Answers to Quiz 2

CIS 675 Algorithms

Distribution of scores

Average: 14.85  Median: 15.5

0- 1:  1
2- 3:  4
4- 5:  4
6- 7:  9
8- 9:  9
10-11: 11
12-13: 11
14-15: 11
16-17: 11
18-19: 11
20:  20

Average ≈ 14.85  Median = 15.5

Note: For the histogram, fractional scores were rounded down.

Problem 1 (8 points) For each of the following give a tight $\Theta(\cdot)$ bound on the number of times the $z=z+1$ statement is executed (2 points) and justify your answer (2 points).

e. (4 points)

```
  j = 1;
  while (j<=n) {
    j = j*2;
    z = z+1;
  }
```

An answer: Since $j$ doubles every time through the loop, at the beginning of the $i$-th iteration, the value of $j$ is $2^i$. The loop exits with the first $i$ with $2^i > n$, i.e., $i > \log_2 n$. Therefore, $z$ increases by $\Theta(\log n)$.

f. (4 points)

```
  j = 1;
  while (j<=n) {
    j = j+j+j;
    z = z+1;
  }
```

An answer: Since $j$ triples every time through the loop, at the beginning of the $i$-th iteration, the value of $j$ is $3^i$. The loop exits with the first $i$ with $3^i > n$, i.e., $i > \log_3 n$. Therefore, $z$ increases by $\Theta(\log n)$.

g. (4 points)

```
  for(k=0; k<=n; k++)
    for(j=n; j>=k;j--)
      z = z+1
```

An answer: The inner loop goes through $n-k+1$ many iterations. So the inner loop increases $z$ by $n-k+1$. In the outer loop, $k$ goes from 0 to $n$ in steps of 1. Therefore, $z$ is increased by $\sum_{k=0}^{n} n-k+1 = (n+1) + n + \cdots + 2 + 1 = \frac{1}{2}(n+1)(n+2)$ by Math Fact j. So $z \in \Theta(n^2)$.

h. (4 points)

```
  for(k=n; k>=0; k--)
    for(j=0; j<k;j++)
      z = z+1
```

An answer: The inner loop goes through $k+1$ many iterations. So the inner loop increases $z$ by $k+1$. In the outer loop, $k$ goes from $n$ to 0 in steps of $-1$. Therefore, $z$ is increased by $\sum_{k=0}^{n} n-k+1 = (n+1) + n + \cdots + 2 + 1 = \frac{1}{2}(n+1)(n+2)$ by Math Fact j. So $z \in \Theta(n^2)$.

Problem 2 (4 points) [This problem is based on DPV Exercise 2.4.] For each of the following state the recurrence for the algorithm (1 point) and solve it via the Master Recurrence Theorem (1 point). (See the Math Facts page for a statement of the theorem.)

a. Algorithm W solves problems of size $n$ by dividing them into nine subproblems of size $n/2$, recursively solving them, and then combining the solutions in $\Theta(n^2)$ time.

An answer: For this case $a = 9$, $b = 2$, and $d = 2$. So $\log_a b = \log_2 9 \approx 3.17 > 2 = d$. Therefore, by the Master Recurrence Theorem, the algorithm runs in $\Theta(n^{\log_2 9})$ time.

b. Algorithm X solves problems of size $n$ by dividing them into five subproblems of size $n/2$, recursively solving them, and then combining the solutions in $\Theta(n)$ time.
Problem 3 (8 points)  [This problem is based on DPV Exercise 2.23(a).] A value $v$ is said to be the majority value of a list $[x_1, \ldots, x_k]$ when more than half of the elements of the list equal $v$, i.e., $\#\{i : x_i = v\} > k/2$; a value that is not a majority value is called a minority value. Given a list, we want to decide if a list has a majority value, and if so, we want to find this value. We assume we can do equality tests in constant time.

a. (1 points) Given a value $y$ and a list $[x_1, \ldots, x_k]$, show how we can test whether $y$ is the majority value of $[x_1, \ldots, x_k]$ in $\Theta(k)$ time. (Hint: Yes, this is really, really easy.)

An answer: Scan the list and count how many times $y$ occurs. Then: $y$ is the majority value if and only if the count is $> k/2$. Clearly this test takes $\Theta(k)$ time.

b. (2 points) Suppose we split the list $[x_1, \ldots, x_k]$ into two parts $[x_1, \ldots, x_{\lceil k/2 \rceil}]$ and $[x_{1+\lceil k/2 \rceil}, \ldots, x_k]$. Suppose $v$ is a minority value in both sublists. Explain why $v$ must be a minority value in the full list. (If you want, you may assume $k$ is even.)

