Canvas Quiz: Greedy Strategies

1. All possible greedy algorithms, at each step, choose what they know is going to lead to an optimal solution for the general problem.
   a. True
   b. False

2. Activities $a_1,a_2,...,a_{12}$ need to use the same resource. In the table below you can see when they start and when they finish. What is the maximum number of activities that can be completed without having conflicts?

<table>
<thead>
<tr>
<th>Activity</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>A9</th>
<th>A10</th>
<th>A11</th>
<th>A12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Finish</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>7</td>
<td>12</td>
<td>14</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

   a. 3
   b. 4
   c. 5

3. What is the tightest worst case complexity of the iterative greedy activity selector algorithm? First assuming the input activities are ordered by monotonically increasing finish time, and second not assuming the inputs are ordered in any particular way.
   a. $\log(n)$ and $n \log(n)$
   b. $n \log(n)$ and $n \log(n)$
   c. $n$ and $n \log(n)$
   d. $n^2$ and $n^2 \log n$

4. Which of the following are not a step in designing a greedy algorithm?
   1. Cast the problem into 2 or more subproblems for which we make the best greedy choices at subsequent steps.
   2. Prove that the greedy choice in each step is the only choice that leads to the optimal solution for the general problem.
   3. Neither are steps in designing a greedy algorithm
   4. They are both steps in designing a greedy algorithm

5. The similarity of greedy algorithms and dynamic programming is that they both pre-calculate optimal solutions to sub-problems.
   a. True
   b. False

6. Consider a text containing the alphabets in the chart. The number below each character shows the frequency. If we make Huffman tree for these characters, what would be the height of the tree. (Suppose the root of the tree has height 0)

<table>
<thead>
<tr>
<th>a</th>
<th>d</th>
<th>e</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>25</td>
<td>17</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

   a. 4
   b. 3
7. In the previous question what would be the length of the shortest and longest code generated by Huffman algorithm?
   a. 1, 4
   b. 1, 5
   c. 2, 4
   d. 2, 5

8. What is the running time of Huffman algorithm if we use a simple array (with no sorting) for storing the frequencies?
   a. O(n²)
   b. O(n² log n)
   c. O(n log n)
   d. O(n)

9. What would be the running time of the Huffman algorithm if we use a min-heap for storing the frequencies?
   O(n)
   O(n log n)
   O(n²)
   O(n² log n)

10. Which one of the choices below could be the main purpose of the Huffman algorithm? (Choose the best answer)
    a. Encrypting the data, so no one can understand the real text.
    b. Storing the data in a way that reduces the size of the text.
    c. Changing the text data to binary numbers.
    d. Making a tree representation of a text file.
Answers:
1. F
2. 4
3. n and n log(n)
4. Neither are steps in designing a greedy algorithm
5. F
6. 4
7. 1, 4
8. O(n^2)
9. O(n log n)
10. Storing the data in a way that reduces the size of the text.