CS60003 Algorithm Design and Analysis, Autumn 2009–10

End-Semester Test

Maximum marks: 50 | November 24, 2009 | Total time: 3 hours

Roll no: ____________________ **Name:** ____

[*Write your answers in the question paper itself. Be brief and precise. Answer* *all* *questions.*]

Guided Thinking Section

In this section, I guide you to arrive at solutions to some computational exercises.
Proceed exactly as I tell you to. Just fill out the missing details.

---

1. Let S be a finite set, and $S_1, S_2, \ldots, S_k$ a collection of subsets of S. A subcollection $S_{i_1}, S_{i_2}, \ldots, S_{i_l}$ with $1 \leqslant i_1 < i_2 < \cdots < i_l \leqslant k$ is called a *cover* of S if $S = \bigcup_{j=1}^{l} S_{i_j}$. In this case, l (the number of subsets in the cover) is called the *size* of the cover. The decision problem SET-COVER takes as input a set S, a collection $S_1, S_2, \ldots, S_k$ of subsets of S, and a positive integer l, and decides whether S has a cover (in the given subsets) of size exactly l. In this exercise, we prove that SET-COVER is an NP-Complete problem. You may assume any standard representation of sets (such as sorted/unsorted arrays, linked lists, or trees).

(a) What is the output of SET-COVER for the following input? $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ with five subsets $S_1 = \{2, 3, 5, 7\}$, $S_2 = \{1, 2, 3, 5, 8\}$, $S_3 = \{1, 2, 4, 8\}$, $S_4 = \{3, 6, 9\}$, and $S_5 = \{4, 6, 8, 9\}$, and $l = 2$. **(1)**

____

(b) Show that SET-COVER $\in$ NP. For an instance $\langle S, (S_1, S_2, \ldots, S_k), l\rangle$ in Accept(SET-COVER), a certificate is: **(2)**

This certificate can be verified in polynomial time as: **(2)**

(c) In order to prove the NP-hardness of SET-COVER, reduce VERTEX-COVER to SET-COVER. Let $\langle G, t\rangle$ be an input for VERTEX-COVER, where $G = (V, E)$ is an undirected graph with $n = |V|$ vertices and $e = |E|$ edges. The reduction algorithm produces an instance $\langle S, (S_1, S_2, \ldots, S_k), l\rangle$ for SET-COVER, where: **(4)**

S = ____

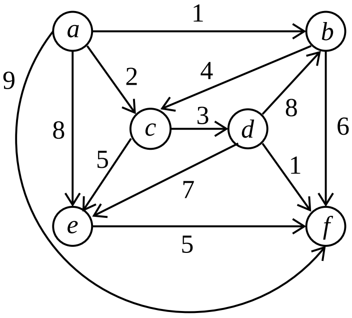
k = ____

S_i = ____

l = ____

Remark: Your reduction must fulfill the following requirement: S has a set cover of size l if and only if G has a vertex cover of size t.

2. Dijkstra's single-source-shortest-path algorithm is applied to the following graph with source a.



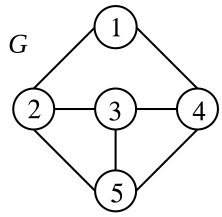
(a) Let us use the notations given in the notes. Fill out the following table to demonstrate how the partition P, Q and the arrays **`D`** and **`prev`** change in different iterations of Dijkstra's algorithm. Assume that these arrays are indexed by a, b, c, d, e, f from left to right. **(6)**

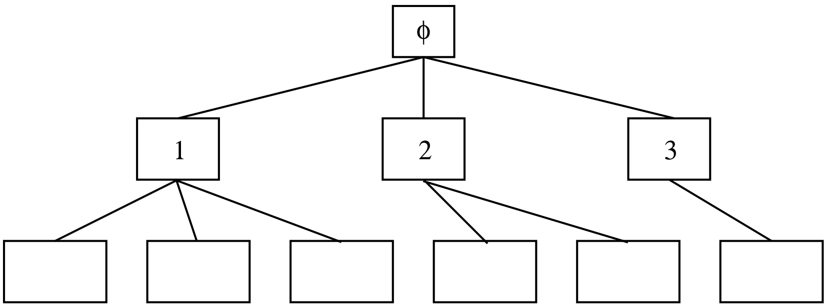
Iteration	P	Q	**`D`**	**`prev`**
Init				
1				
2				
3				
4				
5				

