cs361 Assignment 3

Ryan Scherbarth

February 21, 2024

1 Average Running time of Quick Select

Given that the average running time of quick select is:

$$\mathbf{T(n)} = (n-1) + \frac{2}{n}\sum_{j=\frac{n}{2}}^{n-1}\mathbf{T(j)}$$

Assuming we have an even distribution, we can prove by induction.

1.1 Base Case

Let $n = 2$, and let our guess function be $4n$:

$$T(2) = (2-1) + \frac{2}{2}\sum_{j=\frac{2}{2}}^{2-1}T(j)$$

$$T(2) = 1\sum_{j=1}^{1}T(j)$$

$$T(2) = 1 + 1 = 2$$

$$\mathbf{T(2)} = 2 \le 8$$

1.2 Inductive Step

We next need to prove that $n = k+1$ to be true:

$$T(k+1) = (k+1-1) + \frac{2}{k+1}\sum_{j=\frac{k+1}{2}}^{k+1-1}T(j)$$

$$T(k+1) = (k) + \frac{2}{k+1}\sum_{j=\frac{k+1}{2}}^{k}T(j)$$

$$T(k+1) = k + \frac{8}{k+1}(\sum_{j=1}^{k} j - \sum_{j=1}^{\frac{k+1}{2}-1} j)$$

$$T(k+1) = k + \frac{8}{k+1}(\frac{k(k+1)}{2} + \frac{\frac{k+1}{2}(\frac{k+1}{2}-1)}{2})$$

$$T(k+1) = k + \frac{8}{k+1}(\frac{k^2+k}{2} - \frac{k^2-k+2}{4})$$

$$T(k+1) = k + \frac{8}{k+1}(\frac{k^2+3k-2}{4})$$

$$T(k+1) = k + \frac{2k^2+6k-4}{k+1}$$

$$T(k+1) = k + \frac{2k^2+6k}{k+1} - \frac{4}{k+1} => k + \frac{2(k+1)}{k+1} - \frac{4}{k+1}$$

$$T(k+1) = 3k - \frac{4}{k+1}$$

The induction step holds true so long as it is true that $T(k+1) \leq 4n$

$$3k - \frac{4}{k+1} \leq 4(k+1)$$

$$-\frac{4}{k+1} \leq 4k+4$$

$$-4 \leq k^2 + 5k + 4$$

$$0 \leq k^2 + 5k + 8$$

Holds true for all $k \geq w$, Therefore, via proof by induction, $T(n) =\leq 4n$ for all $n > 1$.

2 Average Running Time of Quicksort

2.1 Average Running Time

The recurrence equation for the average running time of quicksort, assuming an even distribution, is as follows:

$$T(n) = n + \frac{1}{n}\sum_{i=0}^{n-1}(T(i) + T(n-i-1))$$

While similar to the Quick Select Time function, when using Quicksort we also must keep in mind that each time we split an array into a sub-array, the sizes of both these arrays will depend on each other. In Quick Select we can simply just ignore whatever was not in the new sub-array we chose to continue searching in after creating each partition, But that doesn't work in Quicksort. We still need

to come back and sort both sides after creating a partition. For this reason, we slightly modify the time function to track both sub-arrays based off of the other.
I've come to this guess function by noting that our running time has two dependent factors; Firstly we have the partitioning step, which is what gives us our n linear time. After making all of the partitions, we just need to merge these partitions together, which gives us $log(n)$ logarithmic time.

2.2 Guess function

Here, I chose to use a guess function of:

$$f(n) = n * log(n)$$

I chose this because

2.3 Proof by Induction

2.3.1 Base Case

Let $n = 1$,

$$T(1) = 1 * log(1) = 0$$

In our base step, we need to prove that $T(n) \leq c * f(n)$ for any n, Thus, our base case is trivially true that $T(1) \leq c * f(n)$, simplifying to $0 \leq 0$.

2.3.2 Inductive Step

To prove our inductive step, we need to prove $T(m) \leq c * f(m)$ for all $m < n$

$$T(n) = n + \frac{1}{n}\sum_{i=0}^{n-1}(T(i) + T(n-i-1))$$

$$T(n) \leq n + \frac{c}{n}\sum_{i=0}^{n-1}(f(i) + f(n-i-1))$$

$$T(n) \leq n + \frac{c}{n}\sum_{i=0}^{n-1}(2f(i))$$

$$T(n) \leq n + \frac{2c}{n}\sum_{i=0}^{n-1}(f(i))$$

$$T(n) \leq n + \frac{2c}{n} * n * log(n)$$
$$T(n) \leq n + 2c * log(n)$$

$$T(n) \leq 3c * log(n)$$

Therefore, via proof by induction, we have proven that Quicksort has an average running time, when using even distribution, and a guess function of $f(n) = n * log(n)$, of $T(n) = n * log(n)$. Thus, $T(n) = O(f(n))$