ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 335E

ANALYSIS OF ALGORITHMS I
ASSIGNMENT I REPORT

STUDENT NAME	:	Fatma Sena Akçağlayan
STUDENT ID	:	150170085
ASSINGMENT DATE	:	25.11.2021

FALL 2021

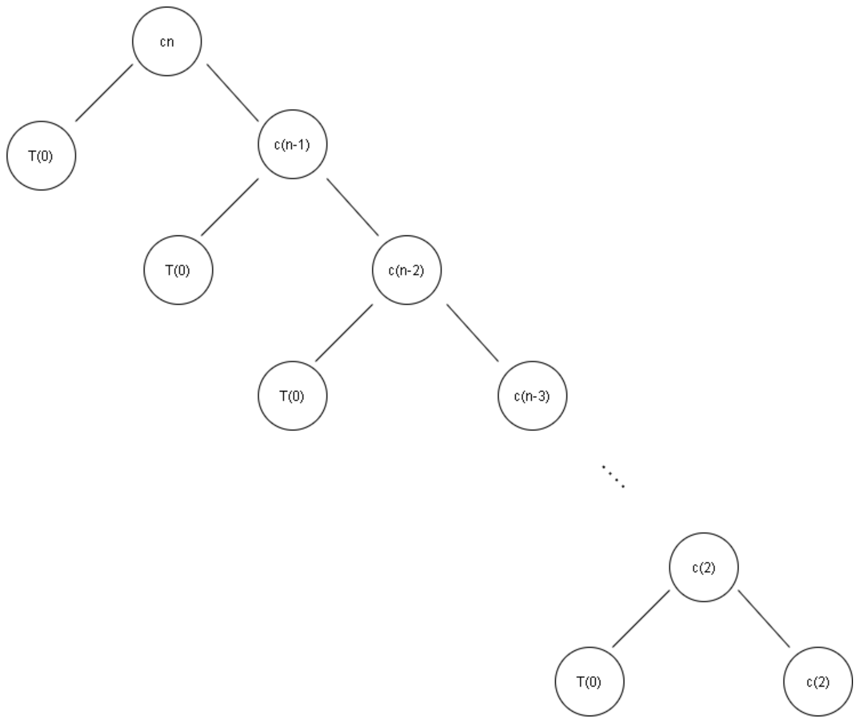
Contents

1 Analysis of The Quicksort with Deterministic Pivot

$\boldsymbol{WorstCase}$

Let T(n) is the worst case running time on an array of n elements. Input array is already sorted in ascending or descending order in the worst case. So, one side of partition always has no elements.

$T(n) = T(0) + T(n-1) + cn$



As shown in Worst case recursion tree, Total partitioning time:

$T(n) = cn + c(n-1) + c(n-2) + ... + 2c$

$= c(n + (n-1) + (n-2) + ... + 2)$

$= c\sum_{n=1} -1$

$= c(\frac{n(n+1)}{2} - 1) = \Theta(n^2)$

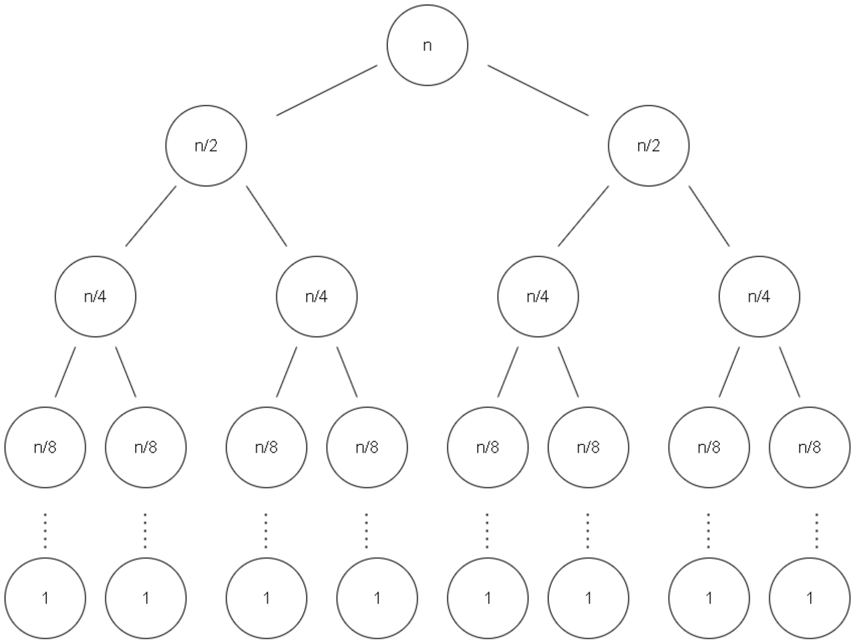
Following is recurrence for worst case:

$T(n) = T(0) + T(n-1) + \Theta(n) = T(n-1) + \Theta(n)$

$= \Theta(n^2) + \Theta(n) = \Theta(n^2)$

BestCase

If partitions are evenly splits, best case occurs. Similar to $T(n)$ Merge Sort
$T(n) = 2T(n/2) + \Theta(n)$



According to above tree, the values from all steps are summed
$T(n) = cn + 2(cn/2) + 4(cn/4) + ... + nc$
$= cn + cn + cn + ... + cn$
$= \Theta(nlogn)$

2 Analysis of The Quicksort with Randomized Pivot

Assume the random numbers are independent and input size equals to n. $T(n)$ is the random variable for the running time of randomized Quicksort.
The indicator random variable is defined for $k = 0, 1, ..., n-1$.
If Partition generates a $k : n-k-1$ split, $X_k = 1$ and $X_k = 0$ otherwise.

$E[X_k] = PrX_k == \frac{1}{n}$ assume that all elements are distinct and all splits are equally likely.

$T(0) + T(n-1) + \Theta(n)$ if $0 : n-1$ split,
$T(1) + T(n-2) + \Theta(n)$ if $1 : n-2$ split,
...
$T(n-1) + T(0) + \Theta(n)$ if $n-1 : 0$ split,
so,
$T(n) = \sum_{k=0}^{n-1} X_k(T(k) + T(n-k-1) + \Theta(n))$
To calculate expectation,
$E[T(n)] = E[\sum_{k=0}^{n-1} X_k(T(k) + T(n-k-1) + \Theta(n))]$
According to Linearity of expectation,
$E[T(n)] = \sum_{k=0}^{n-1} E[X_k(T(k) + T(n-k-1) + \Theta(n))]$
X independence from other random choices, so
$E[T(n)] = \sum_{k=0}^{n-1} E[X_k]E[T(k) + T(n-k-1) + \Theta(n)]$

$E[X_k] = \frac{1}{n}$
$E[T(n)] = \frac{1}{n}\sum_{k=0}^{n-1} E[T(k)] + \frac{1}{n}\sum_{k=0}^{n-1} E[T(n-k-1)] + \frac{1}{n}\sum_{k=0}^{n-1} E[\Theta(n)]$
$= \frac{2}{n}\sum_{k=1}^{n-1} E[T(k)] + \Theta(n)$
$= E[T(n)] \leq anlgn$ for constant $a > 0, n \geq 2$
$= E[T(n)] \leq \frac{2}{n}\sum_{k=2}^{n-1} aklogk + \Theta(n)$

$= \frac{2a}{n}(\frac{1}{2}n^2logn - \frac{1}{8}n^2) + \Theta(n)$
$= anlogn - (\frac{an}{4})$ - $\Theta(n))$
$\leq anlogn$
$an/4$ dominates the $\Theta(n)$ when a is chosen large enouh. Therefore, $T(n) = \Theta(nlogn)$

3 N - runtime Relation for Unsorted Array

	N=1000	N=10K	N=100K	N=500K	N=1M
deterministic	3,13ms	46,88ms	690,63ms	5287,54ms	10090,40ms
randomized	6,25ms	71,42ms	817,19ms	5943,75ms	12937ms

