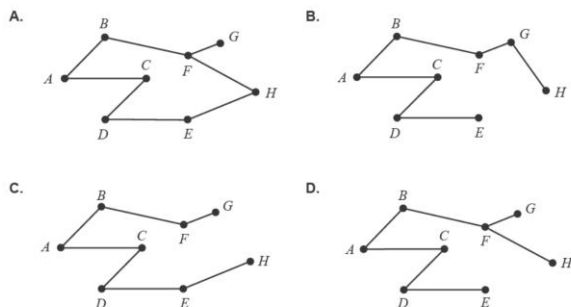
The network below shows the possible connections between the houses.

The vertices represent the houses.

The numbers on the edges represent the length of cable connecting pairs of houses, in metres.



The graph that represents the minimum length of cable needed to connect all the houses is



Source: VCAA 2024 General Mathematics Examination 1 Question 36

$\text{graph} = \{ab15c16d20,bg19f16,ca16f18h19e20d18,da20c18e19,ed19c20h20,fb16g14c18h17,gb19f14h18,hg18f17c19e20\}$

$\text{prim}(\text{graph})$

#### Working:

| "Edge" | "Length" |
|--------|----------|
| "AB"   | 15.      |
| "BF"   | 16.      |
| "FG"   | 14.      |
| "AC"   | 16.      |
| "FH"   | 17.      |
| "CD"   | 18.      |
| "DE"   | 19.      |

#### Solution:

115.

Done

$\text{graph} := \{ab15c16d20,bg19f16,ca16f18h19e20d18,da20c18e19,ed19c20h20,fb16g14c18h17,gb19f14h18,hg18f17c19e20\}$

Contact

lazymath2024@gmail.com

## Hamiltonian

Finds the minimum Hamiltonian cycle or path in a graph. If a Hamiltonian cycle or path cannot be found it returns 'false'.

### Syntax

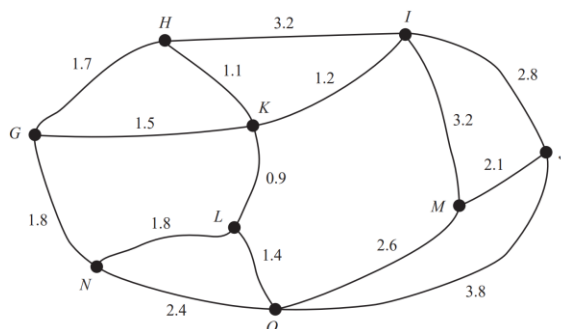
`hamiltonian(VLG, Start, End)`

Where,

*VLG* represents a vertex labelled graph

*Start* represents the label of the start vertex

*End* represents the label of the end vertex



- b. Reynold would like to visit all the landmarks and return to G.

Write down a route that Reynold could follow to minimise the total distance travelled.

1 mark

Source: VCAA 2023 General Mathematics Examination 2 Question 13

```
graph {gh17n18k15,hg17k11i32,ih32k12j28m32,ji28m21o38,kh11g15i12l9,ln18k9o14,mj21i32o26,ng18l18o24,on24l14m26j38}
hamiltonian(graph,g,g)
```

|         |              |        |
|---------|--------------|--------|
| "Type"  | "Nodes"      | "Cost" |
| "Cycle" | "GHKIJMOLNG" | 165    |

Done

```
graph:={gh17n18k15,hg17k11i32,ih32k12j28m32,ji28m21o38,kh11g15i12l9,ln18k9o14,mj21i32o26,ng18l18o24,on24l14m26j38}
```

**Warning:** This program has a long runtime, we do **not** recommend using it in an exam. If you would like to stop the program hold the 'on' button on your CAS calculator.

Contact

lazymath2024@gmail.com

## Traverse Vertices

Finds the minimum path or cycle which traverses all vertices of the graph.

### Syntax

`traverse_vertex(VLG, Start, End)`

Where,

*VLG* represents a vertex labelled graph

*Start* represents the label of the start vertex

*End* represents the label of the end vertex or a blank string

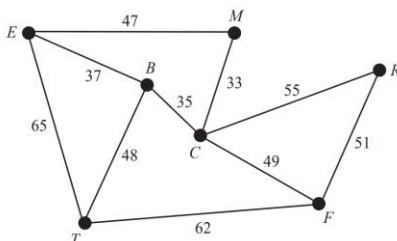
### Example

#### Question 15 (4 marks)

The theme park entrance (*E*), Roller-coaster (*R*), Mini-coaster (*M*), Bumper cars (*B*), Carousel (*C*), Tilt-A-Whirl (*T*), and Ferris wheel (*F*) are represented by the vertices of the graph below.

The edges on the graph represent the paths within the theme park.

The numbers on each edge represent the length, in metres, of each path.



The inspector decides that the path between the Bumper cars (*B*) and the Tilt-A-Whirl (*T*) needs to be closed for maintenance.

