7200. Analysis of Algorithms

Midterm Exam

October 17, 2017

• Answer Problems 1, 2, and 3a. The bonus problem, 3b, is extra.
• You have exactly two hours.
• Answer a question only within the given space by using a readable normal size text.
• Except for the bonus problem, 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>Part</th>
<th>Maximum Points</th>
<th>Your Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>b</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>c</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>d</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Final Grade

<table>
<thead>
<tr>
<th>Problem</th>
<th>Part</th>
<th>Maximum Points</th>
<th>Your Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>b</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Good Luck!
1. For the following four procedures find the exact value of $c$ when the procedure terminates. If you cannot determine the exact value use the $\Theta$-notation.

(a) $f(n)$ (* $n \geq 0$ is an integer *)
$$c = 0$$
for $i = 1$ to $n$
  for $j = i$ to $n$
    $c := c + 1$

(b) $f(n)$ (* $n \geq 0$ is an integer *)
$$c = 0$$
for $i = 1$ to $n$
  for $j = 1$ to $2n$
    for $k = 1$ to $3n$
      $c := c + 1$

(c) $f(n)$ (* $n = k^2$ is a positive square integer *)
$$c = 0$$
for $i = 1$ to $n$
  if $i$ is a square number
    then $c := c + 1$

(d) $f(n)$ (* $n = 2^k$ is a power of 2 integer *)
$$c = 0$$
while $n > 1$
  $n := n/2$
  $c := c + 1$
2. The following *Trinary-Search Procedure* looks for $x$ in the range $\{f, \ldots, \ell\}$ for some positive integers $1 \leq f < \ell \leq n$. Assume that always the size of the range, $\ell - f + 1$, is a power of 3.

- The procedure first checks if $f = \ell$ (not a comparison). If the answer is “YES” then the search terminates.
- Otherwise, let $m = \frac{\ell - f + 1}{3}$ be one third of the size of the range. The procedure partitions the input range into the following 3 equal size ranges: $\{f, \ldots, f + m - 1\}$, $\{f + m, \ldots, \ell - m\}$, and $\{\ell - m + 1, \ldots, \ell\}$. For example the range $\{10, \ldots, 18\}$ is partitioned into the ranges: $\{10, 11, 12\}$, $\{13, 14, 15\}$, and $\{16, 17, 18\}$.
- The first comparison is: $x \leq f + m - 1$ (does $x$ belong to the first range?).
- If the answer is “YES” then the search continues recursively in the first range: $\{f, \ldots, f + m - 1\}$.
- If the answer is “NO” a second comparison is performed: $x \leq \ell - m$ (does $x$ belong to the second range?).
- If the answer is “YES” then the search continues recursively in the second range: $\{f + m, \ldots, \ell - m\}$.
- If the answer is “NO” then the search continues recursively in the third range: $\{\ell - m + 1, \ldots, \ell\}$.

(a) For the range $[1..n]$, what is the exact number of comparisons between $x$ and a number in the range that are performed by Trinary-Search?

- $n = 3$ and $x = 1, 2, 3$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>#comparisons</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $n = 9$ and $x = 1, \ldots, 9$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>#comparisons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $x = n$ for any $n$. Justify your answer.

- $x = 1$ for any $n$. Justify your answer.
(b) Let $T(n)$ be the worst case complexity (number of comparisons) of Trinary-Search. Write the initial value and the recursive formula for $T(n)$.

What is the solution of this recursion using the $\Theta$-notation. Justify your answer.

What is the exact solution of this recursion? Prove your answer.
3. Let $M$ be an $n \times n$ matrix containing the numbers $1, 2, \ldots, n^2$. The $n$ integers of each row of $M$ are sorted in an increasing order (from left to right) and the $n$ integers of each column of $M$ are sorted in an increasing order (from top to bottom).

Let $1 \leq x \leq n^2$ be an unknown numbers. The goal is to find the position $M[i,j]$ of $x$ in $M$ ($x$ is in row $i$ and column $j$) using only comparisons between $x$ and integers in the matrix.

(a) Describe an algorithm that always finds the position of $x$ with $O(n)$ comparisons. Explain why your algorithm is correct and justify why it is a linear time algorithm.
If you are not sure about the correctness of your previous algorithm, describe a (possibly not efficient) **correct** algorithm that finds the position of \( x \) in \( M \). Explain why your algorithm is correct.

What is the worst case number of comparisons between \( x \) and integers in \( M \) made by your algorithm as a function of \( n \)? Justify your answer.
(b) **Bonus question:** Prove that any algorithm that solves this problem must use $\Omega(n)$ comparisons.