CSCI-GA.1170-003/004 Fundamental Algorithms February 28, 2020

Solutions to Problem 2 of Homework 4 (12 points)

Name: Rahul Zalkikar (rz1567) *Due: 8 pm on Thursday, February 27*

Collaborators: ua388, jni215

Assume we are given an array $A[1 \dots n]$ of n *distinct* integers and that $n = 2k$ is *even.*

(a) (4 points) Let $pivot(A)$ denote the rank of the pivot element at the end of the partition procedure, and assume that we choose a random element $A[i]$ as a pivot, so that $pivot(A) = i$ with probability $1/n$, for all i. Let $smallest(A)$ be the length of the smaller sub-array in the two recursive subcalls of the QUICKSORT. Notice, $smallest(A) = \min(pivot(A) - 1, n - pivot(A))$ and belongs to $\{0 \dots k-1\}$, since $n = 2k$ is even. Given $0 \le j \le k-1$, what is the probability that $smallest(A) = j$?

Solution:

The $smallest(A)$ will belong to $\{0 \dots k-1\}$ because the $smallest(A)$ is either belonging to the larger or smaller subarray as divided by the pivot. In addition, the max size of the smaller subarray is $k-1$, and by taking the minimum of the two possible pivot positions we also see that $smallest(A)$ will belong to $\{0 \dots k-1\}$. So, for some $j \ge 0$ and $j \le k-1$ we have positions $n-j-1$ and $j+1$ making the $smallest(A) = j$. Now since we choose an element at random with probability $\frac{1}{n}$ we can see that $smallest(A) = j = \frac{2}{n}$. □

(b) (3 points) Compute the *expected value* of $smallest(A)$; i.e., $\sum_{j=0}^{k-1} \Pr[smallest(A) = j] \cdot j$. (**Hint**: If you solve part (a) correctly, no big computation is needed here.)

Solution:

$\sum_{j=0}^{k-1} j(P(smallest(A) = j)) = \sum_{j=0}^{k-1} j(\frac{2}{n})$

$= \frac{2}{n}(0 + 1 + ... + (k-1)) = \frac{k \times (k-1)}{n} = \frac{\frac{n}{2} \times (\frac{n}{2} - 1)}{n} = \frac{n \times (n-2)}{4n} = \frac{n-2}{4}$ □

(c) (5 points) Write a recurrence equation for the running time $T(n)$ of QUICKSORT, assuming that at every level of the recursion the corresponding sub-arrays of A are partitioned *exactly* in the ratio you computed in part (b). Solve the resulting recurrence equation. Is it still as good as the average case of randomized QUICKSORT?

Solution:

$T(n) = T(\frac{n}{4} - \frac{1}{2}) + T(n - \frac{n}{4} - 1 + \frac{1}{2}) + \Theta(n)$

$T(n) = T(\frac{n}{4} - \frac{1}{2}) + T(\frac{3n}{4} - \frac{1}{2}) + \Theta(n)$

$T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + \Theta(n)$ (ignore the constants)

Now we have a recurrence that is smaller and larger on each side.

For the longer branch we have $T_{long}(n) \leq cn(\log_{\frac{4}{3}}(n))$

For the shorter branch we have $T_{short}(n) = cn(\log_4(n))$

We know that $T_{short}(n) \leq T(n) \leq T_{long}(n)$

$cn(\log_4(n)) \leq T(n) \leq cn(\log_{\frac{4}{3}}(n))$

$cn(\frac{\log(n)}{\log(4)}) \leq T(n) \leq cn(\frac{\log(n)}{\log(\frac{4}{3})})$

$cn(\frac{\log(n)}{\log(4)}) \leq T(n) \leq cn(\frac{\log(n)}{\log(4)-\log(3)})$

$cn(\log(n))(\frac{1}{\log(4)}) \leq T(n) \leq cn(\frac{1}{\log(4)-\log(3)})$

$b_1 cn(\log(n)) \leq T(n) \leq b_2 cn(\log(n))$

We know there exists constants c, b_1, b_2 s.t. the above is satisfied, we can see that $T(n) = \Theta(n\log(n))$.

We note that the average running time for Randomized-Quicksort is $O(n\log(n))$, which is better than the $\Theta(n\log(n))$ running time for Quicksort. $\square$