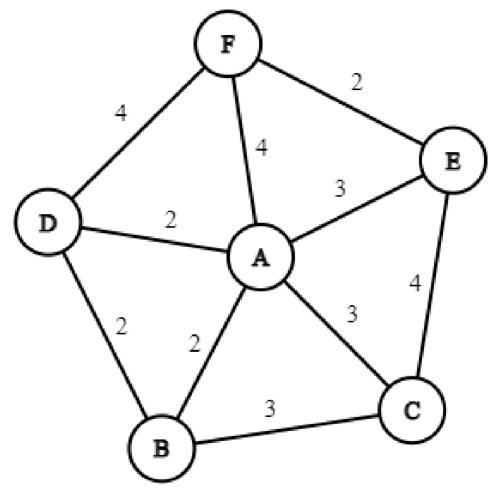
IDC – Algorithms – Homework 6

Due date: Sunday, 8 January 2022

NOTE: the general homework guidelines published in homework 1
are valid for all homework assignments in this course.

The material required to solve questions marked by ♠ will be covered next week.

Problem 1 (25 pts)



Answer the following questions (no need to justify your answers):

1. What is the weight of every MST in the graph?

   12

2. How many different MSTs does the graph have?

   5

3. Kruskal's algorithm is performed on the graph. What are the Union/Find sets after the first four edges are considered? Break ties arbitrarily.

   {B,A,D}, {E,F}, {C}

4. ♠ Assume Prim's algorithm is performed starting from vertex F. Show the best-edge array after two edges are added to the tree.

vertex	a	b	c	d	e	f
to	-	A	A	A	-	-
cost	-	2	3	2	-	-

5. ♠Will your answers to (3) or (4) change if the weight of the edge BC would be 1 instead of 3?

The answer to 4 would not change as edge BC is not considered when adding the first two edges due to the nature of Prims algorithm starting on vertex F

The answer to 3 would change as Kruskal's algorithm sort all the edges by their weights, so changing the weight of BC would change the order in which the edges are considered and more specifically will make BC the first edges to be considered. This will cause Kruskal's algorithm to produce a different result. For example B,C would be in the same component, whereas in the original answer they are not.

Problem 2 (20 pts.) (**answer in the next page**)

Let G=(V,E) be an undirected connected graph that describes the road network of the city Ramat Hod. Let $E = E_1 \cup E_2$, where $E_1 \cap E_2 = \emptyset$. The edges of E_1 correspond to roads that are maintained by Company 1, and the edges of E_2 correspond to roads that are maintained by Company 2. The mayor decides to renew some of the roads. For every e∈E, the cost of renewing road e is c(e)>0. Due to the limited budget of Ramat Hod, it is decided to upgrade only a set of roads that form a spanning tree of G. Moreover, the owner of Company 1 is a cousin of the mayor, and it is also decided to let her company renew all but at most 3 roads (out of the |V|-1 roads that will be renewed).

Suggest an efficient algorithm that decides whether this preference of Company 1 increases the cost of the project (compared to the same project of renewing a ST without a preference to Company 1).

Describe the algorithm, justify its correctness shortly, and analyze its time complexity.

Note that you don't need to determine the set of roads that will be renewed.

♠ **Problem 3** (25 pts.) (**answer in page 4**)

Given a connected, undirected graph $G = (V, E)$ with integral positive weights on the edges, and an edge e, suggest an algorithm that determines whether increasing the weight of e by one, increases the weight of the MST of G as well. The time complexity of your algorithm should be $O(|E|)$. Prove the correctness and justify the time complexity of your algorithm.

Problem 2 – solution:

Algorithm:

Run Kruskal's algorithm and sort the edges according to their weights, however whenever there is a tie between two edges in the sort favour the edge owned by Company 1 over the edge owned by Company 2. Check whether adding an edge from E2 will cause the number of edges from E2 to be more than 3, if so then don't add the edge and skip to the next one in the sorted list. Once that Kruskal's run is done, calculate the cost of the MST by adding the weights of all the edges. Now run Kruskal's algorithm on all the edges normally, with no preferences or skipping of edges and calculate the cost of the resulting MST. Compare the two costs of each MST and if the first MST is more expensive, then the preference to Company 1 increases the cost of the project.

