CSE548/AMS542 Spring 2008 Analysis of Algorithms

Jie Gao*

March 9, 2008

Due **March 11th** 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 3

1. Solving recurrences. Find the asymptotic order of the following recurrence, represented in big-Θ notation. (Each subproblem is 5 pts; the last problem is an extra credit problem.)
   (a) $A(n) = 4A(\lfloor n/2 \rfloor + 5) + n^2$
   (b) $B(n) = B(n-4) + 1/n + 5/(n^2+6) + 7n^2/(3n^3+8)$
   (c) (extra 5 credit) $C(n) = n + 2\sqrt{n}C(\sqrt{n})$
2. When multiple edges have the same weight, it is possible that there are more than 1 minimum spanning trees (with the same minimum weight but different combinatorial structures). Now, let's say that no three edges have the same length (but two edges can have the same length).
   (a) How many different MSTs can you have in the worst case? Give an example to show that. (5pt)
   (b) How fast can you compute *all* minimum spanning trees? Give an algorithm with the best running time you can find. (10pt)
3. Bob is computing the minimum spanning tree of a graph G and gets a tree T, but later some of the edges will have their weights changed. Now he would like to update his MST as fast as possible with the corrections to the edge weights. There are k edges with their weights updated, these edges involve ℓ vertices, give algorithms with the best running time you can find to update the MST. (Hint: do a case analysis on the values of ℓ and k, compared with n and m, for different cases you may want to use different algorithms.) (10pt)
4. In class we covered three algorithms to compute the minimum spanning tree, the Prim's algorithm and the Kruskal's algorithm both run in time $O(m \log n)$. The third algorithm, examining the edges in decreasing weight, and removing edges if they are the longest edge on a cycle, can also output a minimum spanning tree. Given an implementation of the algorithm with the best running time you can find. (10pt)

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

5. A bottleneck spanning tree T of an undirected graph G is a spanning tree of G whose largest edge weight is minimum over all spanning trees of G. We say that the value of the bottleneck spanning tree is the weight of the maximum-weight edge in T.

   (a) Argue that a minimum spanning tree is a bottleneck spanning tree. (5pt)

   (b) Find an efficient algorithm to compute a bottleneck spanning tree. (5pt) If you have an $O(m)$ running time algorithm, then you get extra 5 points.