CSE 2331 HOMEWORK 5

BRENDAN WHITAKER

1. The inner while loop executes $log_2(n-3)-1$ times, and so takes $clog_2(n)$ time.
   (a) In the worst case, $k \geq n/5$. So we have:

$$T(n) = \sum_{i=1}^{n^4} clog_2(n) = cn^4 log_2(n) \in \Theta(n^4 log_2(n)). \tag{1}$$

   (b) $Prob(k < n/5) = \lfloor n/5 \rfloor / n \approx 1/5$. Let X be the number of times $k < n/5$.
   The running time is $clog_2(n)(X+1)+c_1$, since lines 4-7 happen before our if statement.
   The expected running time is $E(clog_2(n)(X+1)+c_1) = clog_2(n)E(X) + clog_2(n) + c_1$, by linearity of expectation.
   We use the formula $E(X) = \sum_{l=1}^{\infty} Prob(X \geq l)$.
   $Prob(X \geq l) = \left(\frac{1}{5}\right)^l$ if $l \leq n^4$.
   $Prob(X \geq l) = 0$ if $l > n^4$.

$$\begin{aligned} E(X) &= \sum_{l=1}^{\infty} Prob(X \geq l) = \sum_{l=1}^{n^4} Prob(X \geq l) = \sum_{l=1}^{n^4} \left(\frac{1}{5}\right)^l \\ &\leq \sum_{l=1}^{\infty} \left(\frac{1}{5}\right)^l = \frac{1}{1-\frac{1}{5}} - 1 = \frac{1}{4} \in O(1), \\ E(X) &\geq \frac{1}{5} \in \Omega(1). \end{aligned} \tag{2}$$

   Thus $E(X) \in \Theta(1)$. Hence $ET(n) = clog_2(n)(c_2+1) + c_1 \in \Theta(log_2(n))$.

2. Steps 3-7 take $clog_2(n)$ time as computed in the above problem.
   (a) In the worst case, $k < n/5$, so we have:

$$T(n) = \sum_{i=1}^{\sqrt{n}} clog_2(n) = n^{1/2} clog_2(n) \in \Theta(n^{1/2} log_2(n)). \tag{3}$$

   (b) $Prob(k < n/5) = \lfloor n/5 \rfloor / n \approx 1/5$.
   Let X be the number of times $k < n/5$.
   The running time is given by $clog_2(n)(X+1)+c_1$ since the steps whose time is given by $clog_2(n)$ execute before the if-return-statement.
   So the expected running time is given by: $ET(n) = E(clog_2(n)(X+1)+c_1) = clog_2(n)(E(X)+1)+c_1$ by linearity of expectation.
   $E(X) = \sum_{l=1}^{\infty} Prob(X \geq l)$.
   $Prob(X \geq l) = \left(\frac{1}{5}\right)^l$ if $l \leq \lfloor n^{1/2} \rfloor$.
   $Prob(X \geq l) = 0$ if $l > \lfloor n^{1/2} \rfloor$.

Date: AU17.

So we have:

$$\begin{aligned}
E(X) &= \sum_{l=1}^{\infty} Prob(X \geq l) = \sum_{l=1}^{\lfloor n^{1/2} \rfloor} Prob(X \geq l) = \sum_{l=1}^{\lfloor n^{1/2} \rfloor} \left(\frac{1}{5}\right)^l \\
&\leq \sum_{l=1}^{\infty} \left(\frac{1}{5}\right)^l = \frac{1}{4} \in O(1), \\
E(X) &= \sum_{l=1}^{\lfloor n^{1/2} \rfloor} \left(\frac{1}{5}\right)^l \geq \frac{1}{5} \in \Omega(1).
\end{aligned} \tag{4}$$

Thus $E(X) \in \Theta(1)$, so let $E(X) = c_2$. Then we have:

$$ET(n) = clog_2(n)(c_2 + 1) + c_1 \in \Theta(log_2(n)). \tag{5}$$

3. Steps 1-11 take $cn^{2.5}$ time.
   (a) In the worst case, $k = n - 1$. So we have:

$$\begin{aligned}
T(n) &= cn^{2.5} + T(n-1) \\
&= cn^{2.5} + c(n-1)^{2.5} + T(n-2) \\
&= cn^{2.5} + c(n-1)^{2.5} + c(n-2)^{2.5} + \cdots + T(0) \\
&\leq cn^{2.5} + cn^{2.5} + cn^{2.5} + \cdots + T(0) \\
&= ncn^{2.5} = cn^{3.5} \in O(n^{3.5}), \\
T(n) &\geq cn^{2.5} + c(n-1)^{2.5} + c(n-2)^{2.5} + \cdots + c(n - \frac{n}{2})^{2.5} \\
&\geq c\left(\frac{n}{2}\right)^{2.5} + c\left(\frac{n}{2}\right)^{2.5} + \cdots + c\left(\frac{n}{2}\right)^{2.5} \\
&= \frac{n}{2} c \left(\frac{n}{2}\right)^{2.5} \in \Omega(n^{3.5}).
\end{aligned} \tag{6}$$

   So $T(n) \in \Theta(n^{3.5})$.
   (b) $Prob(k = l) = \frac{1}{n-1}$ for $1 \leq l \leq n - 1$.
   $Prob(k \leq n/2) = \frac{1}{2}$.
   $Prob(k > n/2) = \frac{1}{2}$.
   $ET(k \leq n/2) \leq ET(k = n/2) = cn^{2.5} + ET(n/2)$.
   $ET(k > n/2) \leq ET(k = n - 1) = cn^{2.5} + ET(n - 1)$.

