Sample Final Exam on

CMPT-307 Data Structures and Algorithms

1. What is an augmenting path in a flow network?

2. Consider a firm that trades shares in n different companies. For each pair $i \neq j$, they maintain a trade ratio r_{ij}, meaning that one share of i trades for r_{ij} shares of j. Here we allow the rate to be fractional; that is $r_{ij} = 2/3$ means that you can trade 3 shares of i for 2 shares of j.

   A *trading cycle* for a sequence of shares $i_1, i_2, \ldots, i_k$ consists of successively trading shares in company i_1 for shares in company i_2, then shares in company i_2 for shares i_3, and so on, finally trading shares in i_k back to shares in company i_1. After such a sequence of trades, one ends up with shares in the same company i_1 that one starts with. Trading around a cycle is usually a bad idea, as you tend to end up with fewer shares than you started with. But occasionally, for short periods of time, there are opportunities to increase shares. We will call such a cycle an *opportunity cycle*, if trading along the cycle increases the number of shares. This happens exactly if the product of the ratios along the cycle is above 1. In analyzing the state of the market, a firm engaged in trading would like to know if there are any opportunity cycles.

   Give an efficient algorithm that finds such an opportunity cycle, if one exists.

3. Explain (or give a pseudocode) the algorithm inserting a new element into a heap.

4. Suppose that we are given a key k to search for in a hash table with positions $0, 1, \ldots, m-1$, and suppose that we have a hash function h that maps the key space into the set $\{0, 1, \ldots, m-1\}$. The search scheme is as follows.
   (a) Compute the value $i := h(k)$, and set $j := 0$;
   (b) Probe in position i for the desired key k. If you find it, or if this position is empty, terminate the search
   (c) Set $j := (j+1) \pmod{m}$ and $i := (i+j) \pmod{m}$, and return to the previous step.
   Assume that m is a power of 2.

   Show that this scheme is an instance of the general quadratic probing scheme by exhibiting the appropriate constants c, d for the corresponding equation

5. How can we save memory in dynamic programming algorithms? Consider the Shortest Path problem as an example.

6. Let $G = (V, E)$ be an undirected graph with costs $c_e \geq 0$ on the edges $e \in E$. Assume you are given a minimum cost spanning tree T of G. Now assume that a new edge is added to G, connecting two nodes $v, w \in V$ with cost c.

(a) Give an efficient algorithm to test if T remains the minimum cost spanning tree with the new edge added to G (but not to the tree T). Make your algorithm run in time $O(|E|)$. Can you do it in $O(|V|)$ time.

(b) Suppose T is no longer a minimum cost spanning tree. Give a linear time algorithm (time $O(|E|)$ to update the tree T to a new minimum cost spanning tree.

7. Describe the Depth First Search algorithm.

8. 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 s_w for all nodes w that connected 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.

9. What is RB-tree?

10. Show that there is no comparison sort algorithm whose running time is linear for at least half of the $n!$ possible inputs. What about a fraction of $1/n$ of the inputs of length n? What about a fraction $1/2^n$?

11. Prove that 3-Coloring is NP-complete.

12. We explore the issue of truthfulness in the Stable Matching problem and specifically in the Gale-Shapley algorithm. The basic question is: Can a man or a woman end up better off lying about his or her preferences? More concretely, we suppose each participant has a true preference order. Consider a woman w. Suppose w prefers man m to m', but both m and m' are low on her list of preferences. Can it be the case that by switching the order of m and m' on her list of preferences (i.e. by falsely claiming that she prefers m' to m) and running the algorithm with this false preference list, w will end up with a man m'' that she truly prefers to both m and m', but would not end up with using the true list of preferences?

    Resolve this question by doing one of the following two things:

    (a) Give a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a woman's partner in the Gale-Shapley algorithm; or

    (b) Give an example of a set of preference lists for which there is a switch that would improve the partner of a woman who switched preferences.

13. Suppose that you have a "black box" worst-case linear time median subrouting. Give a simple, linear time algorithm that solves the k-Smallest problem for an arbitrary k.