İremsu Kardaş

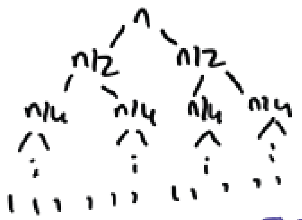
150180081

Assignment 1-Report

1) Deterministic Quick Sort Analysis

- Best case

Each subarray contains n/2 of elements



n

$2 \cdot n/2 = n$

$4 \cdot n/4 = n$

$\vdots$

$n \cdot 1 = n$

$\Big\} \log_2 n = O(\log_2 n \cdot n) = O(n\log n)$

Solving with Master Theorem

$T(n) = T(1/2) + T(1/2) + O(n)$

$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$

$a=2 \quad b=2 \quad d=1 \qquad a=b^d \ (2=2^1) \ (\text{case } 1)$

$T(n) = O(n^1 \log(n)) = O(n\log n)$

- Worst case

Each partition is done, subarray has n-1 elements from previous.

Solving with iteration

$T(n) = O(n) + O(n-1) \ldots O(1)$

$= \sum_{k=1}^{n} O(k) = O\left(\frac{n \cdot n+1}{2}\right) = O(n^2) = O(n^2)$

2) The dominant cost is when doing partitioning.

X_{ij} = z_i is compared to z_j

X = total number of comparison $= X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$

Take Expectation

$$E[X] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right]$$

$Pr[z_i$ is compared to $z_j] = Pr[z_i$ or z_j is the first pivot chosen from $Z_{ij}]$

$$= \frac{1}{j-i+1} + \frac{1}{j-i+1} = \frac{2}{j-i+1}$$

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \quad (k = j-i)$$

$$= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} < \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} = \sum_{i=1}^{n-1} O(\log n) = O(n\log n)$$

→ So randomized quicksort expected running time is $O(n\log n)$

Worst case

$$T(n) = \max_{0 \le q \le n-1} \left(T(q) + T(n-q-1)\right) + \Theta(n)$$

$T(n) \le cn^2$

$$T(n) \le \max_{0 \le q \le n-1} \left(cq^2 + c(n-q-1)^2 + \Theta(n)\right)$$

$$\max_{0 \le q \le n-1} \left(q^2 + (n-q-1)^2\right) \le (n-1)^2 \quad (q \text{ is positive})$$

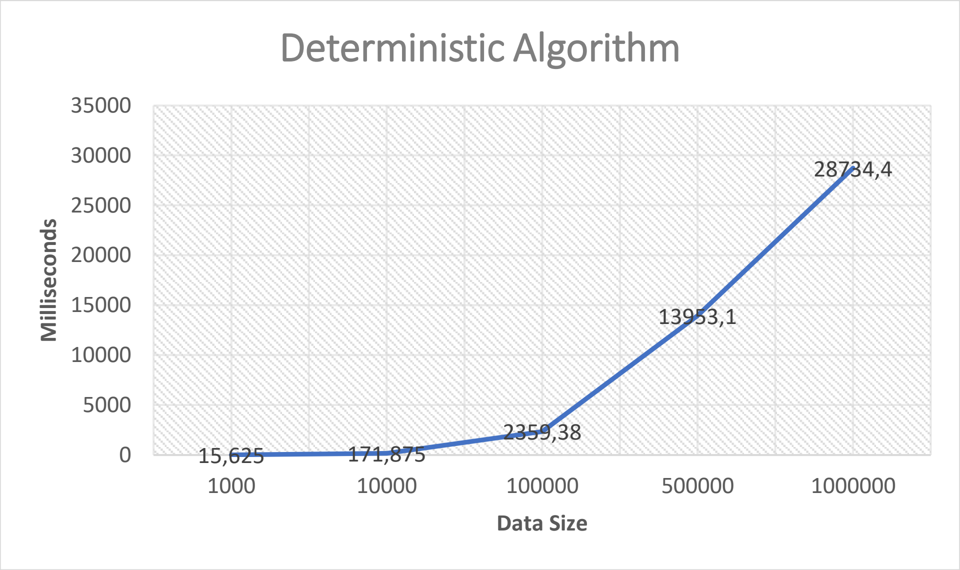
$$= n^2 - 2n + 1$$

Worst case $\Theta(n^2)$

3)

