CS 141 - Intermediate Data Structures and Algorithms

Quiz #2

30 points Monday 2/27/2017
45 minutes ** Closed book and notes **

NAME: ___________________________ ID #: ____________________

<table>
<thead>
<tr>
<th>Q1 (15)</th>
<th>Q2 (15)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Q1. (15 points) [Coin changing problem] Suppose that you want to carry a money amount of \( n \) cents and you want to minimize the number of coins to carry. We only consider pennies, nickels, dimes, and quarters with denominations 1, 5, 10, and 25 cents, respectively. For example, an amount of \( n = 71 \) cents becomes two quarters, two dimes and a penny. An amount of \( n = 81 \) becomes three quarters, a nickel, and a penny.

(a) (5 points) Describe a greedy algorithm to solve the above problem and write a pseudo code for it.

(b) (5 points) Prove the optimality of your algorithm by establishing the greedy choice and optimal substructure properties. Hint: To prove the greedy choice property, you can consider each of the four possible decisions separately, i.e., picking a penny, a nickel, a dime, and a quarter, preferably in this order.

(c) (3 points) Establish the running time of your algorithm.

(d) (2 points) Apply your algorithm to the problem when \( n = 57 \).
Extra page for Q1
Q2. (15 points) [Largest square block of ones]: Given a square matrix $M$ of dimensions $n \times n$ where each entry $m_{ij}$ is either 0 or 1, we would like to find the largest square block of ones. For example, in the matrix shown below, the largest square block of ones is of size 4 and is located at position 1, 4 of the matrix, i.e., its top left is at row $i = 1$ and column $j = 4$. Develop a dynamic programming algorithm that finds both the size and location of the maximum square block of ones in an input matrix.

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0
\end{bmatrix}
\]

(a) (6 points) Develop a recursive formula for the function $LSBO(i, j)$ which defines the size of the largest square block of ones for any position $i, j$ in the matrix. Prove the correctness of the developed formula.

(b) (6 points) Develop a bottom-up dynamic programming algorithm based on the recursive formula. You need only to write the pseudo code of the algorithm.

(c) (3 points) Establish the running time of your bottom-up algorithm.
Extra page for Q2