“CS 374” Spring 2015 — Homework 8

Due Tuesday, April 7th, 2015 at 10am

• • • Some important course policies • • •

- **You may work in groups of up to three people.** However, each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the names and NetIDs of each person contributing.
- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use. See the academic integrity policies on the course web site for more details.
- **Submit your pdf solutions in Moodle.** See instructions on the course website and submit a separate pdf for each problem. Ideally, your solutions should be typeset in LaTex. If you hand write your homework make sure that the pdf scan is easy to read. Illegible scans will receive no points.
- **Avoid the Three Deadly Sins!** There are a few dangerous writing (and thinking) habits that will trigger an automatic zero on any homework or exam problem. Yes, we are completely serious.
  - Give complete solutions, not just examples.
  - Declare all your variables.
  - Never use weak induction.
- Unlike previous editions of this and other theory courses we are not using the “I don’t know” policy.

See the course web site for more information.

If you have any questions about these policies,
please don’t hesitate to ask in class, in office hours, or on Piazza.

---

1. In the lecture on greedy algorithms we saw an optimum algorithm for the problem of coloring a given set of intervals with as few colors as possible such that intervals colored with the same color do not conflict. Consider now an alternate recursive algorithm. Let $\mathscr{I}$ be the given set of intervals and let I_1 be an interval that has the leftmost right end point. Recursively color intervals in $\mathscr{I} \setminus I_1$ and color I_1 with the smallest numbered color that can be used given the coloring for $\mathscr{I} \setminus I_1$.
   - Write down the pseudocode for the algorithm.
   - Either prove that the algorithm outputs the optimum coloring on all instances or give a counter example.

2. Let $G = (V, E)$ be an undirected graph with edge weights $c(e), e \in E$. Let $A \subset V$ be a set of special nodes. We wish to build a minimum cost spanning tree T of G with the additional constraint that all the nodes in A are leaves of T. Call a tree T whose leaves contain A an "A-leaf-respecting" tree. Describe an algorithm that given G, c and A either outputs that there is no A-leaf-respecting spanning tree in G or outputs the cheapest A-leaf-respecting spanning tree. The cost of the tree is simply the sum of the costs of the edges of the tree. Note that if $A = \emptyset$ we are seeking to find the MST of the graph.

3. Let $G = (V, E)$ an undirected edge-weighted graph. The bottleneck weight of a spanning tree T is the weight of the maximum weight edge in T.
   - Prove that the minimum spanning tree of a graph is also a spanning tree which minimizes the bottleneck weight.
   - Describe an algorithm to compute a spanning tree with minimum bottleneck weight in linear time (that is, $O(|V| + |E|)$ time). No proof of correctness necessary but you need to justify the running time. *Hint:* Start by computing the median of the edge weights.