NAME: _____

CSE 331
Introduction to Algorithm Analysis and Design
Sample Mid-term Exam

Jesse Hartloff

DIRECTIONS:

- Open Notes. Closed book. No electronics.
- Time Limit: 120 minutes.
- Make sure you write your NAME on the paper.
- If you need extra space use the back of a page.

1	/40
2	/45
3	/15
Total	/100

FEW GENTLE REMINDERS:

- You can quote any result that we covered in class or any problem that was in a homework (but remember to explicitly state where you are quoting a result from).
- The ordering of the problems is somewhat related to their relative difficulty. However, the order might be different for you!
- You might be better off by first reading all questions and answering them in the order of what you think is the easiest to the hardest problem. Keep the points distribution in mind when deciding how much time to spend on each problem.

1. ($4 \times 10 = 40$ points) Answer True or False to the following questions and briefly JUSTIFY each answer. A correct answer with no or totally incorrect justification will get you 4 out of the total 10 points. (Recall that a statement is true only if it is logically true in all cases while it is is false if it is not true in some case).

   (a) For any instance of the stable marriage problem with n men and n women, there are $n! = n \times (n-1) \times \cdots \times 1$ many possible perfect matchings.

   (b) Consider $f(n) = \log \log n$ and $g(n) = 10^{10^{10^{10^{10}}}}$. Then $f(n)$ is $O(g(n))$.

   (c) For any graph, there is a unique BFS tree for it.

   (d) Given a graph on n vertices in its adjacency matrix, there is an $O(n^2)$ time algorithm to convert it into its adjacency list representations.

2. (20 + 25 = 45 points) Given an array A of n integers, consider the the following algorithm that computes a related value (and an intermediate matrix B):

    For every $i = 1, \ldots, n$

        For every $j = i, \ldots, n$

            Assign $B[i, j]$ to be the maximum value among $A[i], A[i + 1], \ldots, A[j]$.

    Output the minimum value among all values in $B[i, j]$ (over all $i = 1, \ldots, n$ and $j = i, \ldots, n$).

   (a) Prove that the algorithm runs in $O(n^3)$ time.

(b) Present another algorithm that solves the same problem but runs in $O(n^2)$ time. (Briefly justify the running time and correctness of your algorithm.)

3. (15 points)

Ms. Greedy Chef has to schedule n clients for lunch for the next day. These n clients are regular customers, so for each customer i, Ms. Chef knows the exact time b_i it would take her to cook the burger exactly the way i likes it. Unfortunately, Ms. Chef's restaurant is new and the kitchen is small, so she can only cook one burger at a time. Ms. Chef is also gunning for a record, so she has decided not to waste any time moving from one burger to the next. However, given that her customers are very picky, Ms Chef has to spend one time unit between making two successive burgers to clean up her utensils. Help Ms. Chef design an $O(n \log n)$ time algorithm that schedules all the n clients, which provably minimizes the *sum* of the *service time* for every client, where the service time for client i is b_i plus the time i had to wait before Ms. Chef started i's burger. You should assume that all the clients came into the restaurant at the same time. As usual, you must prove the optimality of your algorithm.

Here is a simple example: say $n = 3$ and

$$b_1 = 5, \quad b_2 = 10, \quad b_3 = 4.$$

Now consider the schedule

$$1, \text{ break }, 2, \text{ break }, 3$$

i.e. 1 gets her burger first and then leaves, 2 gets his burger after 1 and 3 gets his burger after 2. Note that the service time for 1 is 5. The service time for client 2 is his wait time (which is $5 + 1 = 6$) plus b_2, which is 16. The service time for client 3 is his wait time (which is $5 + 1 + 10 + 1 = 17$) plus b_3, which is 21. Thus, the sum of the service times for this schedule is $5 + 16 + 21 = 42$. This schedule, however, is *not* optimal.