An answer: Suppose the number of $y$’s in $[x_1, \ldots, x_{\lceil k/2 \rceil}]$ is $\leq \frac{1}{2} \lceil k/2 \rceil$ and the number of $y$’s in $[x_{1+\lceil k/2 \rceil}, \ldots, x_k]$ is $\leq \frac{1}{2} (k - \lfloor k/2 \rfloor)$. Then the number of $y$’s in $[x_1, \ldots, x_k]$ must be

$$\leq \frac{1}{2} \left\lfloor \frac{k}{2} \right\rfloor + \frac{1}{2} (k - \left\lfloor \frac{k}{2} \right\rfloor) = \frac{k}{2}.$$

c. (1 point) Suppose $v$ is a majority value in $[x_1, \ldots, x_k]$. Use part b to explain why $v$ must be a majority value in at least one of $[x_1, \ldots, x_{\lceil k/2 \rceil}]$ and $[x_{1+\lceil k/2 \rceil}, \ldots, x_k]$.

An answer: By the previous part, if $y$ is the majority in $[x_1, \ldots, x_k]$, it must also be the majority in at least one of $[x_1, \ldots, x_{\lceil k/2 \rceil}]$ and $[x_{1+\lceil k/2 \rceil}, \ldots, x_k]$.

d. (2 points) Use parts a and b to give a divide-and-conquer algorithm for the majority value problem. In the case that the list fails to have a majority value, the algorithm should return the value of the first element of the list.
An answer: Here is the algorithm:

```
procedure majority([x₁, . . . , xₖ])
    if k = 1 then return x₁
    mid ← ⌈k/2⌉
    m_left ← majority([x₁, . . . , x.mid])
    m_right ← majority([x.mid+1, . . . , xₖ])
    if m_left occurs > k/2 many times in [x₁, . . . , xₖ] then return m_left
    else if m_right occurs > k/2 many times in [x₁, . . . , xₖ] then return m_right
    else return x₁
```

e. (2 points) Use the master method to give a tight $O(\cdot)$-time bound on the algorithm’s run time.

An answer: On a list of length $k > 1$ there is a recursion on a list of length $\lceil k/2 \rceil$ and another on a list of length $\lfloor k/2 \rfloor$. Besides the recursions, the work done in a call is $O(k)$-time. So the recurrence is: $T(k) = 2T(\lceil k/2 \rceil) + O(k)$. Hence, $a = b = 2$, and $d = 1 = \log_2 2 = \log_b a$. Therefore, $T(k) \in \Theta(k \log k)$.

Some Reference Math Facts

a. $a^n \cdot a^n = a^{m+n}$.

b. $a^{m-n} = (a^m)^n = (a^n)^m$.

c. $a^n \cdot b^n = (a \cdot b)^n$.

d. $\log_a a^n = n$.

e. $a^{\log_a n} = n$.

f. $c \cdot \log_2 a = \log_2 a^c$.

g. $\log_2 (a \cdot b) = (\log_2 a) + (\log_2 b)$.

h. $(\log_a x)/(\log_a b) = \log_b x$.

i. $a^{\log_b c} = c^{\log_b a}$.

j. $\sum_{k=1}^{n} k = \frac{1}{2} \cdot n \cdot (n + 1)$.

k. $\sum_{k=0}^{n} a^k = \frac{a^{n+1} - 1}{a - 1}$.

l. $\sum_{k=1}^{\infty} 2^{-k} = 1$.

m. $\sum_{k=1}^{n} k^2 = \frac{1}{6} \cdot n \cdot (n + 1) \cdot (2n + 1)$.

n. $\sum_{k=1}^{\infty} k \cdot 2^{-k} = 2$.

o. For $a > 1$ and $b, c > 0$:

$$\lim_{n \to \infty} \frac{\log_n n}{n^c} = \lim_{n \to \infty} \frac{n^b}{n^c} = 0.$$  

p. If $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$, then:

<table>
<thead>
<tr>
<th>$c = 0$</th>
<th>$0 &lt; c &lt; \infty$</th>
<th>$c = \infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(n) \in O(g(n))$</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>$f(n) \in \Theta(g(n))$</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>$f(n) \in \Omega(g(n))$</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

q. The Master Recurrence Theorem: Suppose $T(n) = a \cdot T(n/b) + O(n^d)$ where $a, b \geq 1$ and $d \geq 0$. Then:

```

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\log_2 n$</th>
<th>$\log_3 n$</th>
<th>$\log_5 n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>0.631</td>
<td>0.431</td>
</tr>
<tr>
<td>3</td>
<td>1.585</td>
<td>1.000</td>
<td>0.683</td>
</tr>
<tr>
<td>4</td>
<td>2.000</td>
<td>1.262</td>
<td>0.861</td>
</tr>
<tr>
<td>5</td>
<td>2.322</td>
<td>1.465</td>
<td>1.000</td>
</tr>
<tr>
<td>6</td>
<td>2.585</td>
<td>1.631</td>
<td>1.113</td>
</tr>
<tr>
<td>7</td>
<td>2.807</td>
<td>1.771</td>
<td>1.209</td>
</tr>
<tr>
<td>8</td>
<td>3.000</td>
<td>1.893</td>
<td>1.292</td>
</tr>
<tr>
<td>9</td>
<td>3.170</td>
<td>2.000</td>
<td>1.365</td>
</tr>
</tbody>
</table>
```