The next day Rowan will again need to collect money from the ticket booths at each of the attractions, starting and finishing at the entrance. To do this, Rowan may need to visit one or more attractions twice.

- c. If Rowan visits each attraction, starting and finishing at the entrance, what is the length of the shortest path Rowan can take, in metres?

1 mark

```
graph {em47b37t65,be37c35,cb35m33r55,me47c33,rc55f51,fc49r51t62,tf62e65}
traverse_vertex(graph,e,e)
```

►Repeated Vertices:

| "Vertex" | "Visited" |
|----------|-----------|
| "C"      | 2         |
| "E"      | 2         |

►Solution:

| "Type"  | "Nodes"     | "Cost" |
|---------|-------------|--------|
| "Cycle" | "EBCMCRFTE" | 371    |

Warning: May be inaccurate for large networks.

Done

**graph**:=`{em47b37t65,be37c35,cb35m33r55,me47c33,rc55f51,fc49r51t62,tf62e65}`

**Warning:** This program has a long runtime, we do **not** recommend using it in an exam. If you would like

to stop the program hold the 'on' button on your CAS calculator.

Contact

lazymath2024@gmail.com

## Traverse Edges

Finds the minimum Eulerian trail or circuit, if possible. Otherwise, it determines the minimum trail which traverses all edges.

### Syntax

`traverse_edge(VLG, Start, End)`

Where,

*VLG* represents a vertex labelled graph

*Start* represents the label of the start vertex

*End* represents the start vertex (for circuit), or a blank string (for trail)

### Example

- c. Shyla would like to travel along all the roads.  
To complete this journey in the minimum distance, she will travel along two roads twice.  
Shyla will leave from landmark *G* but end at a different landmark.
- Complete the following by filling in the boxes provided. 1 mark
- The two roads that will be travelled along twice are the roads between:
- vertex  and vertex
  - vertex  and vertex

*Source: VCAA 2023 General Mathematics Examination 2 Question 13*

```
graph {gh17n18k15,hg17k11i32,ih32k12j28m32,ji28m21o38,kh11g15i12l9,ln18k9o14,mj21i32o26,ng18l18o24,on24l14m26j38}
traverse_edge(graph,g,"")
```

```
►Repeated Edges:
┌"Edge"  "Cost"
├"JM"    21.
└"LN"    18.
►Total Cost:
354.
►Trail:
GHIJMIKGNLNOJMKH
```

Done

**graph:**={gh17n18k15,hg17k11i32,ih32k12j28m32,ji28m21o38,kh11g15i12l9,ln18k9o14,mj21i32o26,ng18l18o24,on24l14m26j38}

**Warning:** This program has a long runtime, we do **not** recommend using it in an exam. If you would like to stop the program hold the ‘on’ button on your CAS calculator.

Contact  
lazymath2024@gmail.com

## Float Time

Determines the float times for all activities in an edge labelled graph.

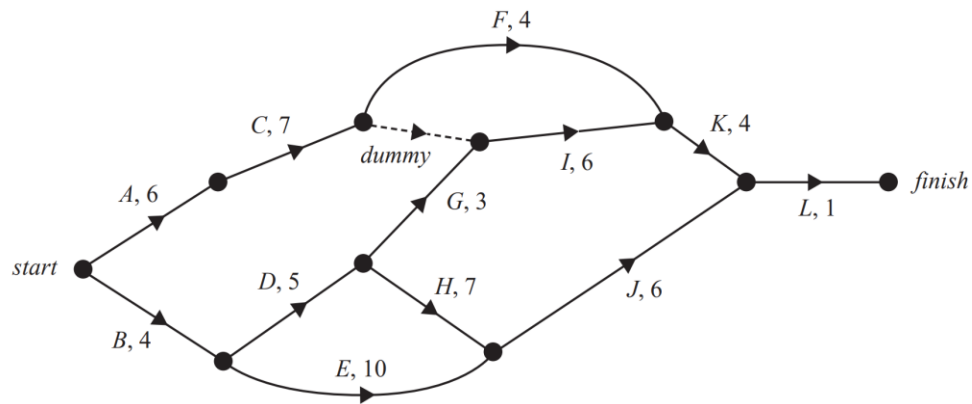
### Syntax

$\text{float\_time}(\text{ELG})$

Where,

$\text{ELG}$  is an edge labelled graph

### Example



Source: VCAA 2023 General Mathematics Examination 2 Question 13

c. How many activities have a float time of zero?

