Name:- Pratiksingh Rajeshsingh Thakur
Registration No.: 2019bcs133
Roll No.: A 63
Division: A
Subject: Advanced Algorithm

Practical 5

Aim:- Implement algorithm of Selection Problem (Randomised). Refer the algorithm taught in the class. Write compelete theory party including algorithm, proof of time complexity, etc. Indicate input and output of your program. Write conclusion properly about your implementation of the algorithm and its behavior with varying size of inputs.

Theory:-

· Time complexity

We know for each partition $1 \le k \le n$, probability of $(k-1)$ & $(n-m)$ partition $\frac{1}{n}$

case 1: Partition can be 1 & n-1 which is worst case senerio case II :- Partition can be k & n-k for all other k.

This gives us:-

$$T(n) \leq \frac{1}{n}\left[T(\max(1, n-1)) + \sum_{k=1}^{n-1} T(\max(k-1, n-k))\right] + O(n)$$

here $O(n)$ = It is for partition

further:

$$T(n) \leq \frac{1}{n}\left(T(n-1) + 2\sum_{k=\lceil n/2 \rceil}^{n-1} T(k)\right) + O(n)$$

since $\max(k-1, n-k) = \begin{cases} k-1 & \text{if } k > \lceil n/2 \rceil \\ n-k & \text{if } k \leq \lceil n/2 \rceil \end{cases}$

Simplifying this gives us

$$T(n) \leq \frac{2}{n} \sum_{k=\lceil n/2 \rceil}^{n-1} T(k) + O(n)$$

By induction we assume that $T(n) \leq c.n$ where $k < n$ and $T(k) \leq ck$

$$\Rightarrow T(n) \leq \frac{2c}{n} \sum_{k=\lceil n/2 \rceil}^{n-1} k + O(n)$$

$$\leq \frac{2c}{n}\left[\sum_{k=1}^{n-1} k + \sum_{k=1}^{\lceil n/2\rceil -1} k\right] + O(n)$$

$$\leq \frac{2c}{n}\left[\frac{n(n-1)}{2} - \frac{\lceil n/2\rceil\left(\lceil n/2\rceil -1\right)}{2}\right] + O(n)$$

$$\leq \frac{2c}{n}\left[\frac{n^2-n}{2} - \frac{n^2/4 - 3n/2 + 2}{2}\right] + O(n)$$

$$\leq \frac{c}{n}\left[n^2-n - \frac{n^2}{4} + \frac{3n}{2} - 2\right] + O(n)$$

$$\leq \frac{c}{n}\left[n^2-n - \frac{n^2}{4} + \frac{3n}{2} - 2\right] + O(n)$$

$$\leq c\left[n-1 - \frac{n}{4} + \frac{3}{2} - \frac{2}{n}\right] + O(n)$$

$$\leq cn + O(n)$$

$$\Rightarrow T(n) = O(n)$$

Average Time Complexity ~~and~~ worst time complexity is still $O(n^2)$

Steps involved in Selection Problem

Randomised select (A, p, r, i):
    if (p = = r) then return A [p]
        q = Randomised partition (A, p, r)
        j = q - p + 1
        if (i = = j) then return A [q]
        else if (i < j) then return randomised select
        else random select (A, q + i, r, i - j)

Randomised partition (A, p, r):
    i = random (p, r)
    swap (a [i] , a [r]);
    partiation (A, p, r);