

Efficient Algorithms and Programming

Week 4

The references to the textbook in parenthesis refer to the second edition of the book.

Reading before Monday September 17th

You should have read Chapter 22 (chap. 21) excluding subsection 22.4 (21.4) and 24 (chap. 23) before Monday September 17th.

Exercises for Monday September 17th

1. Exercises 22.2-2, 22.3-1, and 22.3-2 (21.2-2, 21.3-1, 21.3-2).
2. Exercises 24.1-4, 24.1-5, 24.2-2, 24.2-3, 24.2-4 and 24.2-5 (23.1-4, 23.1-5, 23.2-2, 23.2-3, 23.2-4 and 23.2-5).

Solve the following problem:

Let $G = (V, E)$ be a connected, undirected, and weighted graph. Let H be a sub-graph of G that does not contain any cycles. Describe an efficient algorithm to find the minimum spanning tree T that includes H as a sub-graph. Give the asymptotic runtime of the algorithm.

Programming assignment for Friday September 28th

Write a program (in C, C++, Java, ML, or . . .) that implements one of the algorithms for computing minimum spanning trees. Choose *one* of the following three possible implementations:

- Kruskal's algorithm as described on p. 505.
- Prim's algorithm (p. 509) with the priority queue implemented as a binary heap.
- Prim's algorithm with the priority queue implemented as a Fibonacci heap. (You should only do this if you already have an implementation of a Fibonacci heap!)

Test your implementation on some of the graphs available from the course Web-page. Notice that the sample graphs are *directed* while the minimum spanning tree algorithms work on *undirected* graphs. So for each edge (u, v) in the file, you should insert both (u, v) and (v, u) in the graph. If both (u, v) and (v, u) exists in the file, we define the weight of the edge $\{u, v\}$ to be $\max(w(u, v), w(v, u))$. Self-loops (edges (u, u)) should be ignored.

State for each graph with n vertices the runtime and the total weight of the spanning tree. Plot the runtimes in a graph as a function of the number of vertices. Do your experiments correspond to what you would expect ?

Hand in a print-out of the program, the weights of the minimum spanning trees, the plots, and your comments to the plots.

Reading for next week

You should read Chapter 26 (chap. 25) excluding section 26.4 (this section has been eliminated in the second edition of the textbook) for Monday September 24th. The FLOYD-WARSHALL algorithm is the most important part, but you should also read about the JOHNSON algorithm since it is extremely elegant in the way it combines the algorithms BELLMAN-FORD and DIJKSTRA.