

# CS180 Homework 3

Due: 11:59 pm PST, May 24 (No late submission is allowed)

Note that for all the questions below we assume  $m$ (number of edges)  $\geq n$ (number of nodes) so you don't need to distinguish between  $O(m)$  and  $O(m + n)$ .

1. (25 pt) Given an undirected weighted graph  $G$  with  $n$  nodes and  $m$  edges, and we have used Prim algorithm to construct a minimum spanning tree  $T$ . Suppose the weight of one of the tree edge  $((u, v) \in T)$  is changed from  $w$  to  $w'$ , design an algorithm to verify whether  $T$  is still a minimum spanning tree. Your algorithm should run in  $O(m)$  time, and explain why your algorithm is correct. You can assume all the weights are distinct.
2. (21 pt) Give the time complexity for the following divide-and-conquer algorithms:
  - Algorithm A solves the problem by dividing it into 9 subproblems of one third the size, recursively solves each subproblem, and then combines the solutions in quadratic time.
  - Algorithm B solves the problem of size  $n$  by recursively solving two subproblems of size  $n - 1$ , and then combine the solutions in constant time.
  - Algorithm C solves the problem by dividing it into 5 subproblems of half the size, recursively solves each subproblem, and then combines the solutions in linear time.
3. (25pt) Consider an  $n$ -node complete binary tree  $T$ , where  $n = 2^d - 1$  for some  $d$ . Each node  $v$  of  $T$  is labeled with a real number  $x_v$ . You may assume that the real numbers labeling the nodes are all distinct. A node  $v$  of  $T$  is a local minimum if the label  $x_v$  is less than the label  $x_w$  for all nodes  $w$  that are joined to  $v$  by an edge.

You are given such a complete binary tree  $T$ , but the labeling is only specified in the following implicit way: for each node  $v$ , you can determine the value  $x_v$  by probing the node  $v$ . Show how to find a local minimum of  $T$  using only  $O(\log n)$  probes to the nodes of  $T$ . Explain why your algorithm is correct.
4. (29pt) Given a list of intervals  $[s_i, f_i]$  for  $i = 1, \dots, n$ , please design a divide-and-conquer algorithm that returns the length of the largest overlap among all pairs of intervals. The algorithm should run in  $O(n \log n)$  time. For example, if  $n = 3$ , the intervals are  $\{[1, 6], [2, 7], [6, 8]\}$ , then the length of the largest overlap is 4. (Hint: sort the intervals by  $s_i$  values, and divide them into two parts based on their  $s_i$  values).

---

★ Homework assignments are due on the exact time indicated. Please submit your homework using the Gradescope system. Email attachments or other electronic delivery methods are not acceptable. To learn how to use Gradescope, you can:

- 1. Watch the one-minute video with complete instructions from here:  
<https://www.youtube.com/watch?v=-wemzmvGPfg>
- 2. Follow the instructions to generate a PDF scan of the assignments:  
[http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting\\_hw\\_guide.pdf](http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf)
- 3. **Make sure you start each problem on a new page.**

★ We recommend to use  $\text{\LaTeX}$ ,  $\text{\LyX}$  or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receive fewer points.

★ Unless specified, you should justify your algorithm with proof of correctness and time complexity.