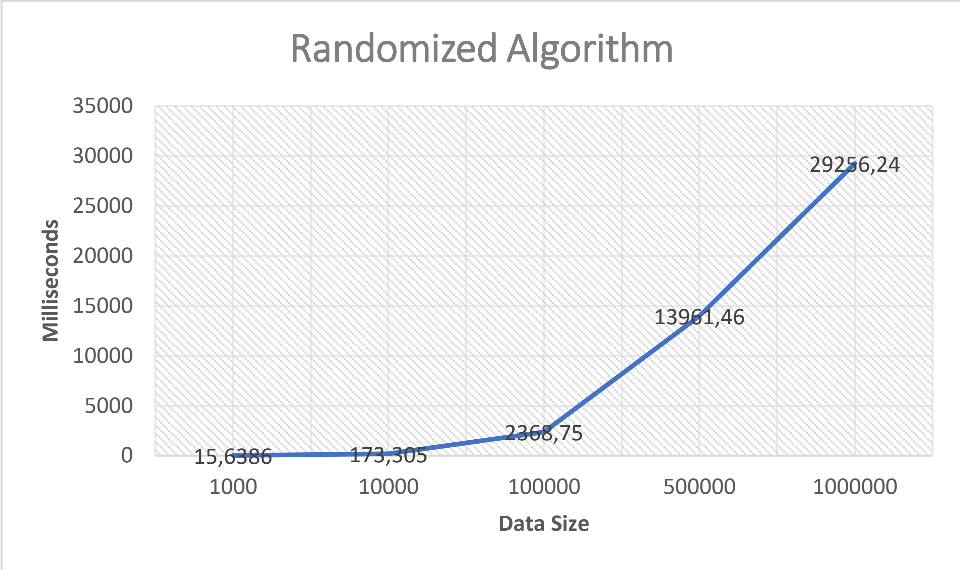
There can be seen in the tables and plot run time milliseconds for deterministic algorithm depends on the how many tweets will be sorted.

Data Size	Milliseconds
1000	15,625
10000	171,875
100000	2359,38
500000	13953,1
1000000	28734,4

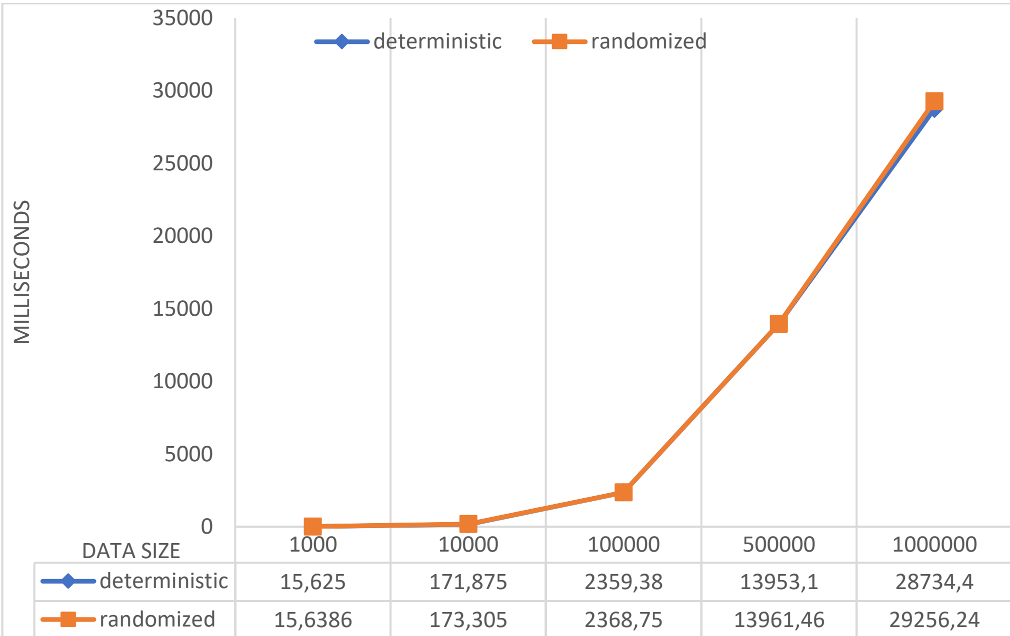


For randomized algorithm average milliseconds (for 5 times) and how many tweet will be sorted can be seen below.