1 mark

$\text{graph} := \{a6, b4, c7a, d5b, e10b, f4c, g3d, h7d, i6cg, j6he, k4fi, l1jk\}$   $\{a6, b4, c7a, d5b, e10b, f4c, g3d, h7d, i6cg, j6he, k4fi, l1jk\}$   
 $\text{float\_time}(\text{graph})$

►Float Time:

| "Activity" | "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "Finish" |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------|
| "EST"      | 0.  | 0.  | 6.  | 4.  | 4.  | 13. | 9.  | 9.  | 13. | 16. | 19. | 23. | 24.      |
| "LST"      | 0.  | 1.  | 6.  | 5.  | 7.  | 15. | 10. | 10. | 13. | 17. | 19. | 23. | 24.      |
| "Float"    | 0.  | 1.  | 0.  | 1.  | 3.  | 2.  | 1.  | 1.  | 0.  | 1.  | 0.  | 0.  | 0.       |

►Non-Critical Activities:

$\{\text{"B"}, \text{"D"}, \text{"E"}, \text{"F"}, \text{"G"}, \text{"H"}, \text{"J"}\}$

Total: 7.

►Critical Activities:

$\{\text{"A"}, \text{"C"}, \text{"I"}, \text{"K"}, \text{"L"}\}$

Total: 5.

►Critical Path(s):

ACIKL

►Critical Time:

24.

Done

$\text{graph} := \{a6, b4, c7a, d5b, e10b, f4c, g3d, h7d, i6cg, j6he, k4fi, l1jk\}$

Contact

lazymath2024@gmail.com

## Dummy Activity

Determines the number and location of dummy activities for a given project network.

### Syntax

*dummy(ELG)*

Where, *ELG* represents an edge labelled graph

### Example

The project requires 11 activities, *A* to *K*.

Determine the number of dummy activities required and where they occur.

| Activity | Immediate predecessor(s) |
|----------|--------------------------|
| <i>A</i> | —                        |
| <i>B</i> | —                        |
| <i>C</i> | —                        |
| <i>D</i> | <i>A</i>                 |
| <i>E</i> | <i>B, F</i>              |
| <i>F</i> | <i>C</i>                 |
| <i>G</i> | <i>C</i>                 |
| <i>H</i> | <i>D, E</i>              |
| <i>I</i> | <i>F</i>                 |
| <i>J</i> | <i>E, G, I</i>           |
| <i>K</i> | <i>H, J</i>              |

Source: VCAA 2022 Further Mathematics-NHT Written Examination 1 Section B Module 2 Question 8

```
graph {a,b,c,da,ebf,fc,gc,hde,"if",jegi,khj}
dummy(graph)
```

►Dummy Activities:

```
[ "End"  "Start"
  "E"    "H"
  "E"    "J"
  "F"    "E"
  "Total" 3. ]
```

Warning: This program assumes multiple distinct activities can start and end at the same nodes.

Done

**graph**:=*{a,b,c,da,ebf,fc,gc,hde,"if",jegi,khj}*

## Project Crashing

Determines the reductions in the project's activities which result in the minimum cost within a specified budget.

### Syntax

*crashing*(*ELG*, *Tasks*, *Reduction(s)*, *Budget*, *Deadline*)

Where,

*ELG* represents an edge labelled graph

*Tasks* represents either a matrix or a list containing the activity labels

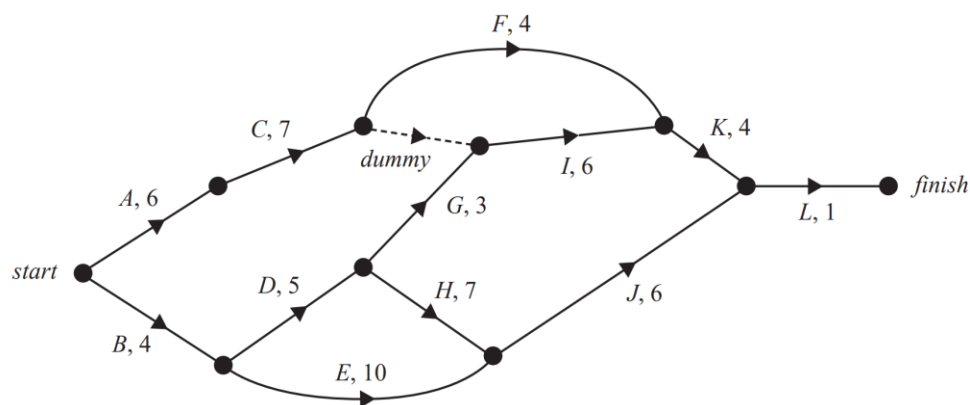
*Reduction(s)* represents either a number or a list used to specify the maximum reduction of each activity

*Budget* represents the available budget

*Deadline* represents either the target completion time (if positive), the maximum reduction in completion (if negative), or the maximum possible reduction (if a blank string)

**Warning:** Please enter the starting activities into the list in **alphabetical order** to ensure accuracy.

### Example 1



### Example 1 – Continued

The managers of the project are able to reduce the time, in days, of six activities.  
These reductions will result in an increase in the cost of completing the activity.  
The maximum decrease in time of any activity is two days.

| Activity        | <i>A</i> | <i>B</i> | <i>F</i> | <i>H</i> | <i>I</i> | <i>K</i> |
|-----------------|----------|----------|----------|----------|----------|----------|
| Daily cost (\$) | 1500     | 2000     | 2500     | 1000     | 1500     | 3000     |

- e. The managers of the project have a maximum budget of \$15 000 to reduce the time for several activities to produce the maximum reduction in the project's overall completion time.

