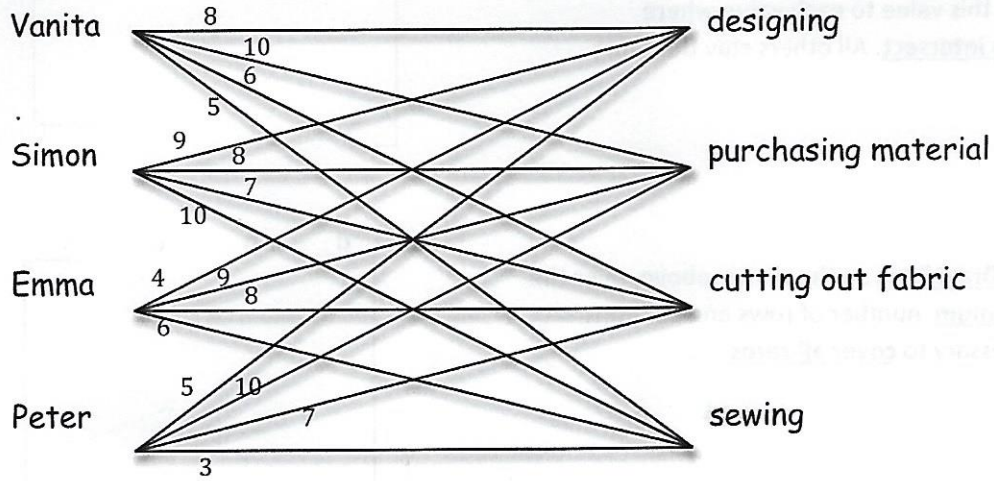


Bipartite Graphs/ Allocation problems/Hungarian Algorithm

Four students got together to design and construct their outfits for the end of year school social. The bipartite graph below shows the tasks with which each student is willing to assist. The weightings show how long each task takes each person in hours.



Vanita, Simon, Emma and Peter each want to perform one task. They want to allocate the tasks so that the overall project gets completed in the quickest time. **BUT it is not as simple as deciding who is the quickest at what.** Both Vanita and Peter are quicker at sewing than at any other task. Peter being the quicker of the two. Allocating sewing to Peter may result in Vanita taking on a task she is extremely slow at. This would delay the overall completion of the project.

Use a Hungarian Algorithm to find the allocation of tasks to keep the overall completion time to a minimum. What time did the project take?

Method

1. Represent the data as a matrix
2. Subtract the smallest number from each row

ROW REDUCTION

Draw lines in the matrix above, over the minimum number of rows and columns necessary to cover all zeros

	d	p	c	s
V	8	10		5
S	9		7	10
E	4	9	8	6
P	5	10	7	3

3. From each column select the smallest number and subtract it from all other values in that column

COLUMN REDUCTION

Draw lines in the matrix above, over the minimum number of rows and columns necessary to cover all zeros

	d	p	c	s
V				
S				
E				
P				

If the number of lines equals the number of required allocations match allocation to zeros. **ALLOCATE** if not, continue.

4. **Subtract the smallest uncovered** number from every **uncovered number**.

Add this value to each value where **lines intersect**. All others stay the same.

	d	p	c	s

5. **Draw lines** in the matrix above, over the **minimum** number of rows and columns necessary to **cover all zeros**.

	d	p	c	s

If the number of lines equals the number of required allocations match allocation to zeros. **ALLOCATE**, if not, repeat steps 4 and 5.

6. To **ALLOCATE**, look at where the zeros occur. Select one zero in each row and column and draw a box around it. Transfer these boxes to the original matrix. This is your solution!

	d	p	c	s
V	8	10	6	5
S	9	8	7	10
E	4	9	8	6
P	5	10	7	3

7. Write out the solution and calculate the total time