CSCI-GA.1170-001/002 Fundamental Algorithms	October 6, 2015

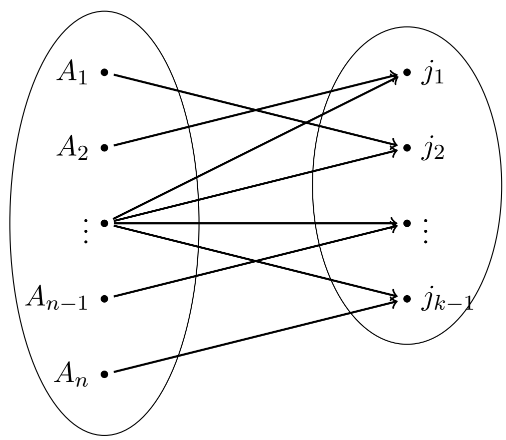
Solutions to Problem 3 of Homework 4 (12 points)

Name: GOWTHAM GOLI (N17656180) *Due: Tuesday, October 6*

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 $0 \leq j \leq k-1$, what is the probability that $smallest(A) = j$?

Solution: Given that the probability to choose a random element $A[i]$ as a pivot is $1/n$. It is clear that there exists a pair $(A_i, A_j) \leq n$ such that $A[i]$ and $A[j]$ will result in the same *smallest(A)* when chosen as a pivot. Therefore, there will be total $n/2$ pairs of (A_i, A_j) that will result in the same *smallest(A)* when chosen as a pivot which is illustrated in the figure below (j_k denotes the case when $j = k$)



As we know that the probability of selecting any A_i as pivot is $1/n$. The probability that $smallest(A)$ will be $2/n$ (as two elements from the Pivot set point to one element in the Smallest set as seen above).

$$\Pr(smallest(A) = j) = 2/n$$

$\square$

(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:

$$\begin{aligned}
\sum_{j=0}^{k-1} \Pr(smallest(A) = j) \cdot j &= \sum_{j=0}^{k-1} 2/n.j \\
&= 2/n \sum_{j=0}^{k-1} j \\
&= \frac{2}{n} . \frac{k.(k-1)}{2} \\
&= \frac{n-2}{4}
\end{aligned}$$

□

(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:

For relatively large n we can assume that $n - 2/4 \approxeq n/4$. Therefore at every level of the recursion the array is partitioned into sub arrays of length $n/4$ and $3n/4$.

$$T(n) = T(n/4) + T(3n/4) + O(n)$$

This recurrence equation can be solved using Recursive tree method as shown below in the figure

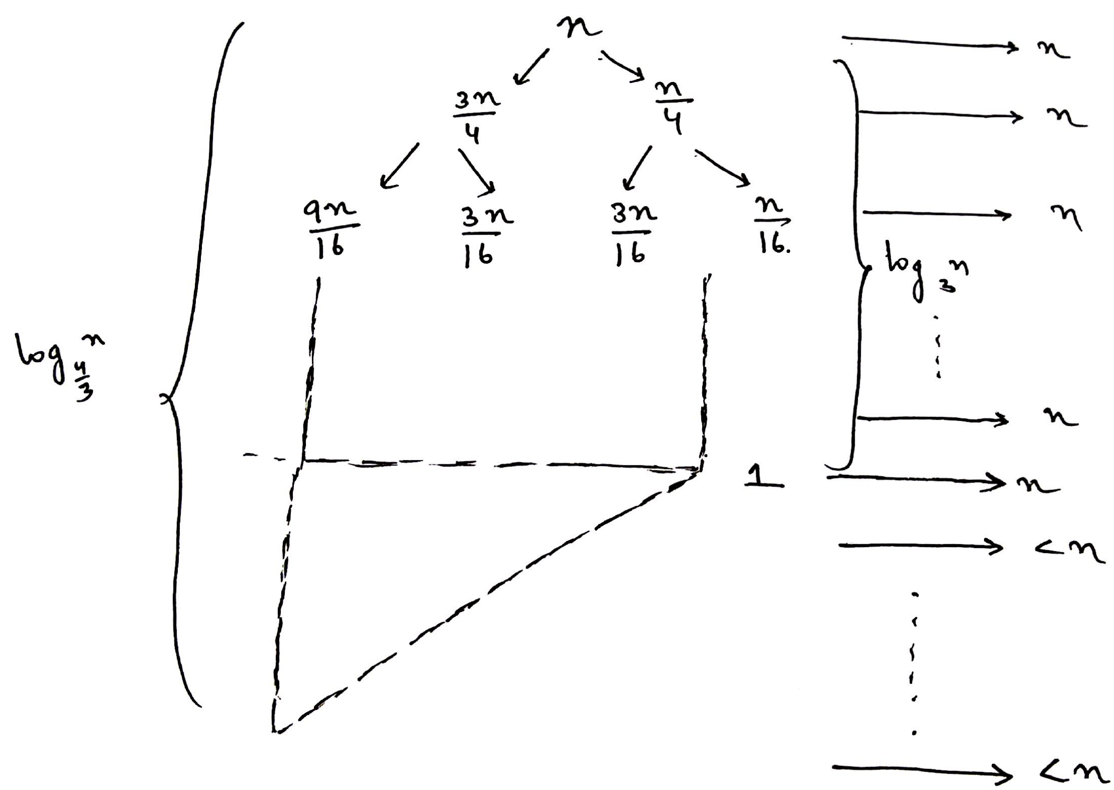


Figure 1: Recursive Tree of $T(n)$

$$
\begin{aligned}
n \log_4 n &\leq T(n) \leq n \log_{4/3} n \\
n \frac{\log_2 n}{\log_2 4} &\leq n \leq \frac{\log_2 n}{\log_2 4/3} \\
\implies T(n) &= \Theta(n \log n)
\end{aligned}
$$

We can see that it as good as the average case of randomized quick sort $\square$