(b) Using the **`prev`** array, trace the shortest path from a to f. **(2)**

3. Consider the following non-deterministic algorithm for the CLIQUE problem. Let $G = (V, E)$ be an undirected graph with n vertices numbered as $1, 2, 3, \ldots, n$. We are required to find out whether G contains a clique of size t. We non-deterministically choose t vertices $v_1, v_2, \ldots, v_t$. In order to avoid repetitions, we choose the vertices in increasing order, that is, satisfying $1 \leqslant v_1 < v_2 < \cdots < v_t \leqslant n$.

(a) The following figure shows an incomplete non-deterministic computation tree for a graph G on 5 vertices and for $t = 3$. Complete the drawing. That is, draw the complete tree with each node labeled by appropriate non-deterministic choices and with leaf nodes marked additionally by Yes/No decisions. **(4)**





(b) Suppose that a backtracking algorithm is carried out on a computation tree for the above algorithm for CLIQUE. Describe a pruning strategy to identify appropriate intermediate nodes as dead ends. **(2)**

(c) Mark/state which non-leaf nodes in the tree of Part (a) are dead ends (for your pruning strategy). **(2)**

Independent Thinking Section

In this section, I supply you no guidelines.
You yourself are required to arrive at solutions to some computational exercises.

---

4. Let $P_1, P_2, \ldots, P_n$ be n points in the plane in general position. Denote by m_{ij} the slope of the segment P_iP_j. Supply an $O(n \log n)$-time algorithm for identifying the pair of points P_i, P_j for which $|m_{i,j}|$ is maximum. (In this case, P_iP_j is the steepest among all the line segments connecting the given points.) **(5)**

5. Let $\mathrm{IS\text{-}HAM\text{-}CYCLE}$ denote the computational problem that, given an undirected graph G, decides whether G contains just those edges necessary to form a Hamiltonian cycle in G (no more, no less). Prove or disprove: $\mathrm{IS\text{-}HAM\text{-}CYCLE}$ is $\mathrm{NP\text{-}Complete}$. **(5)**

6. Consider the optimization version of the set covering problem of Exercise 1. That is, given a finite set S and a collection of k subsets $S_1, S_2, \ldots, S_k$ of S, we intend to find out a cover of S (from the given collection) of size as small as possible. Let us denote this optimization problem by $\mathrm{MIN\text{-}SET\text{-}COVER}$.

(a) For instance, take $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ with five subsets $S_1 = \{2, 3, 5, 7\}$, $S_2 = \{1, 2, 3, 5, 8\}$, $S_3 = \{1, 2, 4, 8\}$, $S_4 = \{3, 6, 9\}$, and $S_5 = \{4, 6, 8, 9\}$. What is an output of $\mathrm{MIN\text{-}SET\text{-}COVER}$ on this input? **(2)**

Let $S = \{x_1, x_2, \ldots, x_n\}$ be of size n, and let f_i be the count of the subsets S_j containing the element x_i. Finally, let $f = \max(f_1, f_2, \ldots, f_n)$. (The counts f_i are the frequencies of the elements, and f is the maximum frequency.)

(b) Design a polynomial-time f-approximation algorithm for $\mathrm{MIN\text{-}SET\text{-}COVER}$. **(6)**

(c) Prove that your algorithm achieves an approximation ratio of f. **(6)**

(d) Prove or disprove: The approximation ratio f achieved by your algorithm is tight. **(6)**

Clue: $f = 2$ for the $\mathrm{MIN\text{-}VERTEX\text{-}COVER}$ problem.

ROUGH WORK