Weekplan: Minimum Spanning Trees

Philip Bille Inge Li Gørtz

Reading

Introduction to Algorithms, 4th edition, Cormen, Rivest, Leisersons, and Stein (CLRS): Chapter 21.1-21.4 + *Competitive Programmer's Handbook*, Laaksonen (CSES): Chapter 15.

Exercises

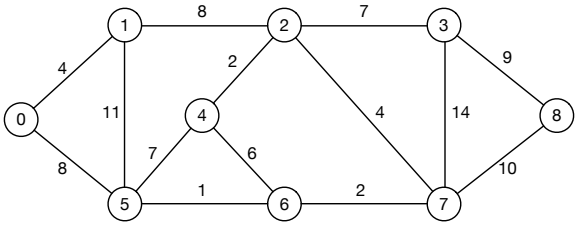

Figure 1: Graph for the exercises.

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

1.1 [w] Run Kruskal's algorithm on G by hand.

1.2 Run Prim's 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 Give an efficient algorithm to find a spanning tree.

2 [†] **Fiber Optic Cables** At the University of Algorithms there are N buildings (numbered $1, 2, \ldots, N$). Josefine is responsible for ensuring they are all interconnected with the newest fiber optic cables. Two buildings B_i and B_j can be connected by a fiber optic cable for a certain price. Josefine has been given a list of M prices for pairwise connecting two buildings (buildings not in this list, cannot be directly connected). The buildings are said to be all interconnected if and only if there for any pair of buildings is a path of fiber optic cables between the two buildings (not neccessarily direct cables). Given the list of prices, give an algorithm that can help Josefine determine the cheapest total price that ensures the buildings are all interconnected. You can assume the buildings can always be interconnected.

3 Reverse 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 disconnect the graph. If not, we remove it and otherwise leave it. Return the final set of remaining edges.

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

3.2 Argue why the algorithm finds a MST of G.

4 Properties of MSTs Let G be a weighted graph.

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

4.2 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?

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

4.4 Assume that edges are non-distinct. Kruskal's algorithm can return different MSTs for a graph G depending on how we break ties when sorting the edges. Show that for any minimum spanning tree T in G, there is a way to sort the edges in G such that Kruskal's algorithm returns T.

5 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.

6 MSTs on Graphs with Non-Distinct Weights

6.1 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 Conclude that Prim's and Kruskal's 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.