Complete the table below, showing the reductions in individual activity completion times that would achieve the earliest completion time within the \$15 000 budget.

1 mark

| Activity | Reduction in completion time<br>(0, 1 or 2 days) |
|----------|--------------------------------------------------|
| <i>A</i> |                                                  |
| <i>B</i> |                                                  |
| <i>F</i> |                                                  |
| <i>H</i> |                                                  |
| <i>I</i> |                                                  |
| <i>K</i> |                                                  |

Source: VCAA 2023 General Mathematics Examination 2 Question 13

graph

$\{a6,b4,c7a,d5b,e10b,f4c,g3d,h7d,i6cg,j6he,k4fi,lljk\}$

crashing graph,  $\begin{pmatrix} a & 1500. \\ b & 2000. \\ f & 2500. \\ h & 1000. \\ i & 1500. \\ k & 3000. \end{pmatrix}, 2, 15000, \text{" "}$

►Working:

| "Path"      | "Start" | "↓2A" | "↓2B" | "↓2H" | "↓2I" | "↓1K" |
|-------------|---------|-------|-------|-------|-------|-------|
| "BDGIKL"    | 23      | 23    | 21    | 21    | 19    | 18    |
| "ACIKL"     | 24      | 22    | 22    | 22    | 20    | 19    |
| "ACFKL"     | 22      | 20    | 20    | 20    | 20    | 19    |
| "BEJL"      | 21      | 21    | 19    | 19    | 19    | 19    |
| "BDHJL"     | 23      | 23    | 21    | 19    | 19    | 19    |
| " " "       | " " "   | " " " | " " " | " " " | " " " | " " " |
| "Crit Time" | 24      | 23    | 22    | 22    | 20    | 19    |
| "Min Cost"  | 0       | 3000  | 7000  | 9000  | 12000 | 15000 |

►Solution:

| "Cost" | "Time" | <i>a</i> | <i>b</i> | <i>f</i> | <i>h</i> | <i>i</i> | <i>k</i> |
|--------|--------|----------|----------|----------|----------|----------|----------|
| 15000. | 19.    | 2.       | 2.       | 0.       | 2.       | 2.       | 1.       |

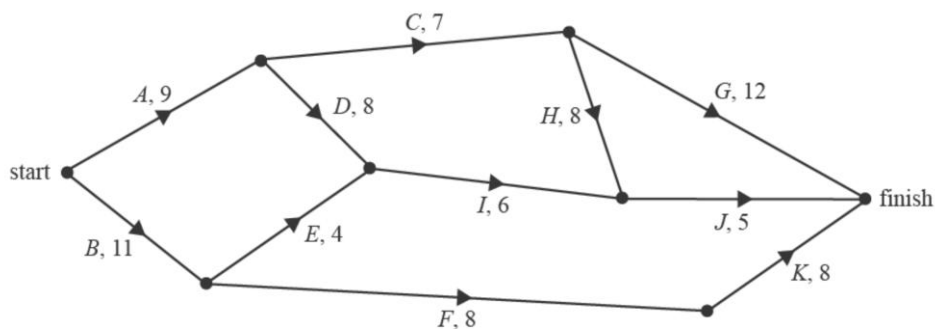
Done

graph:={a6,b4,c7a,d5b,e10b,f4c,g3d,h7d,i6cg,j6he,k4fi,lljk}

Contact

lazymath2024@gmail.com

## Example 2



- e. The owners of the supermarket want the project completed earlier.

They will pay to reduce the time of some of the activities.

A reduction in completion time of an activity will incur an additional cost of \$10 000 per week.

Activities can be reduced by a maximum of two weeks.

The minimum number of weeks an activity can be reduced to is seven weeks.

What is the minimum amount the owners of the supermarket will have to pay to reduce the completion time of the project as much as possible?

1 mark

Source: *VCAA 2024 General Mathematics Examination 2 Question 13*

`graph`  $\{a9,b11,c7a,d8a,e4b,f8b,g12cde,h8cde,i6de,j5hi,k8f\}$   
`crashing(graph,{a,b,d,f,h,g,k,10000},{2,2,1,1,1,2,1},\infty,"")`

