CSCI-GA 3520: Honors Analysis of Algorithms

Final Exam: Thu, Dec 22 2016, Room WWH-101, 10:00-2:00pm.

- This is a four hour exam. There are six questions, worth 10 points each. Answer all questions and all their subparts.
- This is a closed book exam. No books, notes, reference material, either hard-copy, soft-copy or online, is allowed.
- In the past years, to pass this exam, one needed to answer 3 or 4 problems well, instead of answering all the problems poorly.
- 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.
- **You must prove correctness of your algorithm and prove its time bound unless stated otherwise. The algorithm can be written in plain English (preferred) or as a pseudo-code.**

Best of luck!

Problem 1

Let $A[0], A[1], \ldots, A[n-1]$ be an array of nonnegative integers and assume that $0 \leqslant A[i] < n$ for all $0 \leqslant i < n$. Give a $O(n)$ time sorting algorithm with input array A that returns (a possibly different) array with values in non-decreasing order.

Now assume that $0 \leqslant A[i] < n^3$ for all $0 \leqslant i < n$. Again give a $O(n)$ time sorting algorithm with input array A that returns (a possibly different) array with values in non-decreasing order.

In both cases, you must describe the algorithm and argue its correctness and running time.

Problem 2

A 3-*coloring* of an undirected graph $G(V, E)$ is a map $\psi : V \to \{\mathsf{Red}, \mathsf{Blue}, \mathsf{Green}\}$, i.e. each vertex is assigned one of the three colors. Given a 3-coloring ψ, an edge $\{u, v\} \in E$ is said to be *properly colored* if $\psi(u) \neq \psi(v)$, i.e. if the endpoints of the edge receive different colors.

Let n be the number of vertices and m be the number of edges in the graph. Show that there exists a 3-coloring $\psi : V \to \{\mathsf{Red}, \mathsf{Blue}, \mathsf{Green}\}$ such that at least $\frac{2m}{3}$ edges are properly colored. Give a polynomial time algorithm (deterministic or randomized) that finds a 3-coloring $\psi : V \to \{\mathsf{Red}, \mathsf{Blue}, \mathsf{Green}\}$ such that at least $\frac{2m}{3}$ edges are properly colored.

Problem 3

Let $G(V, E)$ be a directed graph with non-negative weights on the edges. Assume adjacency list representation of the graph. State (do not describe, prove, or analyze) what Dijkstra's shortest path algorithm achieves and what its running time is. Now solve the following problem.

Tucker, who lives at a node s of a directed graph $G(V, E)$ with non-negative weights on the edges, is invited to an exciting party located at a node t, where he will meet the girl of his dreams, Sharona. Naturally, Tucker wants to get from s to t as soon as possible, but he is told to buy some beer on the way over. He can get beer at any supermarket, and the supermarkets form a subset of the vertices $B \subseteq V$. Thus, starting at s, he must go to some node $b \in B$ of his choice, and then head from b to t, using the shortest overall route possible (assume he wastes no time at the supermarket).

Show how Tucker can find the shortest route using Dijkstra's algorithm. For full credit, the running time of his algorithm should be of the same order as that of Dijkstra's algorithm.

Problem 4

Consider the following algorithm (the print operation prints a single # sign; the operation $x \leftarrow 2x$ doubles the value of the variable x).

For $k = 1$ **to** n {
    $x = k$
    While $x < n$ {
        Print #
        $x \leftarrow 2x$
    }
}

How many (exact value) # signs are printed during the execution of the inner **While** loop as a function of k and n?

Write the total number of # signs printed during the execution of the entire algorithm as a sum. Show that the sum, asymptotically as a function of n, is $\Theta(n)$. Note that you have to show that the sum is *at least* $\Omega(n)$ as well as *at most* $O(n)$.

Problem 5

We have a list of n jobs $(J_1, J_2, \ldots, J_n)$ such that executing the i^{th} job on a single machine takes t_i units of time. We have k machines that can work in parallel and our goal is to schedule (assign) the jobs onto machines so as to complete all jobs as soon as possible. However we do have one constraint: *jobs assigned to any machine must be a consecutive subsequence of the list* $(J_1, J_2, \ldots, J_n)$.

Here is an example. Suppose $n = 10$ and $k = 4$. A feasible schedule could be

$$(J_1, J_2, J_3), \quad (J_4), \quad (J_5, J_6, J_7, J_8), \quad (J_9, J_{10}).$$

The four groups are assigned to the four machines respectively, which take time

$$t_1 + t_2 + t_3, \quad t_4, \quad t_5 + t_6 + t_7 + t_8, \quad t_9 + t_{10},$$

respectively. Thus all jobs are completed in time

$$\max\{t_1 + t_2 + t_3, \quad t_4, \quad t_5 + t_6 + t_7 + t_8, \quad t_9 + t_{10}\}.$$

Design a dynamic programming based algorithm that runs in time polynomial in n and k and finds a schedule with minimum completion time.

Problem 6

Recall that a 3CNF formula ϕ consists of n boolean variables $x_1, \ldots, x_n$ and m clauses where each clause is a logical OR of three literals. A literal is a variable x_i or its negation $\overline{x}_i$. Recall that

$$\mathsf{3SAT} = \{\phi \mid \phi \text{ is a 3CNF formula that has an assignment satisfying all clauses}\}.$$

In a *Non-mixed-3CNF* formula, each clause may have one, two, or three literals, and in each clause, either all variables appear in un-negated form or all variables appear in negated form. That is, each clause is of one of the following types:

$$x_i, \quad \overline{x}_i, \quad x_i \vee x_j, \quad \overline{x}_i \vee \overline{x}_j, \quad x_i \vee x_j \vee x_k, \quad \overline{x}_i \vee \overline{x}_j \vee \overline{x}_k.$$

Let

$$\mathsf{Non\ mixed\ 3SAT} = \{\phi \mid \phi \text{ is a Non-mixed-3CNF formula that has an assignment satisfying all clauses}\}.$$

Show that $\mathsf{Non\ mixed\ 3SAT}$ is NP-complete. You can give a reduction from $\mathsf{3SAT}$.