CSE 101 Final Exam

Fall 2019

Instructions: Do not open until the exam starts. The exam will run for 180 minutes. The problems are roughly sorted in increasing order of difficulty. Answer all questions completely. You are free to make use of any result in the textbook or proved in class. You may use up to 12 1-sided pages of notes, and may not use the textbook nor any electronic aids. Write your solutions in the space provided, the pages at the end of this handout, or on the scratch paper provided (be sure to label it with your name). If you have solutions written anywhere other than the provided space be sure to indicate where they are to be found. Please sit in the seat indicated below.

If the problem asks for an algorithm, giving a correct algorithm with worse runtime efficiency than what is asked for will be awarded partial credit.

Name:

ID Number:

Seat:

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 1 (Karatsuba Multiplication, 30 points). What three recursive top-level multiplies are called for when one uses Karatsuba multiplication to multiply the binary numbers 10110101 and 11101001?

Please write out the three new multiplication calls (you don’t need to evaluate the answers). Your answers should also be in binary.
Question 2 (Independent Set Computation, 30 points). Compute the size of the largest independent set of the graph given below.

[It is recommended that you show your work so that you can be given partial credit if your answer is not correct.]
Question 3 (Vacation Days Planning, 35 points). Over the course of the next $n$ workdays Abby accrues one vacation day every 30 (even if she is on vacation at the time). She can save up vacation days as much as she likes, but can never spend days that she hasn't earned yet. During this period, there are a number of possible vacations that she is considering going on. Each vacation has a range of consecutive days that it would take. Abby cannot go for only part of the vacation, nor can she go on more than one vacation at once. Each vacation also has a fun rating.

Abby wants to figure out which collection of vacations she should go on in order to maximize her total fun without exceeding her allotted vacation days. Give a polynomial time algorithm to determine the most possible fun she can achieve and analyze its runtime.

[Hint: for each day and each $k$ find the most amount of fun that Abby can have by that day with at least $k$ vacation days remaining.]
Question 4 (Travel Planning, 35 points). Wendy is trying to travel through Neverland. She has a map of locations and routes including how long each route takes to traverse. Unfortunately, the landscape is constantly changing, and each route will only be possible for a known interval of time (though she can wait as long as she wants at any of the locations).

Wendy wants an algorithm that given a time for the start of her journey, n locations (two designated as start and end), and m routes where for each route she is given the start location of the route, the end location of the route, the time it takes to traverse, and the start and end times of its window of availability (she can only begin her journey along the route at times during the window).

Give an algorithm that allows her to compute the earlier possible time that she could arrive at her destination. For full credit, your algorithm should run in time $O(n \log(n) + m)$ or better.
Question 5 (Bad Chain Avoidance, 35 points). Let $G$ be a DAG whose vertices are labelled black and white. Give an algorithm that given $G$, an integer $k$ and vertices $s$ and $t$ determines whether or not there is a path from $s$ to $t$ that does not pass through $k$ black vertices in a row at any point.

For full credit your algorithm should have runtime $O(|V| + |E|)$ or better.
Question 6 (Greedy Candyland, 35 points). In the game of Candyland, the board consists of a path of colored squares. You begin at the first square and try to reach the end of the path. On each turn you draw a card with a color on it and advance your piece to the next square with that color (or stay where you are if there are no further squares of that color).

Adam is able to cheat by stacking the deck in order to get whatever colors of cards he wants. In order to get to the end in as few moves as possible, he arranges the deck so that on each turn he gets the color of card that will allow him to advance his piece as far as possible.

Does this necessarily allow Adam to finish the game in the minimal number of turns? Prove this or provide a counterexample.