G22.3520: Honors Analysis of Algorithms

Final Exam, Dec 15, 2008, 1:00-4:30pm

- This is a three and a half hour exam.
- There are six questions, worth 10 points each. Answer all questions.
- Please print your name and SID on the front of the envelope only (not on the exam booklets).
- Please answer each question in a separate booklet, and number each booklet according to the question.
- Read the questions carefully. Keep your answers legible, and brief but precise.
- Assume standard results, unless asked otherwise.

Best of luck!

Problem 1

These questions concern a binary search tree T in which each node x has a integer key value denoted $\mathsf{key}[x]$. Recall that a binary search tree is a rooted tree, each node has at most two children, and satisfies: (i) if y is a left child of x, then $\mathsf{key}[y] < \mathsf{key}[x]$. (ii) if z is a right child of x, then $\mathsf{key}[x] < \mathsf{key}[z]$. Recall also that the height is the longest distance from the root to any leaf (if the tree consists of just the root, then its height is zero). Let h be the height of T.

1. Suppose T has 5000 nodes. What is the largest its height h can be?
2. Suppose T has 1023 nodes. What is the smallest its height h can be?
3. Describe in words how to find the maximum element of a binary search tree.
4. Describe in words how to add to the tree a new node x with $\mathsf{key}[x] = a$. How long does this algorithm take as a function of the height h.
5. Describe in words how to delete a node x with $\mathsf{key}[x] = a$. How long does this algorithm take as a function of the height h.

Problem 2

Recall that a boolean variable x takes value in $\{\mathsf{True}, \mathsf{False}\}$. A literal is either a variable or its negation, i.e. x or $\overline{x}$. A clause is a logical OR of one or more distinct literals.

An instance of $(2,3)$-SAT problem consists of n boolean variables $\{x_1, x_2, \ldots, x_n\}$ and a set of m clauses $\{C_1, C_2, \ldots, C_m\}$ where each clause has either two or three distinct literals. Assume that a variable and its negation do not appear in the same clause.

1. Show that there exists a $\{\mathsf{True}, \mathsf{False}\}$-assignment to the variables $\{x_1, x_2, \ldots, x_n\}$ that satisfies at least $\frac{3}{4}m$ clauses.
2. Now assume that $\frac{m}{3}$ of the clauses have two literals and the remaining $\frac{2}{3}m$ clauses have three literals. Find a constant c, as large as you can, such that every such instance must have an assignment that satisfies $c \cdot m$ clauses. Prove your claim.

Problem 3

You are given an n-node undirected connected graph $G(V, E)$ (as adjacency list representation). Let $v_0 \in V$ be a special start node. For every node $w \in V$, let $d(v_0, w)$ denote the distance of w from v_0, i.e. the length of the shortest path from v_0 to w.

1. Name one well-known algorithm that determines $d(v_0, w)$ for all nodes w. What is its running time? It is not necessary to describe the algorithm.

2. Now suppose that you are given a node $t \in V$ and guaranteed that $d(v_0, t) > \frac{n}{2}$. Show that there must exist a node x such that after removing the node x (along with all edges incident on it) from the graph, there is no path from v_0 to t. Give $O(|V| + |E|)$ time algorithm to find such a node x.

Problem 4

Let Σ be a fixed finite alphabet. You are given a directed graph $G(V, E)$ along with a label $\sigma(u, v) \in \Sigma$ for every edge $(u, v) \in E$. The label of any directed path is defined to be the string obtained by concatenating the labels of edges along the path. You are also given a string $s \in \Sigma^*$, say $s = a_1 a_2 \dots a_k$ with $a_i \in \Sigma$ for $1 \leq i \leq k$. Let $v_0 \in V$ be a special start node. Assume adjacency list representation for the graph.

1. Give an algorithm to find a path that starts at v_0 and has the string s as its label, if such a path exists. The algorithm should detect if no such path exists. The algorithm should run in time polynomial in $|V| = n$, $|E| = m$ and $|s| = k$.

   Hint: For $1 \leq i \leq k$, define a set W_i to be the set of nodes that can be reached from v_0 by a path labeled $a_1 a_2 \dots a_i$.

2. Now suppose that every edge also has a weight $w(u, v) \geq 0$. The weight of a path is defined to be the sum of the weights of the edges on this path. Modify your algorithm so that it finds a path that starts at v_0, has the string s as its label, and has maximum possible weight among all such paths (if one exists).

Problem 5

Let the alphabet be $\Sigma = \{0, 1, \#\}$. Consider the language:

$$L = \{x\#y \mid x, y \in \{0,1\}^*, x \neq y\}.$$

1. Show that L is a context free language (*Hint: Grammar or a Push Down Automaton?*).

   The goal of the next two parts is to prove that L is not regular by showing that L cannot be accepted by a DFA with fewer than k states for any fixed integer $k \geq 1$.

2. Give a construction of k strings $x_1, x_2, \ldots, x_k \in \{0,1\}^*$ such that for every $1 \leq i < j \leq k$, there exists a string $z_{ij} \in \{0,1\}^*$ such that

$$x_i\#z_{ij} \notin L \qquad \text{and} \qquad x_j\#z_{ij} \in L.$$

3. Show that L cannot be accepted by a DFA with fewer than k states.

   Hint: Look at the state reached by the DFA after reading the string $x_i, 1 \leq i \leq k$.

Problem 6

Let $G(V, E)$ be an undirected graph. A set of nodes $S \subseteq V$ is called *triangle-free* if for any three distinct nodes $a, b, c \in S$, at least one of $\{a, b\}, \{b, c\}, \{a, c\}$ is not an edge. Let

$$\textsf{TRIANGLE-FREE-SET} = \{\langle G, k\rangle \mid 1 \leq k \leq |G|,\ G \text{ has a triangle free set of size } k\}.$$

Show that TRIANGLE-FREE-SET is NP-complete by giving a reduction from INDEPENDENT-SET problem. (*Hint: Add new nodes.)*

Recall: *A set $S' \subseteq V'$ in a graph $G'(V', E')$ is called independent if for any two distinct nodes $a', b' \in S'$, $\{a', b'\} \notin E'$.*

$$\textsf{INDEPENDENT-SET} = \{\langle G', k'\rangle \mid 1 \leq k' \leq |G'|,\ G' \textit{ has an independent set of size } k'\}.$$

You can assume that INDEPENDENT-SET *is NP-complete.*