Exam 2 (100 minutes)
CS 330 Discrete Structures
Summer Semester, 2016

Name:
A-ID:

- Only a pen and whatever provided by the instructor are permitted throughout the exam.
- You have to show your work. You will not get partial credits if the grader cannot figure out how you arrived at the answer.
- You do NOT need to calculate and simplify anything. Leave the notations as they are.

<table>
<thead>
<tr>
<th>Question</th>
<th>Maximum Points</th>
<th>Earned Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. Describe the simplest format of Big-Theta notation for the following functions (Do NOT need to explain the reason).

- $3n^2 + 8n + 2$
  \[ \Theta(n^2) \]

- $\left( \begin{array}{c} n \\ 3 \end{array} \right)$
  \[ \Theta(n^3) \text{ since } \left( \begin{array}{c} n \\ 3 \end{array} \right) = \frac{n!}{(n-3)!3!} = \frac{n(n-1)(n-2)}{3 \cdot 2 \cdot 1} \]

- $(\log \log n)^3 + 3(2 + \log n)$
  \[ \Theta(\log n) \]

- $(n + 2) \log(n^2)$
  \[ \Theta(n \log n) \text{ since } \log(n^2) = 2 \log n \]

- $2^{2n} + 3^n$
  \[ \Theta(4^n) \text{ since } 2^{2n} = 4^n \]
2. \( A[1], A[2], \ldots, A[n] \) is an array of \( n \) integers.

```plaintext
1: for i ← 1 to n do
2:   if A[i] > 100 then
3:     Break
4:   end if
5: end for
6: return i
```

- Analyze the time complexity of the algorithm in terms of the Big-Oh or Big-Theta notation.
  There is a for loop which will be iterated for \( n \) times in the worst case, but it depends on the input sequence \( A \). In the best case, the for loop will be iterated only once. Therefore, the time complexity is \( O(n) \) instead of \( \Theta(n) \).
- What does this algorithm do?
  It returns the index of the first element that is larger than 100.
3. Fill in the blanks in the following table. Growth rates should be given in the simplest Big-Theta format. For the recurrence and closed formula, any one example is sufficient.

<table>
<thead>
<tr>
<th>Annihilator</th>
<th>Growth Rate</th>
<th>Recurrence</th>
<th>Closed formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(E - 4)^2$</td>
<td>$\Theta(n4^n)$</td>
<td>$a_n = 8a_{n-1} - 16a_{n-2}$</td>
<td>$a_n = (\alpha n + \beta)4^n$</td>
</tr>
<tr>
<td>$(E - 4)$</td>
<td>$\Theta(4^n)$</td>
<td>$a_n = 4a_{n-1}$</td>
<td>$a_n = \alpha 4^n$</td>
</tr>
<tr>
<td>$(E - 1)^n$</td>
<td>$\Theta(n^n)$</td>
<td>$S_n = S_{n-2} + n^3$</td>
<td>$\alpha n^3 + \beta n^3 + \gamma n^2 + \delta n + \eta$</td>
</tr>
<tr>
<td>N/A</td>
<td>$\Theta(n \log n)$</td>
<td>$T(n) = 2T\left(\frac{n}{2}\right) + n$</td>
<td>$(\alpha \log n + \beta)n$</td>
</tr>
</tbody>
</table>

The annihilator of the last row is beyond this level’s course. Full credit is given to everyone for that cell.
4. The instructor presented an algorithm whose time complexity is defined by

\[ T(n) = 4T\left(\frac{n}{2}\right) + n \]

Later, a student said his algorithm is better than that, and the time complexity of that algorithm is defined by

\[ T(n) = 10T\left(\frac{n}{4}\right) + n \]

Is this student’s algorithm better than the instructor’s one? Explain your answer with Master Theorem.

- Instructor’s algorithm: \( a = 4, b = 2, f(n) = n \Rightarrow \frac{af(n/b)}{f(n)} = 2 > 1 \), therefore, \( T(n) = \Theta(n^{\log_b a}) = \Theta(n^2) \)

- Student’s algorithm: \( a = 10, b = 4, f(n) = n \Rightarrow \frac{af(n/b)}{f(n)} = 2.5 > 1 \), therefore, \( T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_4 10}) \)

Since \( \log_4 10 < \log_4 16 = 2 \), the student’s algorithm has a better asymptotic performance.
5. Sort the following characters by using the ‘merge sort’, and show its procedure step by step.

\[ H, B, G, D, A, E, F, C \]

\begin{align*}
& H, B, G, D \quad A, E, F, C \\
& H, B \quad G, D \quad A, E \quad F, C \\
& H \quad B \quad G \quad D \quad A \quad E \quad F \quad C \\
& B, H \quad D, G \quad A, E \quad C, F \\
& B, D, G, H \quad A, C, E, F \\
& A, B, C, D, E, F, G, H
\end{align*}
6. • What is the adjacency matrix of the following graph?

![Graph Diagram]

\[
\begin{bmatrix}
0 & 8 & 1 & 0 & 7 & 2 & 0 \\
8 & 0 & 4 & 2 & 4 & 0 & 3 \\
1 & 4 & 0 & 6 & 0 & 0 & 2 \\
0 & 2 & 6 & 0 & 0 & 5 & 4 \\
7 & 4 & 0 & 0 & 0 & 5 & 1 \\
2 & 0 & 0 & 5 & 5 & 0 & 3 \\
0 & 3 & 2 & 4 & 1 & 3 & 0
\end{bmatrix}
\]

• What is the length of the path \((a, b, c, d, f, e)\)? (Note that the graph is weighted). What would be the length of the same path if the graph were unweighted?

• **Weighted:** The length is 28.

• **Unweighted:** The length is 5.
7. • Is there an Euler circuit in the following graph? Why?

Every vertex’s degree is even, therefore there is an Euler circuit.

• Is the following graph planar? If so, draw a planar version. Otherwise, explain your answer.

Yes, the graph is planar because it is isomorphic to the following one.
8. Show how to color the following graph with 4 colors in the graph coloring problem.
9. Given the following graph:

![Graph Image]

- Show a result of breadth-first search performed on the graph when it started at vertex $a$. Mark the distances for all vertices.

![Search Result Diagram]
• Show a result of depth-first search performed on the graph when it started at vertex $a$. Mark the start/finish time for all vertices.