Input and Output format

The input for all the programs must be read from .txt files. The detailed format for each problem is given in the below.

1. (2 marks) Given a graph (0-1 adjacency matrix) and a coloring/labelling to the vertices of , verify that the coloring is valid or not.

Input: The input file (.txt format) contains rows. The first row contains a single value (positive integer) and it denotes the number of vertices in the graph . The next rows and columns represent the 0-1 adjacency matrix. The elements in each row are separated by a space. The last row contains positive integers, need not be distinct. Each distinct integer denotes a color. The integers in the last row are separated by a space and -th color, from left to right, is corresponding to -th vertex.

Output: print 1, if the coloring is valid, otherwise print 0.

1. (2 marks) Given a graph (0-1 adjacency matrix), verify is bipartite graph or not.

Input: The input file (.txt format) contains rows. The first row contains a single value (positive integer) and it denotes the number of vertices in the graph . The next rows and columns represent the 0-1 adjacency matrix. The elements in each row are separated by a space.

Output: print 1, if the graph is bipartite, otherwise print 0.

1. (2 marks) Given a bipartite graph (0-1 adjacency matrix), verify is there any complete matching from to (Hint: use Hall’s marriage theorem).

Input: The input file (.txt format) contains rows. The first row contains two values (positive integers) and they are separated by a space. The first number denotes and the second value denotes . Further, the sum of the two values denotes . The next rows and columns represent the 0-1 adjacency matrix. The elements in each row are separated by a space. Further, the first rows and columns are corresponding to the vertices in and the latter rows and columns are corresponding to the vertices in .

Output: print 1, if there is a complete matching from to , otherwise print 0.

1. (3 marks) Given a graph (0-1 adjacency matrix), find the number of connected components in using DFS method and display the vertices in each component as well.

Input: The input file (.txt format) contains rows. The first row contains a single value (positive integer) and it denote the number of vertices in the graph . . The next rows and columns represent the 0-1 adjacency matrix. The elements in each row are separated by a space.

Output: print your output in the format where denotes the number of connected components in the graph and contains the vertices in -th connected components and if are vertices in -th connected component, then is of the form .

1. (3 marks) Given a graph (0-1 adjacency matrix), and a designated root (a vertex of ), find the levels of all the nodes in the spanning tree rooted at obtained by applying BFS method.

Input: The input file (.txt format) contains rows. The first row contains a single value (positive integer) and it denote the number of vertices in the graph . The next rows and columns represent the 0-1 adjacency matrix. The elements in each row are separated by a space. The last row contains a value which denote the root . Assume that the vertex naming is .

Output: print the level of each vertex, in the order from left to right, separated by a $.

1. (3 marks) Given a weighted graph , find the cost of the minimum spanning tree using Kruskal’s algorithm.

Input: The input file (.txt format) contains rows. The first row contains a single value (positive integer) and it denote the number of vertices in the graph . The next rows and columns represent the weight matrix where is a non-negative integer and it denotes the cost of the edge and if is not an edge in the graph, then .

Output: print a single value which is the cost of MST returned by Kruskal’s algorithm.