CSE548/AMS542 Fall 2008 Analysis of Algorithms

Jie Gao*

September 16, 2008

Due **October 2nd** before class. Each problem, unless specified otherwise, has a maximum of 10 points. (i) You write down the solution clearly. If we can not recognize your writing then you may lose points. (ii) Avoid too many details. A succinct and clean proof is the best. You may use the algorithms we covered in class without referring to the details. (iii) If you discuss some of the problems with another fellow student (at most 2 students per group), write down his/her name and the problems. If you consult any books/webpages, cite them.

Homework 2

1. **Scheduling to minimize average completion time.** Given a set of n tasks, where task i requires t_i amount of time to finish once it is started, you are asked to schedule the tasks on a computer such that the average completion time is minimized. The computer can only run one task at a time. Suppose you can start the first task at time 0 and all the tasks are available all at once at time 0. Once a task is started it will run until it is completed. The average completion time is
$$\sum_{i=1}^{n} f_i/n,$$
where f_i is the completion time for task i. You need to decide in which order should the computer work on the tasks. Give a greedy algorithm to solve this problem and analyze its running time.

2. The same problem as above with the following modifications: each task has a release time s_i, before which it is not available to be processed. However, you can suspend a task and restart it later.

3. **Reliable path.** A wireless network has n wireless nodes with directed edges between them. This network can be represented by a directed graph G. An edge from u to v has an associated failure probability $0 \leq p(u, v) \leq 1$ that specifies the probability that a transmission fails. Find between two given vertices x, y the most reliable path. The reliability of a path is the probability that a message from x will reach the destination y. Note that on each link there is only one chance and no retransmission is possible.

4. **Spanner.** The minimum spanning tree problem is to find a subgraph G' of a connected weighted undirected graph G, such that G' is connected with minimum total cost. Now we look at a similar problem in which we ask for a subgraph G' of G such that

*Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA, jgao@cs.sunysb.edu.

- For each pair of nodes u, v, the shortest path length in G' is at most α times the shortest path length in G. Such a subgraph is called a spanner with stretch factor α.
- G' is sparser than G (i.e., it has fewer number of edges).

Here is an algorithm that Alice designed. Assume that all the edges have distinct positive weights. The weight of an edge u, v is denoted as $w(u, v)$.

(a) Sort all the edges in the order of increasing length.

(b) Start with G' of n vertices and no edges. Consider each edge $e = (u, v)$ in the order of increasing length: if u, v are in different connected components then e is included in G' (this is what Kruskal's algorithm would do as well); if u, v are in the same connected component but the length of the shortest path in the *current* graph G' is greater than $3 \cdot w(u, v)$, then add this edge to G'. For other cases, do not add e.

Prove the following claims:

(a) The graph G' obtained at the end of Alice's algorithm is a spanner with stretch factor 3. (10pts)

(b) **Challenging question**: the number of edges in G' is $o(n^2)$. (10pts)

5. **Colored spanning tree.** Given a graph G with its edges colored either red or blue, give a polynomial algorithm that either (1) returns a spanning tree of G with exactly k red edges; or (2) reports that such a tree does not exist. The running time of the algorithm can not have k on the exponent.