Correctness:

Denote the first tree made by Kruskal by T1. We know by the correctness of Kruskal that this is an MST. We also know that there is at most 3 edges from E2 as it is ensured by our algorithm to skip over edges that would've made the number of edges from E2 more than 3. For the second tree made by Kruskal, T2, we know it is an MST by the correctness of Kruskal. Comparing the cost of the two trees is trivially correct.

Time complexity:

Running Kruskal twice : O(mlogm) where m is the number of edges in E

Comparing Costs : O(1)

Total time: O(mlogm)

Problem 3 – solution

Algorithm:

Remove from the graph all the edges with weight W(e) or more and call the graph G'. G' only contains edges with weight less than W(e). Assume e=(a,b). Run DFS on G' to determine if a and b are connected. If they are not connected run the algorithm again with W(e) + 1

Correctness:

The correctness of a similar algorithm was proven in the recitation to show if e is in an MST or not. Therefore running the algorithm on e+1 is correct and when compared to the answer of running the algorithm on e we can confirm if the cost of the MST has changed:

1. If a,b are connected without e then the MST value has not changed
2. If a.b are ont connected without e but are connected with e+1 then there is another vertex of W(e) that can connect a and b and is being used in the MST
3. If a.b are not connected without e and not connected without e+1 then the value of the MST has changed as e+1 has been used in the MST now

Time Complexity:

Algorithm given by adjacency list: O(m)

DFS: O(m)

Total: O(m)

Problem 3 – Extra Space (if needed)

Problem 4 (15 pts)

For each of the following claims, decide whether it is true or false and prove your answer.

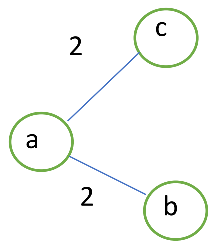
1. If two edges or more in a graph have the same weight, the graph has at least two different MST's.

2. Let G’ be a graph produced from a graph G by reducing the weight of a single edge. There exists an MST of G’ that can be produced by replacing at most one edge in an MST of G.

4a.

False:

Edges (a,b) and (a,c) have the same weight but there is only one MST



4b.

True. By reducing the weight of one edge it changes the ordering of the edges done by Kruskal’s algorithm. Let T be the MST returned by Kruskal with the original edge weights. Let the ordering of the edges in the run of Kruskal be e1,e2, … ,ek, … em. By reducing the weight of one edges, lets say edge ek, then ek is either moved to the left in the ordering or stays where it is (all other edges stay where they are). Running Kruskal on this new ordering returns an MST which is different from T by at most 1 edge, since the ordering of edges was changed by at most 1.

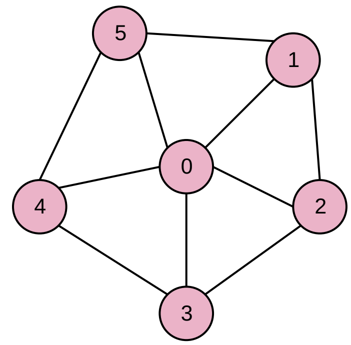
Problem 5 (15 pts)

A **wheel graph** W_n is defined as follows for every $n \geq 3$:

$V(W_n) = \{0,1,\ldots,n\}$

$E(W_n) = \{(0,i) \mid i>0\} \cup \{(i,i+1) \mid 0<i<n\} \cup \{(1,n)\}$

For example, the figure shows the wheel graph W_5.



Edges adjacent to node 0 are called chords.
The rest of the edges are called outer edges.

1. Given is a wheel graph W_n. All the chords have weight 1, all outer edges have weight 2. How many different MSTs does the graph have? Justify

There will be 1 MST, that MST being a tree consisting of all the chords (call it T). This is because to connect vertex 0 to the other edges there is 2 options, using the just the chord that connects them or using a chord to a different vertex and then an outer edge, The first options is always cheaper and therefore all other ST besides T will be more expensive

2. The following questions refer to a wheel graph W_5 in which the chord $(0,i)$ has weight i (for $i=1,\ldots,5$), and all outer edges have weight 3.

   a. The weight of any MST in this graph is $\boxed{12}$ (no need to justify).

   b. ♠ Prim's algorithm is executed on the graph. Is there an execution in which the edge (0,1) is the last to be added to the spanning tree? Explain.

   c. ♠ Prim's algorithm is executed on the graph. Is there an execution in which the edge (4,3) is the last to be added to the spanning tree? Explain.