►Working:

| "Path"      | "Start" | "↓2A" | "↓1B" | "↓1D" | "↓1H" |
|-------------|---------|-------|-------|-------|-------|
| "BFK"       | 27      | 27    | 26    | 26    | 26    |
| "BEIJ"      | 26      | 26    | 25    | 25    | 25    |
| "ADIJ"      | 28      | 26    | 26    | 25    | 25    |
| "BEHJ"      | 28      | 28    | 27    | 27    | 26    |
| "ADHJ"      | 30      | 28    | 28    | 27    | 26    |
| "ACHJ"      | 29      | 27    | 27    | 27    | 26    |
| "BEG"       | 27      | 27    | 26    | 26    | 26    |
| "ADG"       | 29      | 27    | 27    | 26    | 26    |
| "ACG"       | 28      | 26    | 26    | 26    | 26    |
| "□"         | "□"     | "□"   | "□"   | "□"   | "□"   |
| "Crit Time" | 30      | 28    | 28    | 27    | 26    |
| "Min Cost"  | 0       | 20000 | 30000 | 40000 | 50000 |

►Solution:

| "Cost" | "Time" | a  | b  | d  | f  | h  | g  | k  |
|--------|--------|----|----|----|----|----|----|----|
| 50000. | 26.    | 2. | 1. | 1. | 0. | 1. | 0. | 0. |

Done

`graph:=\{a9,b11,c7a,d8a,e4b,f8b,g12cde,h8cde,i6de,j5hi,k8f\}`

Contact  
lazymath2024@gmail.com

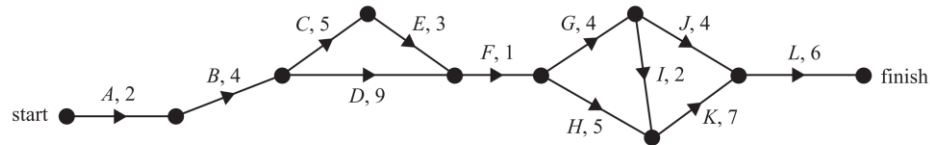
### Example 3

#### Question 3 (6 marks)

Fencedale High School is planning to renovate its gymnasium.

This project involves 12 activities,  $A$  to  $L$ .

The directed network below shows these activities and their completion times, in weeks.



The minimum completion time for the project is 35 weeks.

- e. The reduction in completion time for each of these five activities will incur an additional cost to the school.

The table below shows the five activities that can have their completion times reduced and the associated weekly cost, in dollars.

| Activity | Weekly cost (\$) |
|----------|------------------|
| $C$      | 3000             |
| $D$      | 2000             |
| $G$      | 2500             |
| $H$      | 1000             |
| $K$      | 4000             |

The completion time for each of these five activities can be reduced by a maximum of two weeks.

Fencedale High School requires the overall completion time for the renovation project to be reduced by four weeks at minimum cost.

Source: VCAA 2019 General Mathematics Examination 2 Module 2 Question 3

graph
{a2,b4a,c5b,d9b,e3c,f1ed,g4f,h5f,i2g,j4g,k7ih,l6jk}

crashing graph,
\left( \begin{array}{c} c \quad 3000 \\ d \quad 2000 \\ g \quad 2500 \\ h \quad 1000 \\ k \quad 4000 \end{array} \right), 2, \infty, -4

►Working:

| "Path"      | "Start" | "11D" | "12G" | "11H" | "11K" |
|-------------|---------|-------|-------|-------|-------|
| "ABDFHKL"   | 34      | 33    | 33    | 32    | 31    |
| "ABCEFHKL"  | 33      | 33    | 33    | 32    | 31    |
| "ABDFGIKL"  | 35      | 34    | 32    | 32    | 31    |
| "ABCEFGIKL" | 34      | 34    | 32    | 32    | 31    |
| "ABDFGJL"   | 30      | 29    | 27    | 27    | 27    |
| "ABCEFGJL"  | 29      | 29    | 27    | 27    | 27    |
| "□"         | "□"     | "□"   | "□"   | "□"   | "□"   |
| "Crit Time" | 35      | 34    | 33    | 32    | 31    |
| "Min Cost"  | 0       | 2000  | 7000  | 8000  | 12000 |

►Solution:

| "Cost" | "Time" | $c$ | $d$ | $g$ | $h$ | $k$ |
|--------|--------|-----|-----|-----|-----|-----|
| 12000. | 31.    | 0.  | 1.  | 2.  | 1.  | 1.  |

Done

$graph = \{a2, b4a, c5b, d9b, e3c, f1ed, g4f, h5f, i2g, j4g, k7ih, l6jk\}$

Contact

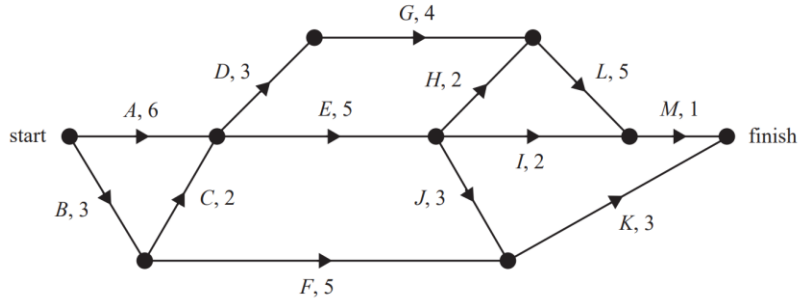
lazymath2024@gmail.com

### Example 4

#### Question 4 (3 marks)

Roadworks planned by the local council require 13 activities to be completed.

