Homework 9

2019.05.02

Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. Exercises 23.1-11

Given a graph *G* and a minimum spanning tree *T*, suppose that we decrease the weight of one of the edges not in *T*. Give an algorithm for finding the minimum spanning tree in the modified graph.

2. Exercises 23.2-5

Suppose that all edge weights in a graph are integers in the range from 1 to |V|. How fast can you make Prim's algorithm run? What if the edge weights are integers in the range from 1 to W for some constant W ?

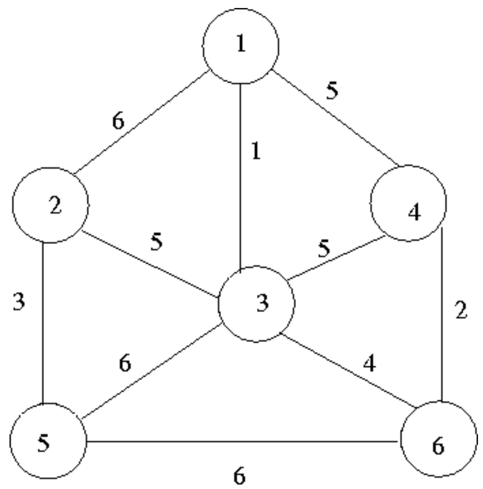
3. Exercises 23.2-7

Suppose that a graph G has a minimum spanning tree already computed. How quickly can we update the minimum spanning tree if we add a new vertex and incident edges to G ?

4. EXT. 8-1

Perform the following algorithms to the figure below.

- Kruskal's algorithm (Please list each step according to PPT unit8 p.10)
- Prim's algorithm (Please list each step according to PPT unit8 p.20)



5. EXT. 8-2

The following statements may or may not be correct. In each case, either prove it (if it is correct) or give a counterexample (if it isn't correct). Always assume that the graph $G=(V,E)$ is undirected and connected. Do not assume that edge weights are distinct unless this is specifically stated.

a. If a graph G has more than $|V|-1$ edges, and there is a unique heaviest edge, then this edge cannot be part of any MST.

b. If G has a cycle with a unique heaviest edge e, then e cannot be part of any MST.

c. Let e be any edge of minimum weight in G. Then e must be part of some MST.

d. If the lightest edge in a graph is unique, then it must be part of every MST.

e. If e is part of some MST of G, then it must be lightest edge across some cut of G.

f. If G has a cycle with a unique lightest edge e, then e must be part of every MST.

g. Prim's algorithm works correctly when there are negative edges.

h. (For any $r>0$, define an r-path to be a path whose edges all have weight $<r$.) If G contains an r-path from node s to t, then every MST of G must also contain an r-path from node s to node t.

6. Problem 16-4 Scheduling variations

Consider the following algorithm for the problem from Section 16.5 (Slide Unit06 P.26) of scheduling unit-time tasks with deadlines and penalties. Let all n time slots be initially empty, where time slot i is the unit-length slot of time that finishes at time i. We consider the tasks in order of monotonically decreasing penalty. When considering task a_j, if there exists a time slot at or before a_j's deadline d_j that is still empty, assign a_j to the latest such slot, filling it. If there is no such slot, assign task a_j to the latest of the as yet unfilled slots.

- Argue that this algorithm always gives an optimal answer.
- Use the fast disjoint-set forest presented in Section 21.3 to implement the algorithm efficiently. Assume that the set of input tasks has already been sorted into monotonically decreasing order by penalty. Please write down your pseudo-code and analyze the running time of your implementation.

7. 有關Minimum Spanning Tree，除了Kruskal's Algorithm和Prim's Algorithm外，還有第三種演算法；此演算法第一步是先將圖中每個點當作一棵樹，然後從各點分別尋找與其連接的最小權重(Weight)邊加入樹中，接著……。
請完成所有步驟（須寫出Pseudo-code）並分析此演算法。