KTH Computer Science
and Communication

Homework I, Approximation Algorithms 2010

Due on Tuesday October 12 at 13.15 (hand in at start of lecture or send an email to osven@kth.se). Solutions to many homework problems, including problems on this set, are available on the Internet, either in exactly the same formulation or with some minor perturbation. It is *not acceptable* to copy such solutions. It is hard to make strict rules on what information from the Internet you may use and hence whenever in doubt contact Ola Svensson. You are, however, allowed to discuss problems in groups with up to three students, but solutions should be handed in individually. On this problem set the first problem is about finding information and here you should feel free to use any source.

1 (30 p) In the introductory lecture, we saw that the symmetric traveling salesman problem with the triangle inequality admits a 2-approximation algorithm (by constructing a tour from a minimum spanning tree). In 1976, Christofides improved upon this by giving a $3/2$-approximation algorithm which is until this date the best known. Explain this algorithm and its analysis using your own words. (You are allowed to use any source but are not allowed to copy the solution literally.)

2 (30 p) *Maximum Coverage* is the following problem. Given a universal set U of n elements, with non-negative weights specified, a collection of subsets of U, $S_1, \ldots, S_\ell$, and an integer k, pick k sets so as to maximize the weight of elements covered.

Show that the obvious algorithm, of greedily picking the best set in each iteration until k sets are picked, achieves an approximation factor of

$$1 - \left(1 - \frac{1}{k}\right)^k > 1 - \frac{1}{e}.$$

3 (40 p) The tight example for the 2-approximation algorithm for Minimum Makespan Scheduling suggested sorting the jobs by decreasing processing times before scheduling them.

3a (20 p) Show that this leads to a $4/3$-approximation algorithm.

Hint: If we let j be the job that completes last in the schedule returned by the greedy algorithm, then the analysis gets easier if we distinguish the two cases when $p_j > \frac{1}{3}OPT$ and $p_j \leq \frac{1}{3}OPT$.

3b (20 p) Provide a tight example for this algorithm. More specifically, show that for any $\epsilon > 0$ there is an instance where the algorithm returns a solution with value at least a factor $(4/3 - \epsilon)$ away from optimal.