HW #4 - Problem 2

Ryan St.Pierre (ras70)

October 4, 2017

Problem 2

The recurrence relation of quick select is given by,

$$\mathbb{E}[X_n] = \frac{1}{n}\sum_{i=1}^{k-1}(\mathbb{E}[X_{n-i}] + An) + \frac{1}{n}\sum_{i=k+1}^{n}(\mathbb{E}[X_{i-1}] + An) + \frac{1}{n}An$$

Above, the first summation accounts for the case when $i < k$. In this case the algorithm partition to the right contains the k^{th} smallest number and is checked. The second summation accounts for the case when $i > k$. In this case the left partition is further analyzed. The final term ($\frac{1}{n}An$) accounts for the case when $i == k$. In this case the k^{th} smallest number has been found. Each term in the two summations and this final term ($\frac{1}{n}An$) each have an equal $\frac{1}{n}$ chance of occurring.

Before I prove by induction that this is bounded by Cn I will first simplify this recurrence relation and show it is that given in the homework sheet.

$$\begin{aligned}
\mathbb{E}[X_n] &= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}(\mathbb{E}[X_{n-i}] + An) + \tfrac{1}{n}\sum_{i=k+1}^{n}(\mathbb{E}[X_{i-1}] + An) + \tfrac{1}{n}An \\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}\mathbb{E}[X_{n-i}] + \tfrac{1}{n}\sum_{i=k+1}^{n}\mathbb{E}[X_{i-1}] + \tfrac{1}{n}\sum_{i=1}^{n}An + A \\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}\mathbb{E}[X_{n-i}] + \tfrac{1}{n}\sum_{i=k+1}^{n}\mathbb{E}[X_{i-1}] + An + A \\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}\mathbb{E}[X_{n-i}] + \tfrac{1}{n}\sum_{i=k+1}^{n}\mathbb{E}[X_{i-1}] + n + 1 && \text{A=1} \\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}\mathbb{E}[X_{n-i}] + \tfrac{1}{n}\sum_{i=k+1}^{n}\mathbb{E}[X_{i-1}] + n && \text{O(n+1)=O(n)}
\end{aligned}$$

This is the recurrence relation given in the homework sheet. This is the one that will be used in the following inductive proof.

Proof by Induction

Induction Hypothesis: $\mathbb{E}[X_k] \leq Cn$ for all $k < n$ where X_n is a r.v. corresponding to the running time of quick select on an array of size n.
Base cases:
n=1

$$\mathbb{E}[X_1] = A \leq 1 * A$$

In the case when there is only one element in the array the running time is constant. This is due to the fact that only simple comparisons are needed. If $n = 1$ the only valid value of k is 1. In other words, if the array is only of size 1, only the smallest number can be requested, **not** the second smallest number since there is only one number in the array. Thus, the algorithm only needs to compare k to one. If k is indeed one the algorithm returns the only element in the array in constant time, else it throws an error. Thus, if we call this constant return time A, a C can be chosen, $C = A$ more specifically, such that $\mathbb{E}[X_k] \leq Cn$ holds.

A similar argument for the case when $n = 0$ can also be made. In this case, when $n = 0$, the algorithm returns in constant time because there is no items in the array to return.

Inductive step:

