Weekplan: Minimum Spanning Trees

Philip Bille

Reading

Introduction to Algorithms, Cormen, Rivest, Leisersons and Stein (CLRS): Chapter 23.

Exercises

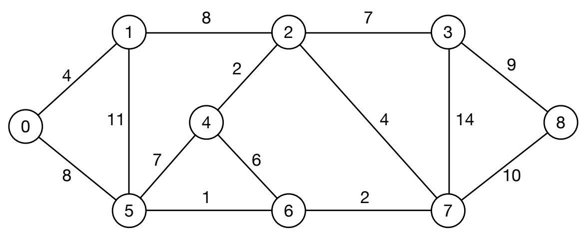


Figure 1: Graph for the exercises.

1 Algorithms and Properties Look at the graph G in Figure 1.

1.1 $[w]$ Run Kruskals algorithm on G by hand.

1.2 Run Prims algorithm on G starting in node 0 by hand. Show the contents of the priority during the execution.

1.3 Show all the minimum spanning trees of G.

1.4 CLRS 23.2-2.

1.5 Give an algorithm to find a spanning tree.

2 Reversed Deletion Consider the following algorithm to compute a MST. Start with a weighted connected graph G. Look at the edges of G in order from the heaviest to the lightest edge. For each edge determine if removal of that edge makes the graph disconnected. If it does let the edge remain, otherwise remove the edge from G.

2.1 Run the algorithm on the graph in Figure 1 by hand.

2.2 Argue why the algorithm finds a MST of G.

3 Properties of MSTs Let G be a weighted graph.

3.1 Show that the lightest edge in a graph G is in a MST for G. How about the heaviest?

3.2 Assume we scale all the edge weights in G by multiplying them with some value $c > 0$. How will MST look for the new graph?

3.3 Show that if all edge weights in G are distinct then there is a unique MST of G. *Hint:* recall the properties of MSTs.

3.4 CLRS 23.2-1

4 Maximal Spanning Tree Given a weighted graph G give an algorithm to compute a *maximal spanning tree* of G, ie. a spanning tree with a maximum total weight. *Hint:* transform the problem.

5 Decreasing the Weight of an Edge CLRS 23.1-11

6 MSTs on Graphs with Non-Distinct Weights

6.1 [BSc] Show that the cut and cycle properties are also true for graphs where the edge weights do not need to be distinct (the properties must be reformulated accordingly).

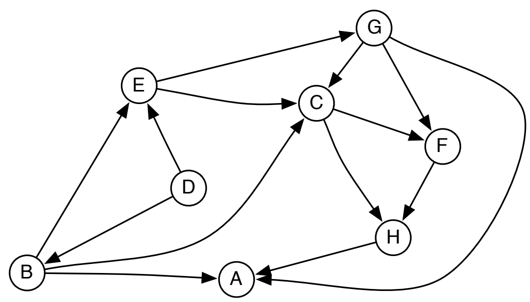
6.2 [BSc] Conclude that Prims and Kruskals algorithms also work in this case.

7 [$*$] **MSTs with Small Edge Weights** Let G be a weighted graph with n nodes and m edges such that all edge weights are values from $\{1, 2, \ldots, 10\}$. Give an efficient algorithm to compute a MST.

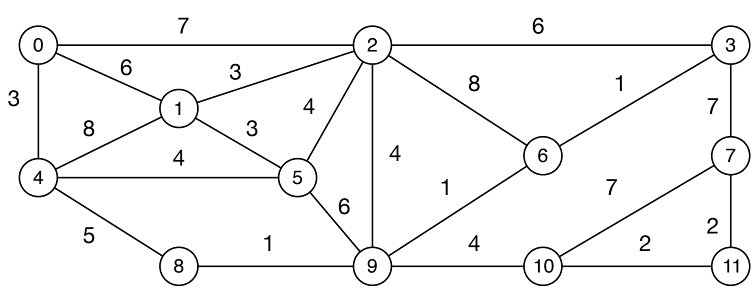
8 [$*$] **Second Best MST** CLRS 23-1

M Mandatory Exercise: Run by Hand Solve the following exercises.

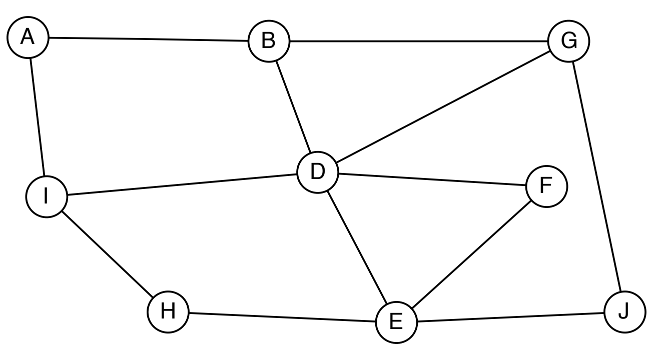
M.1 Find a topological ordering of the nodes in the following graph.



M.2 Draw a MST of the following graph and compute the total weight of the MST.



M.3 Draw a BFS-tree of the following graph. Start in node A and write the BFS layer for each node. Assume the adjacency lists are sorted in ascending order.



M.4 Draw the following max-heap after inserting an element with key 11. Draw it again after an EXTRACT-MAX operation.