The network below shows these 13 activities and their completion times in weeks.



- c. It is possible to reduce the completion time for activities  $A, E, F, L$  and  $K$ .  
The reduction in completion time for each of these five activities will incur an additional cost.  
The table below shows the five activities that can have their completion time reduced and the associated weekly cost, in dollars.

| Activity | Weekly cost (\$) |
|----------|------------------|
| $A$      | 140 000          |
| $E$      | 100 000          |
| $F$      | 100 000          |
| $L$      | 120 000          |
| $K$      | 80 000           |

The completion time for each of these five activities can be reduced by a maximum of two weeks.

The overall completion time for the roadworks can be reduced to 16 weeks.

What is the minimum cost, in dollars, of this change in completion time?

1 mark

$graph = \{a6,b3,c2b,d3ac,e5ac,f5b,g4d,h2e,i2e,j3e,k3ff,l5gh,m1li\}$

$crashing \left( graph, \begin{bmatrix} a & 140 \\ e & 100 \\ f & 100 \\ l & 120 \\ k & 80 \end{bmatrix}, 2, \infty, 16 \right)$

►Working:

| "Path"      | "Start" | "1A"  | "1L"  |
|-------------|---------|-------|-------|
| "BCEIM"     | 13      | 13    | 13    |
| "AEIM"      | 14      | 13    | 13    |
| "BCEHLM"    | 18      | 18    | 16    |
| "AEHLM"     | 19      | 18    | 16    |
| "BCDGLM"    | 18      | 18    | 16    |
| "ADGLM"     | 19      | 18    | 16    |
| "BCEJK"     | 16      | 16    | 16    |
| "AEJK"      | 17      | 16    | 16    |
| "BFK"       | 11      | 11    | 11    |
| "[ ]"       | "[ ]"   | "[ ]" | "[ ]" |
| "Crit Time" | 19      | 18    | 16    |
| "Min Cost"  | 0       | 140   | 380   |

►Solution:

| "Cost" | "Time" | $a$ | $e$ | $f$ | $l$ | $k$ |
|--------|--------|-----|-----|-----|-----|-----|
| 380.   | 16.    | 1.  | 0.  | 0.  | 2.  | 0.  |

Done

$graph := \{a6,b3,c2b,d3ac,e5ac,f5b,g4d,h2e,i2e,j3e,k3ff,l5gh,m1li\}$

Contact

lazymath2024@gmail.com

## Flow

Determines the maximum flow through a vertex labelled network, the edges within the minimum cut, the edges which can be reversed to increase flow, as well as the minimum values of these edges required to provide a maximal increase in flow.

### Syntax

$flow(ELG, Source, Sink)$

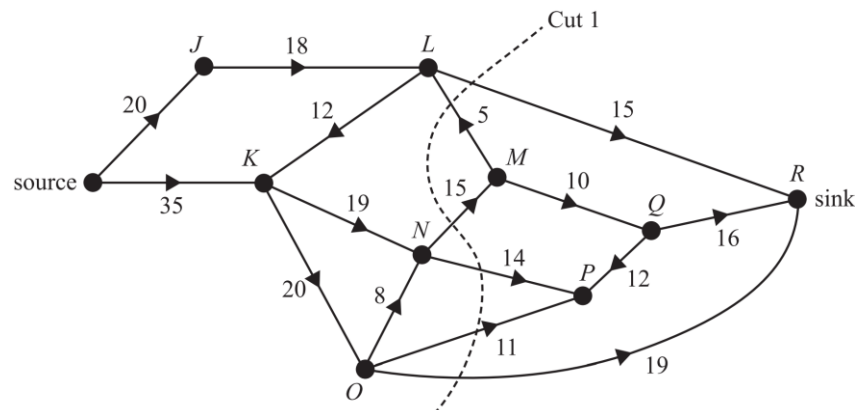
Where,

*ELG* is an edge labelled graph

*Source* is the source of the network

*Sink* is the sink of the network

### Example



- b. What is the maximum flow of stormwater, in litres per minute, from the source to the sink? 1 mark

- c. The direction of flow is reversed through one pipe.

Complete the following sentence by filling in the boxes provided. 1 mark

The pipe that should have its flow reversed to cause the largest increase in flow from source to sink is

the pipe that runs from vertex  to vertex .

### Example – Continued

- d. The capacity of one pipe is increased.

Complete the following sentences by filling in the boxes provided.

1 mark

The pipe that should have its capacity increased to cause the largest increase in flow from source to sink is the pipe that runs from vertex  to vertex . Its new capacity, in litres per minute, should be at least .

