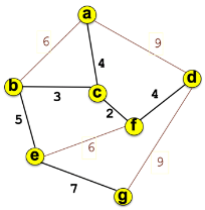
Lab. 11 - Graph Problems (2)



For graph G7 given in Lecture 10, solve the following problems:

11.1 Spanning trees

1. Use Prim's algorithm studied in the lecture to obtain the minimum spanning tree (MST) of a graph, e.g. G7, in the form of an adjacency matrix.

```
prim(G7)
```

2. What is the length of the spanning tree obtained? Use a function with signature

```
def mst_length(G):
```

3. Assume that a graph G corresponds to a distribution network, for which some redundancy is intended. For this purpose, specify a program that obtains, in the form of an adjacency matrix, the two minimal spanning trees that share no arc between them (if they do not exist, return an empty list for each non existing MST). Use function with signature:

```
def prim_2(G):
```

Hint: Obtain a MST, remove the arcs, and obtain a second MST

Note: Test your program in graph G8, obtained by adding the following arcs to G7:

<a,e> = 12; <c,d> = 7; <c,g> = 16;

11.2 Graph Distances

Use the Floyd-Warshall's algorithm to

a) Identify the most central node of the graph;

```
def most_central(G):
```

b) Identify the pair of nodes of the graph more far apart; and

```
def longest_distance(G):
```

c) Obtain the path between these nodes.

```
def longest_path(G):
```

11.2 Other Graphs

Repeat the above problems for other graphs in graphs.zip file, namely

- graph_10_60.txt,
- graph_10_90.txt,
- graph_20_60.txt,
- graph_20_90.txt
- graph_50_60.txt,
- graph_50_90.txt,
- graph_100_60.txt,
- graph_100_90.txt