CS3230 : Tutorial - 3

Bakh Khoussainov

4-September-2012, 1pm deadline.

1. Consider Dijkstra's algorithm that solves the shortest path problem. The algorithm computes the shortest distances $d(v)$ from the start vertex to all other vertices v. Improve the algorithm so that the improved algorithm also outputs a shortest path P_v for every vertex v of the graph.

2. Consider Dijkstra's algorithm presented in the lecture. The inputs to this algorithm are directed weighted graphs with vertex set V, the edge set E, and the start vertex s. The maximum of the number of vertices and the number of edges of the graph is called the **size** of of the graph. When the algorithm is executed it performs the following basic operations:

   - The first operation, given a vertex $u \in S$ and a vertex v, detects if there is an edge from u to v.
   - The second operation computes the values $d(u) + w(u, v)$, where $w(u, v)$ is the weight of the edge from u to v.
   - The third basic operation is the comparison operation that compares the values $d'(u)$ and $d'(u')$ for given u and u'.

   Prove that, for an input graph G of size n, the total number of basic operations needed to compute the shortest path distances from s to all vertices v of G can bounded by n^3 (In fact, the bound is n but you do not need to prove that).

3. Suppose that you are given a connected weighted graph with edge wights that are all distinct. Prove that G has exactly one minimum spanning tree.

4. Suppose you are given two lists $S_1 = a_1, \ldots, a_n$ and $S_2 = b_1, \ldots, b_m$ of integers. We say that S_2 is a *weak subsequence* of S_1 if after removal of some elements from S_1 we can obtain S_2. For instance, $4, 7, 11, 33, 4, 1$ is a weak subsequence of $3, 3, 1, 4, 5, 6, 8, 7, 11, 23, 1, 4, 33, 4, 7, 1, , 3$ but not of $2, 4, 7, 3, 4, 7, 33, 4, 1$. Design an algorithm that given S_1 and S_2 detects if S_2 is a weak subsequence of S_1. In addition, do the following:

   (a) Explain if your algorithm is greedy.

   (b) Prove that your algorithm is correct.

5. There is a long straight country road with houses scattered very sparsely along the road. All residents of the houses use cell-phones. We would like to place cell-phone base stations at certain points along the road. Each station can service houses at 4km radius.

   (a) Design a greedy algorithm that achieves this goal, using as few base stations as possible.

   (b) Explain as to why your algorithm is correct.

6. Let G be a weighted graph (all weights are non-negative). Without any reference to the algorithms that produce the minimum spanning tree for G, give an independent argument as to why minimum spanning trees exist for the graph G.

7. Consider the algorithm (presented in the lecture) that merges two sorted lists $A = a_1, \ldots, a_n$ and $B = b_1, \ldots, b_m$. Here n is called the size of A, m is the size of B and $\max\{n, m\}$ is the **size** of the input (A, B). The basic operation, when the algorithm runs, is the comparison operation. For instance, at the initial stage a_1 and b_1 are compared and the smallest of these is put into list C that the algorithm is building. How many basic operations are needed in order to build the sorted array C as the output of the algorithm. Your answer should be in terms of the size of the input.