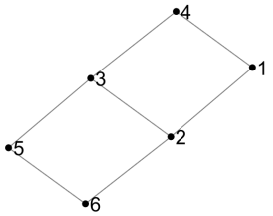
MATH.APP.270 Algorithms for graphs

Programming assignment 1 2022

In this assignment it can be assumed that one starts with a connected, undirected graph $G = (V, E)$. One such graph is the following:

$G_1 =$ 

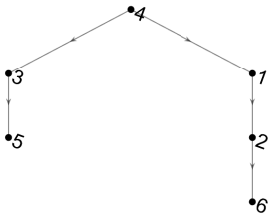
Let $s \in V$ be one particular vertex (the *starting vertex*). Let subgraph $T = (V_T, E_T)$ of graph G_1 have the following properties:

property 1 $V_T = V$ and T is a tree.

property 2 For all vertices x, the distance $d(s, x)$ for G_1 and $d(s, x)$ for T are equal.

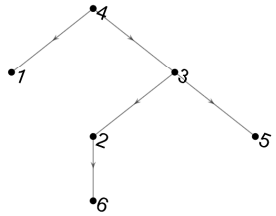
We will call a tree satisfying these properties a *minimal path spanning tree*. As long as G is connected, at least one such minimal path spanning tree will exist.

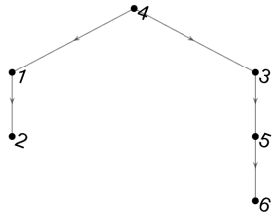
If we perform a breadth-first search (BFS) on G_1, we can obtain one particular minimal path spanning tree T. For example, for the above graph G_1 and $s = 4$, we we might obtain the following minimal path spanning tree from a BFS:

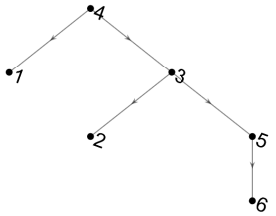
$T_1 =$ 

Even though the original graph G_1 was undirected, we have used directed edges in T_1 to emphasize that paths are being sought from vertex $s = 4$.

Graph G_1 actually has 4 minimal path spanning trees. Here are the remaining three:







Your task in this assignment is to produce and test a function (method) called `allMinSpanT` that fulfills the following specifications:

- `allMinSpanT` takes as input a graph G and a starting vertex s
- `allMinSpanT` generates all minimal path spanning trees. The tree can be generated in two different ways:

– as an edge set E_T, with all edges pointing away from the starting vertex

– as a parent function:

$$p(u) = \begin{cases} x & \text{when there is a directed edge } (x,u) \text{ in the minimal path spanning tree} \\ \infty & \text{for } u = s \end{cases}$$

Either of these formats is acceptable in other words `allMinSpanT` need not generate both formats. As an example, consider tree T_1 shown above. Its edge set is

$$E_T = \{(4,3),(4,1),(1,2),(2,6),(3,5)\}$$

and its parent function is

x	1	2	3	4	5	6
$p(x)$	4	1	4	∞	3	2

- `allMinSpanT` or some auxiliary function should store each tree in a separate file. These files should have some sequential numbering in the filename. For example, assuming 4 trees were generated, then the following filenames would be reasonable: `tree01.txt`, `tree02.txt`, `tree03.txt` and `tree04.txt`.

Strategy
It will not be possible to solve this problem simply by using a standard BFS algorithm (see exercise set 1). In a standard BFS, each vertex u, except s, is assigned a single parent $p(u)$. To obtain all minimal path spanning trees, it will be necessary to form a set of parents for each vertex (except s). Then one must form all possible combinations of parents.

Data for testing
A set of graphs for testing purposes will be published separately.