Table 1: Average running times for different N values

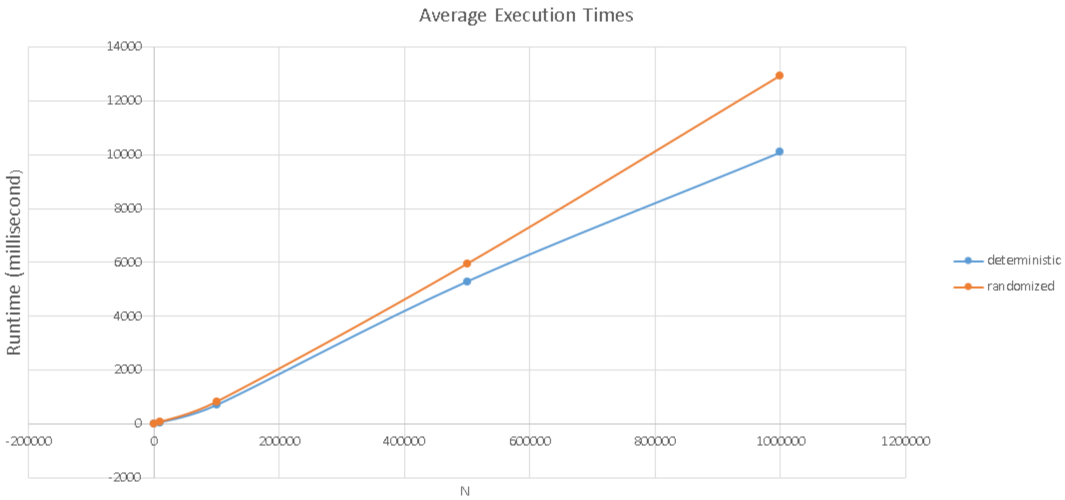


Figure 1: N - runtime for unsorted array

As N increases, the running time also increases. When the Figure3 is examined, it is seen that the running time increases according to the $n.logn$ function. In Question 1 and Question 2, the best case time and the average case time were found $n.logn$. Accordingly, the given sequence is close to the best case for deterministic pivot selection and close to the average case for random pivot selection. It can also be said that choosing a random pivot is not an advantage in such a case.

4 N - runtime Relation for Sorted Array

	N=1000	N=10K	N=100K	N=500K	N=1M
deterministic	156,25ms	16971ms	2500000ms	15000000ms	316000000ms
randomized	6,25ms	62ms	1000ms	4128ms	8609,38ms

Table 2: Average running times for different N values

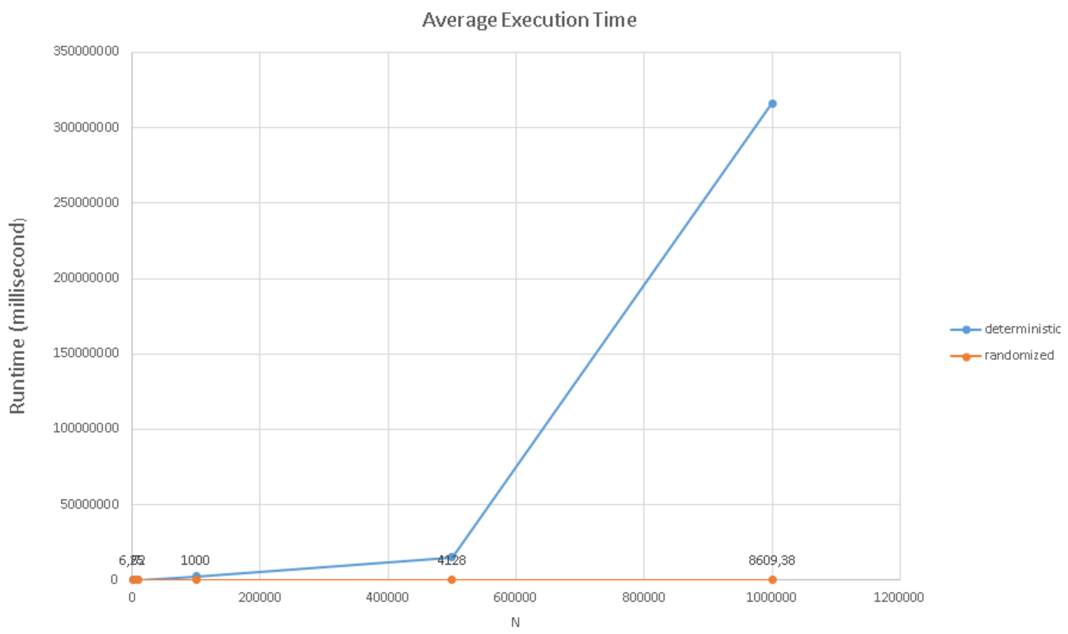


Figure 2: N - runtime for sorted array

Sorting a sorted array with deterministic Quicksort is the worst case. In this case, one side of partition always has no elements and other side has all elements. As shown in Figure4, the running time of the Quicksort with a deterministic pivot is similar to the function with n^2. Same as the worst case running time found in Quiestion 1. However, when Quicksort is run with a random pivot selected, the selected array is close to the best case and average case, and the results are according to the $n.logn$ function. As you can see, a deterministic Quicksort gives a good result for an unsorted array, but a quite bad result when trying to sort a sorted array. Randomized Quicksort, on the other hand, ran fast regardless of the entered data string. In an array containing ordered data, the random pivot should definitely be chosen for Quicksort.

5 Analysis of Dual Pivot Quicksort Algorithm

For dual pivot Quicksort, the array is split into three. In this case, the following equation is written for the value of $T(n)$:
$T(n) = 3T(n/3) + cn$

The Master Theorem can be used to find the asymptotic upper bound. Since f(n) is equal to cn, the Case 2 of the $Master Theorem$ is applied. In Case 2, the best case for $a = 3$ and $b = 3$ is $T(n) = O(nlogn)$. Since the logarithm is in base 3, it will run faster than the normal Quicksort. In addition, since it proceeds by dividing it into three, the probability of being the worst case decreases.