
Nome: - Pratiksingh Rajestsligh	Thakur
Registration No.: 2019bcs133	
ROI No.: A 63	
Division: A	
Subject: Advanced Algorithm	
Practical 5	
Ain: - Implement algorithm	of Selection Problem
(Randomiced). Refer the	algorithm taught in
the class. Write compelete	
algorithm proof of time	complexity etc. Indirate
Input and output of	your program. Wrote
conclusion properly about	your implementation.
of the algorithm and i	B behavior with
vorying cize of inputs.	
Ain: - Implement algorithm (Randomised). Refer the the class. Write compelete algorithm, proof of time Input and output of conclusion properly about of the algorithm and i vorying size of inputs.	
·Time Complexity	
1, /	
We know for each part	
of (K-1) le (n-m) pa	stition !
2 (1)	n
case 1: Varition can be	
worst case	seneno
Time Complexity We know for each past of (K-1) le (n-m) pa Case 1: Partition can be worst case	
*******	**************************************

K + O(n)

T(n)

	$\leq 2c \left[\sum_{k=1}^{n-1} k \right] + O(n)$
	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$
	$\leq 2c \left[\frac{n^2 - n^{\frac{3}{2}}}{2} - \frac{n^{\frac{3}{2}} + 2}{2} + \frac{3n}{2} + \frac{3n}{$
	$\frac{6 \cdot 2 \cdot (n^{2} - n - n^{2} + 3n - 2)}{4} + O(n)$
	$\frac{C}{n} = \frac{(n^2 - n^2 + 3n - 2)}{4} + O(n)$
	$\frac{1}{4}\left[\begin{array}{cccccccccccccccccccccccccccccccccccc$
	<pre>< cn + o(n)</pre>
=> Pucago	2 cn + o(n) $t(n) = o(n)$ e Time (ompleaity and word complexity is $t(n)$) **********************************

```
✓ RAM Disk Editing ^
      import random import numpy as np
Q
                def random_Select(arr, 1, r, k):
\{x\}
                     if (k > 0 \text{ and } k \leftarrow r - 1 + 1):
pos = randomPartition(arr, 1, r)
                        if (pos - 1 == k - 1):
    return arr[pos]
if (pos - 1 > k - 1):
    return random_Select(arr, 1, pos - 1, k)
                          return random_Select(arr, pos + 1, r, k - pos + 1 - 1)
                     return 999999999999
                arr[i],arr[r]=arr[r],arr[i]
return i
                def randomPartition(arr, 1, r):
    n = r - 1 + 1
    pivot = int(random.random() * n)
    arr[1 + pivot], arr[r].arr[1, arr[1 + pivot]
    return partition(arr, 1, r)
<>
\equiv
>_
              arr = np.random.randint(1,1000,10)
              n = len(arr)
             n = len(arr)
print ("array generated is ", arr)
print ("Sorted array looks like ", sorted(arr))
k = int(input("Enter the value of k "))
print ("K'th smallest element is", random_Select(arr, 0, n - 1, k))
      array generated is [986 906 828 61 867 639 99 525 282 247]
Sorted array looks like [61, 99, 247, 282, 525, 639, 828, 867, 906, 986]
Enter the value of k 5
              K'th smallest element is 525
```