Source: VCAA 2022 Further Mathematics Written Examination 2 Question 3

*graph*  $\{sj20k35,jl18,kn19o20,lk12r15,mq10l5,nm15p14,on8p11r19,p,qr16p12,r0\}$   
*flow*(*graph*,*s*,*r*)

►Maximum Flow:

44.

►Minimum Cut:

| "Edge" | "Cap" | "New Flow" | "ΔFlow" |
|--------|-------|------------|---------|
| "LR"   | 23.   | 52.        | 8.      |
| "MQ"   | 15.   | 49.        | 5.      |
| "OR"   | 20.   | 45.        | 1.      |

►Reversible Edges:

| "Edge" | "New Flow" | "ΔFlow" |
|--------|------------|---------|
| "QP"   | 50.        | 6.      |

Done

**graph**:= $\{sj20k35,jl18,kn19o20,lk12r15,mq10l5,nm15p14,on8p11r19,p,qr16p12,r0\}$

### Note

*Cap* refers to the minimum weight the edge can have to cause a maximal increase in flow from source to sink

*New Flow* refers to the maximum new flow which can be produced by increasing the weight of the edge

*ΔFlow* refers to the increase in flow

## Path Flow

Determines the flow along each path of the graph.

### Syntax

*path\_flow(VLG, Source, Sink)*

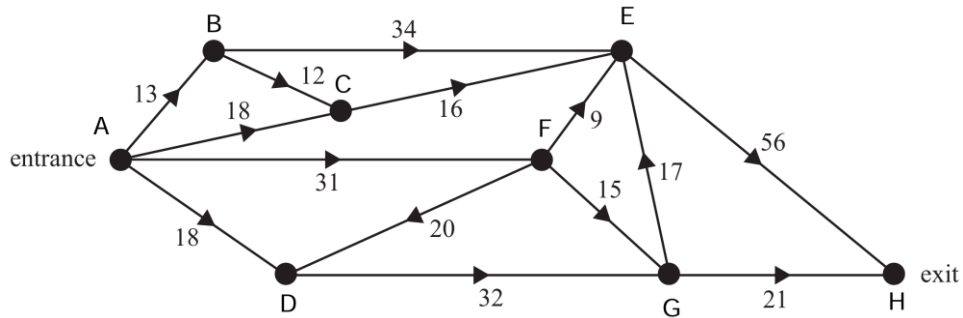
Where,

*VLG* represents a vertex labelled graph

*Source* represents the source of the network

*Sink* represents the sink of the network

### Example



Source: VCAA 2024 General Mathematics-NHT Written Examination 1 Question 38 and Question 39

Determine the path in the graph with the greatest flow.

*graph* {ab13c18d18f31,bc12e34,ce16,dg32,eh56,fe9d20g15,ge17h21,h}

*path\_flow(graph,a,h)*

| "Path"      | "Flow" |
|-------------|--------|
| "AFDGH"     | 20     |
| "ADGH"      | 18     |
| "ADGEH"     | 17     |
| "AFDGEH"    | 17     |
| "ACEH"      | 16     |
| "AFGEH"     | 15     |
| "AFGH"      | 15     |
| "ABEH"      | 13     |
| "ABCEH"     | 12     |
| "AFEH"      | 9      |
| "Total: 10" | "0"    |

Done

**graph**:= {ab13c18d18f31,bc12e34,ce16,dg32,eh56,fe9d20g15,ge17h21,h}

Contact  
lazymath2024@gmail.com

## Allocate

Determines all optimal tasks allocations which result in a minimum completion time.

### Syntax

*allocate*(*Cost Matrix*)

Where, *Cost Matrix* represents the matrix containing workers as its rows, and the time each take to complete a task in its columns

### Example

|       | Alan | Brianna | Chamath | Deidre | Ewen |
|-------|------|---------|---------|--------|------|
| Job 1 | 5    | 8       | 5       | 8      | 7    |
| Job 2 | 5    | 7       | 6       | 7      | 4    |
| Job 3 | 9    | 5       | 7       | 5      | 9    |
| Job 4 | 7    | 7       | 9       | 8      | 5    |
| Job 5 | 4    | 4       | 4       | 4      | 3    |

Source: VCAA 2016 Further Mathematics Examination 1 Section B Module 2 Question 8

Determine all allocations which allow the five jobs to be completed in the minimum amount of time.

|                    |                                                                                        |
|--------------------|----------------------------------------------------------------------------------------|
| <i>cost_matrix</i> | 5. 5. 9. 7. 4.<br>8. 7. 5. 7. 4.<br>5. 6. 7. 9. 4.<br>8. 7. 5. 8. 4.<br>7. 4. 9. 5. 3. |
|--------------------|----------------------------------------------------------------------------------------|

*allocate*(*cost\_matrix*)

►Allocation 1. :

| "Worker" | "Task" | "Duration" |
|----------|--------|------------|
| 1.       | 2.     | 5.         |
| 2.       | 3.     | 5.         |
| 3.       | 1.     | 5.         |
| 4.       | 5.     | 4.         |
| 5.       | 4.     | 5.         |

►Allocation 2. :

| "Worker" | "Task" | "Duration" |
|----------|--------|------------|
| 1.       | 2.     | 5.         |
| 2.       | 5.     | 4.         |
| 3.       | 1.     | 5.         |
| 4.       | 3.     | 5.         |
| 5.       | 4.     | 5.         |

►Minimum cost: 24.

