7200. Analysis of Algorithms

Midterm Exam

March 22, 2016

Id: .................................................................

- Answer all questions.
- Answer a question only within the given space by using a readable normal size font text.
- You get 20% of the credit if you leave the answer blank. You get no credit for a wrong answer.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Maximum Points</th>
<th>Your Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Good Luck!
1. For the following 10 multi-choice problems you will earn 2 points for a correct answer and 0 point for a wrong answer or no answer.

(a) Which of the following describes most accurately the relationship between $\log_2(n^2)$ and $\log_{10}(\sqrt{n})$?
   i. $\log_2(n^2) = o(\log_{10}(\sqrt{n})$
   ii. $\log_2(n^2) = O(\log_{10}(\sqrt{n})$
   iii. $\log_2(n^2) = \Theta(\log_{10}(\sqrt{n})$
   iv. $\log_2(n^2) = \Omega(\log_{10}(\sqrt{n})$
   v. $\log_2(n^2) = \omega(\log_{10}(\sqrt{n})$

(b) Which of the following describes most accurately the relationship between $n \log_2(n)$ and $n$?
   i. $n \log_2(n) = o(n)$
   ii. $n \log_2(n) = O(n)$
   iii. $n \log_2(n) = \Theta(n)$
   iv. $n \log_2(n) = \Omega(n)$
   v. $n \log_2(n) = \omega(n)$

(c) What is the solution to the recursive formula $T(1) = 1$ and $T(n) = T(\lceil n/2 \rceil) + 1$?
   i. $T(n) = \Theta(\log(n))$
   ii. $T(n) = \Theta(\log^2(n))$
   iii. $T(n) = \Theta(n)$
   iv. $T(n) = \Theta(n \log(n))$
   v. $T(n) = \Theta(n^2)$

(d) What is the solution to the recursive formula $T(1) = 1$ and $T(n) = T(\lceil n/2 \rceil) + n$?
   i. $T(n) = \Theta(\log(n))$
   ii. $T(n) = \Theta(\log^2(n))$
   iii. $T(n) = \Theta(n)$
   iv. $T(n) = \Theta(n \log(n))$
   v. $T(n) = \Theta(n^2)$

(e) What is the solution to the recursive formula $T(1) = 1$ and $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil)$?
   i. $T(n) = \Theta(\log(n))$
   ii. $T(n) = \Theta(\log^2(n))$
   iii. $T(n) = \Theta(n)$
   iv. $T(n) = \Theta(n \log(n))$
   v. $T(n) = \Theta(n^2)$
(f) What is the solution to the recursive formula
\[ T(1) = 1 \text{ and } T(n) = T(n-1) + n? \]

i. \( T(n) = \Theta(\log(n)) \).
ii. \( T(n) = \Theta(\log^2(n)) \).
iii. \( T(n) = \Theta(n) \).
iv. \( T(n) = \Theta(n \log(n)) \).
v. \( T(n) = \Theta(n^2) \).

(g) Algorithm A solves Problem P with a worst-case tight bound of \( \Theta(n) \) on its running time. Which of the following statements may be true?

i. For infinitely many values of \( n \), there are instances of P of length \( n \) for which the running time of A is \( n^2 \).
ii. For all \( n \), and for all instances of P of length \( n \), the running time of A is \( \Omega(n \log n) \).
iii. For infinitely many values of \( n \), there are instances of P of length \( n \) for which the running time of A is \( \log(n) \).
iv. For all \( n \), and for all instances of P of length \( n \), the running time of A is \( O(\sqrt{n}) \).
v. None of the above.

(h) For Problem P with parameters \( k \) and \( n \), Algorithm A has a worst-case tight bound of \( \Theta(kn) \) on its running time and Algorithm B has a worst-case tight bound of \( \Theta(k^2 \sqrt{n}) \) on its running time. For which value of \( k \) both algorithms have the same asymptotic complexity?

i. \( k = \Theta(1) \)
ii. \( k = \log_2(n) \)
iii. \( k = \sqrt{n} \)
iv. \( k = n \)
v. \( k = n \sqrt{n} \)

(i) Which of the following statements is always correct?

i. If \( f(n) = O(g(n)) \) then \( f(n) \leq g(n) \) for all \( n \geq 1 \).
ii. If \( g(n) = \Omega(f(n)) \) then \( g(n) \geq f(n) \) for all \( n \geq 2^{100} \).
iii. If \( f(n) = O(g(n)) \) and \( g(n) = O(h(n)) \) then \( f(n) = \Theta(h(n)) \).
iv. If \( f(n) = O(g(n)) \) and \( g(n) = O(h(n)) \) then \( f(n) = O(h(n)) \).
v. None of the above.

(j) What is the time complexity of the following segment of code?

\[
\begin{align*}
x &= 1 \\
\text{while } x \leq n^{100} \text{ do } \\
\quad x &= 2x
\end{align*}
\]

i. \( \Theta(1) \)
ii. \( \Theta(\log(n)) \)
iii. \( \Theta(n) \)
iv. \( \Theta(n \log(n)) \)
v. \( \Theta(n^2) \).


Describe an efficient algorithm to determine whether there exists an index $i$ such that $A[i] = B[i]$.

What is the worst-case number of comparisons made by your algorithm? Try to be exact. Otherwise, you may use the $O$-notation.

Justify the correctness and complexity of your algorithm.
If you are not sure about your previous algorithm, describe any (possibly not efficient) correct algorithm that solves the problem.

What is the worst-case number of comparisons made by your algorithm?

Try to be exact. Otherwise, you may use the $O$-notation.

Justify the correctness and complexity of your algorithm.


Let \( k \) be a positive integer.

Describe an efficient algorithm to determine whether there exist two indices \( 1 \leq i, j \leq n \) such that \( A[i] + B[j] = k \).

What is the worst-case number of additions made by your algorithm?

Try to be exact. Otherwise, you may use the \( O \)-notation.

Justify the correctness and complexity of your algorithm.
If you are not sure about your previous algorithm, describe any (possibly not efficient) correct algorithm that solves the problem.
What is the worst-case number of comparisons made by your algorithm?
Try to be exact. Otherwise, you may use the $O$-notation.
Justify the correctness and complexity of your algorithm.