Let $n > k$

$$\begin{aligned}
\mathbb{E}[X_n] &= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}\mathbb{E}[X_{n-i}] + \tfrac{1}{n}\sum_{i=k+1}^{n}\mathbb{E}[X_{i-1}] + n && * \\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{k-1}\mathbb{E}[X_{n-i}] + \tfrac{1}{n}\sum_{i=k+1}^{n}\mathbb{E}[X_{i-1}] + n && ** \\
&\leq \tfrac{C}{n}\textstyle\sum_{i=1}^{k-1}(n-i) + \tfrac{C}{n}\sum_{i=k+1}^{n}(i-1) + n \\
&\leq \tfrac{C}{n}(n\textstyle\sum_{i=1}^{k-1}1 - \sum_{i=1}^{k-1}i + \sum_{i=k+1}^{n}i - \sum_{i=k+1}^{n}1)) + n \\
&\leq \tfrac{C}{n}(n(k-1) - \tfrac{(k-1)(k)}{2} + \textstyle\sum_{i=1}^{n}i - \sum_{i=1}^{k}i - (n-k)) + n \\
&\leq \tfrac{C}{n}(n(k-1) - \tfrac{(k-1)(k)}{2} + \tfrac{(n)(n+1)}{2} - \tfrac{(k)(k+1)}{2} - n + k) + n \\
&\leq \tfrac{C}{n}(nk - 2n - \tfrac{(k-1)(k)}{2} + \tfrac{(n)(n+1)}{2} - \tfrac{(k)(k+1)}{2} + k) + n \\
&\leq \tfrac{C}{2n}(2nk - 4n - (k-1)(k) + (n)(n+1) - (k)(k+1) + 2k) + n \\
&\leq \tfrac{C}{2n}(2nk - 4n + 2k - k^2 + k + n^2 + n - k^2 - k) + n \\
&\leq \tfrac{C}{2n}(2nk - 3n + 2k - 2k^2 + n^2) + n
\end{aligned}$$

* By recurrence relation
** By Induction Hypothesis

At this point we have proven

$$\mathbb{E}[X_n] \leq \frac{C}{2n}f(k) + n$$

where $f(x) = 2nk - 3n + 2k - 2k^2 + n^2$.

If $\mathbb{E}[X_n]$ is strictly less than a function for all values of k it must also be less than that function when it is at its maximum. We can find this maximum by finding the critical point of $f(k)$. This is equivalent to thinking of the running time of the algorithm in the worst case, when k has been chosen to maximize the expected running time.

Critical Points of f(k)

$$\begin{aligned} f(k) &= 2nk - 3n + 2k - 2k^2 + n^2 \\ f'(k) &= 2n + 2 - 4k \end{aligned}$$

Setting $f'(k)$ equal to zero to find the critical points yields,

$$\begin{aligned} 2n + 2 - 4k &= 0 \\ 4k = 2n + 2 & \\ k = \tfrac{n+1}{2} & \end{aligned}$$

Since $f(k)$ is a downward facing parabola, this critical point must be a maximum. Plugging this expression of k into $\mathbb{E}[X_n]$ gives,

$$\begin{aligned} \mathbb{E}[X_n] &\leq \tfrac{C}{2n}(n^2 - 2(\tfrac{n+1}{2})^2 + n(\tfrac{n+1}{2}) - 3n + 2n\tfrac{n+1}{2}) + n \\ &\leq \tfrac{C}{2n}(n^2 - \tfrac{n^2+2n+1}{2} + n + 1 - 3n + n^2 + n) + n \\ &\leq \tfrac{C}{4n}(2n^2 - n^2 - 2n - 1 + 2n + 2 - 6n + 2n^2 + 2n) + n \\ &\leq \tfrac{C}{4n}(3n^2 - 4n + 1) + n \\ &\leq \tfrac{3Cn}{4} + n - C + \tfrac{C}{4n} \end{aligned}$$

Since n and C are positive numbers $-C + \frac{C}{4n}$ is a negative quantity.This means $\frac{3Cn}{4} + n - C + \frac{C}{4n} < \frac{3Cn}{4} + n$. Therefore,

$$\mathbb{E}[X_n] \leq \frac{3Cn}{4} + n$$

Now it needs to be shown that there is some value of C such $\mathbb{E}[X_n]$ is bounded by Cn. More specifically, a value of C greater than needs to be found such that the following holds:

$$\mathbb{E}[X_n] \leq \frac{3Cn}{4} + n \leq Cn$$

Any value of $C \geq 4$ makes this above inequality hold. The computation for this is shown below.

$$\begin{aligned}
\tfrac{3Cn}{4} + n &\leq Cn \\
n(\tfrac{3C}{4} + 1) &\leq Cn \\
\tfrac{3C}{4} + 1 &\leq C \\
1 &\leq C - \tfrac{3C}{4} \\
1 &\leq \tfrac{1}{4}C \\
4 &\leq C \\
C &\geq 4
\end{aligned}$$

Let's chose $C = 4$. By induction we have $\mathbb{E}[X_n] \leq 4n = O(n)$