$$\begin{aligned}
ET(n) &= Prob(k \leq n/2)Time(k \leq n/2) + Prob(k > n/2)Time(k > n/2) \\
&= \frac{1}{2}ET(k \leq n/2) + \frac{1}{2}ET(k > n/2) \\
&\leq \frac{1}{2}(cn^{2.5} + ET(n/2)) + \frac{1}{2}(cn^{2.5} + ET(n-1)) \\
&= cn^{2.5} + \frac{1}{2}ET(n/2) + \frac{1}{2}ET(n-1) \\
&\leq cn^{2.5} + \frac{1}{2}ET(n/2) + \frac{1}{2}ET(n) \\
ET(n) - \frac{1}{2}ET(n) &\leq cn^{2.5} + \frac{1}{2}ET(n/2) \\
\frac{1}{2}ET(n) &\leq cn^{2.5} + \frac{1}{2}ET(n/2) \\
ET(n) &\leq 2cn^{2.5} + ET(n/2) \\
&= c_2 n^{2.5} + ET(n/2) \\
&= c_2 n^{2.5} + \frac{1}{2}c_2 n^{2.5} + \frac{1}{2^2}c_2 n^{2.5} + \cdots + \frac{1}{2^{log_2(n)}}ET(1) \\
&\leq c_2 n^{2.5}\left(1 + \frac{1}{2} + \frac{1}{2^2} + \cdots\right) \\
&= 2c_2 n^{2.5} \in O(n^{2.5}).
\end{aligned} \tag{7}$$

And as a lower bound we take $k = 1$, then the function takes $cn^{2.5}$ time, so $ET(n)\Omega(n^{2.5})$. Hence $ET(n) \in \Theta(n^{2.5})$.

4. Steps 1-7 take cn time.
   (a) In the worst case, $k = 1$, so step 8 takes c_1 time, and we have:

$$\begin{aligned}
T(n) &= cn + +c_1 + T(n-1) \\
&\approx cn + T(n-1) \\
&= cn + c(n-1) + c(n-2) + \cdots + T(0) \\
&\leq cn + cn + cn + \cdots + T(0) \\
&= n(cn) \in O(n^2), \\
T(n) &\geq cn + c(n-1) + c(n-2) + \cdots + c(n - \frac{n}{2}) \\
&\geq c\frac{n}{2} + c\frac{n}{2} + \cdots + c\frac{n}{2} \\
&= \frac{n}{2}c\frac{n}{2} \in \Omega(n^2).
\end{aligned} \tag{8}$$

   So $T(n) \in \Theta(n^2)$.

   (b) In the best case, $k = n/2$, so we have:

$$\begin{aligned}
ET(n) &\geq cn + ET(n/2) + ET(n/2) \\
&= cn + 2ET(n/2) \in \Omega(nlog_2(n)).
\end{aligned} \tag{9}$$

   We find an upper bound for the expected running time.
   $ET(n) = Prob(k < n/4)ET(k < n/4) + Prob(k \geq n/4)ET(k \geq n/4)$.
   $Prob(k < n/4) = Prob(k \geq n/4) = \frac{1}{2}$.
   $ET(k < n/4) \leq ET(n = 1) = cn + ET(n-1)$.
   $ET(k \geq n/4) \leq ET(n = n/4) = cn + ET(n/4) + ET(3n/4)$.

Thus:

$$\begin{aligned} ET(n) &\leq \frac{1}{2}(cn + ET(n-1)) + \frac{1}{2}(cn + ET(n/4) + ET(3n/4)) \\ &\leq cn + \frac{1}{2}ET(n) + \frac{1}{2}(ET(n/4) + ET(3n/4)) \\ ET(n) - \frac{1}{2}ET(n) &\leq cn + \frac{1}{2}(ET(n/4) + ET(3n/4)) \\ \frac{1}{2}ET(n) &\leq cn + \frac{1}{2}(ET(n/4) + ET(3n/4)) \\ ET(n) &\leq c_2 n + ET(n/4) + ET(3n/4). \end{aligned} \tag{10}$$

Now we use a recursion tree. Each node has two children for each of the two recursive calls. The sum over each level is cn, and the height of the tree is the height of the slowest path, so it is $log_{4/3}(n)$, so $ET(n) \leq log_{4/3}(n)cn = cc_3 nlog_2(n) \in O(nlog_2(n))$. Hence $ET(n) \in \Theta(nlog_2(n))$.

5. Assume array has no duplicates.
$ET(0) = 0$.
After partition, $A[s] = p$. **What does this mean??**
Let $m = s - i + 1$. **why??**
After partition, p is m-th element of $A[i], A[i+1], ..., A[j]$.
Split into fourths, treat the two recursive calls like $Func(k), Func(n-k)$.
Check MAX's midterm to figure this one out, do we need to know it???
Note that this problem is equivalent to a function with a few lines taking cn time, and a random k being chosen between $n/10$ and $9n/10$, with two recursive calls, one being $T(k)$ and the other being $T(n-k)$. This is equivalent to a random k between 1 and n, with recursive calls $T(n/10)$, $T(9n/10)$. Note we only want the worst case!

In the worst case, choose $k = n/10$, but either extreme works due to symmetry. So we have:

$$\begin{aligned} T(n) &= cn + T(n/10) + T(n - n/10) \\ &= cn + T(n/10) + T(9n/10) \end{aligned} \tag{11}$$

Using a recursion tree, we see that each level adds cn to the sum. The height of the tree is the worst case which gives us $log_{10/9}(n)$ levels, and thus a time of $cnlog_{10/9}(n)$. The best case gives us $log_{10}(n)$ levels and hence a time of $cnlog_{10}(n)$. So $T(n) \in \Theta(nlog_2(n))$.