Data Size	Milliseconds
1000	15,6386
10000	173,305
100000	2368,75
500000	13961,46
1000000	29256,24



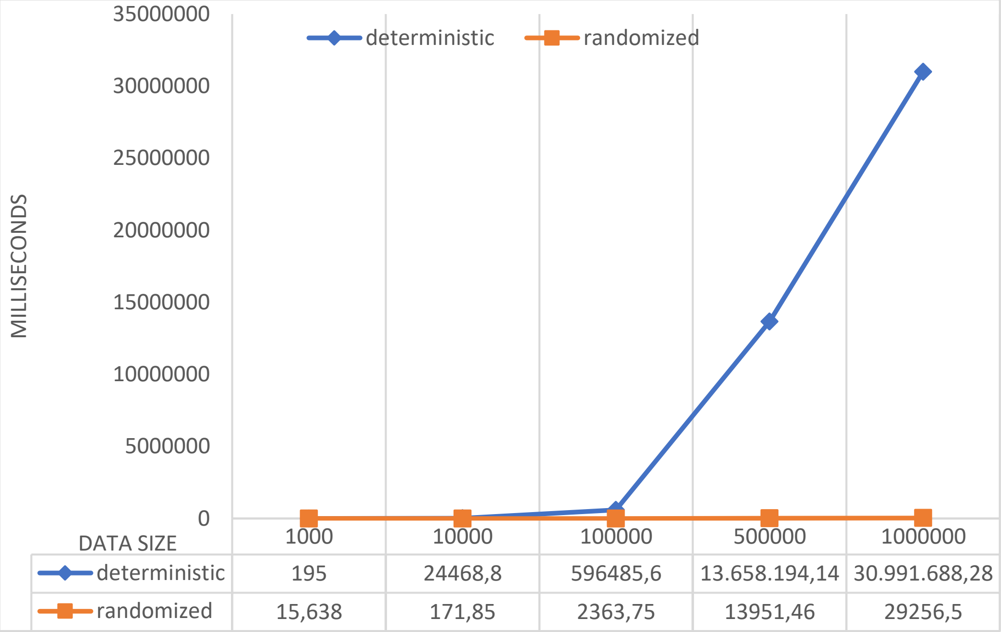
Both can be shown in the below table. Randomized algorithms work a little bit slower than deterministic for average time. However, for some cases, randomized algorithm can work faster than deterministic algorithm.



4)

In the worst case of the Quick Sort algorithm, which is sorted, the results are as follows. In the deterministic algorithm, because all the elements are checked one by one, the run time times are too high which is O (n^2). The randomized algorithm yielded results similar to the unsorted case because it starts by choosing a random pivot each time. In this case, it would be correct to say that the randomized algorithm works faster than the deterministic algorithm.

Sorted Array	Deterministic (Milliseconds)	Randomized (Milliseconds)
1000	195	15,638
10000	24468,8	171,85
100000	596485,6	2363,75
500000	13.658.194,14	13951,46
1000000	30.991.688,28	29256,5



5)

Dual quicksort algorithm is little bit faster the single pivot quicksort algorithms, however, worst case is same with single pivot quicksort algorithm which is O (n^2). Worst case shows up when the array is descending or ascending. The best-case running time of dual quicksort algorithm is $\theta(n \log_3 n)$ instead of $\theta(n \log_2 n)$ because each partition divides to third part. Additionally, if one pivot is not good (max or min this array or close to the smallest or close to largest data), second pivot can be better than this. So dual pivot quicksort algorithm has $\Omega(nlogn), \theta(nlogn)$ and O (n^2).