*cost\_matrix*:=*[[5,5,9,7,4][8,7,5,7,4][5,6,7,9,4][8,7,5,8,4][7,4,9,5,3]]*

Contact

lazymath2024@gmail.com

## Hungarian Algorithm

Uses the Hungarian algorithm to determine the optimal allocations to produce the minimum cost. The rows of the matrix represent the workers, while the columns represent the tasks.

### Syntax

*hungarian(Cost Matrix)*

Where, *Cost Matrix* represents the matrix containing workers as its rows, and the time each take to complete a task in its columns

### Example

Anush, Blake, Carly, and Dexter are workers on a construction site. They are each allocated one task. The time, in hours, it takes for each worker to complete each task is shown in the table below.

|        | Task 1 | Task 2 | Task 3 | Task 4 |
|--------|--------|--------|--------|--------|
| Anush  | 12     | 8      | 16     | 9      |
| Blake  | 10     | 7      | 15     | 10     |
| Carly  | 11     | 10     | 18     | 12     |
| Dexter | 10     | 14     | 16     | 11     |

*Source: VCAA 2024 General Mathematics Written Examination 1 Question 39*

The tasks must be completed sequentially and in numerical order.

Determine the minimum time, in hours, required to complete all four tasks.

$$\text{cost\_matrix} \quad \begin{bmatrix} 12. & 8. & 16. & 9. \\ 10. & 7. & 15. & 10. \\ 11. & 10. & 18. & 12. \\ 10. & 14. & 16. & 11. \end{bmatrix}$$

## Example – Continued

*hungarian(cost\_matrix)*

► Given cost matrix:

|             |     |     |     |     |
|-------------|-----|-----|-----|-----|
| "Task"      | 1.  | 2.  | 3.  | 4.  |
| "Worker 1." | 12. | 8.  | 16. | 9.  |
| "Worker 2." | 10. | 7.  | 15. | 10. |
| "Worker 3." | 11. | 10. | 18. | 12. |
| "Worker 4." | 10. | 14. | 16. | 11. |

► Step 1: Row reduction

|             |    |    |    |    |       |
|-------------|----|----|----|----|-------|
| "Task"      | 1. | 2. | 3. | 4. | "Min" |
| "Worker 1." | 4. | 0. | 8. | 1. | 8.    |
| "Worker 2." | 3. | 0. | 8. | 3. | 7.    |
| "Worker 3." | 1. | 0. | 8. | 2. | 10.   |
| "Worker 4." | 0. | 4. | 6. | 1. | 10.   |

► Step 2: Column reduction

|             |    |    |    |    |
|-------------|----|----|----|----|
| "Task"      | 1. | 2. | 3. | 4. |
| "Worker 1." | 4. | 0. | 2. | 0. |
| "Worker 2." | 3. | 0. | 2. | 2. |
| "Worker 3." | 1. | 0. | 2. | 1. |
| "Worker 4." | 0. | 4. | 0. | 0. |
| "Min"       | 0. | 0. | 6. | 1. |

► Step 3. : Update by the min value of 1.

|             |    |    |    |    |
|-------------|----|----|----|----|
| "Task"      | 1. | 2. | 3. | 4. |
| "Worker 1." | 3. | 0. | 1. | 0. |
| "Worker 2." | 2. | 0. | 1. | 2. |
| "Worker 3." | 0. | 0. | 1. | 1. |
| "Worker 4." | 0. | 5. | 0. | 1. |

► Step 4. : Number of lines equals number of workers, ready for allocation

► Optimal allocations

|             |    |    |    |    |
|-------------|----|----|----|----|
| "Task"      | 1. | 2. | 3. | 4. |
| "Worker 1." | 0. | 0. | 0. | 1. |
| "Worker 2." | 0. | 1. | 0. | 0. |
| "Worker 3." | 1. | 0. | 0. | 0. |
| "Worker 4." | 0. | 0. | 1. | 0. |

► Step 5. : Calculate minimum cost

9.+7.+11.+16.=43.

► Summary:

|          |        |            |
|----------|--------|------------|
| "Worker" | "Task" | "Duration" |
| 1.       | 4.     | 9.         |
| 2.       | 2.     | 7.         |
| 3.       | 1.     | 11.        |
| 4.       | 3.     | 16.        |

Min Cost = 43.

*cost\_matrix:=*[[12,8,16,9][10,7,15,10][11,10,18,12][10,14,16,11]]