Analytical Assessment - 05

D. Hemalatha
192311107
CS10669

1) To implement the median of Medians algorithm ensures that you handle the worst-case time complexity efficiently while finding the kth smallest element in an unsorted array.

arr = [12, 3, 5, 7, 19] K = 2

arr = [12, 3, 5, 7, 4, 19, 26] K = 3

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] K = 6

i) arr = [12, 3, 5, 7, 19] = [3, 5, (7), 12, 19]

Median = 7

Arrange the values less than 7 in leftside of '7' and greater than 7 in right side.

[12, 3, 5, 7, 19] = [3, 5, (7), 12, 19]

and smallest element = 5.

ii) arr = [12, 3, 5, 7, 4, 19, 26]; K = 3

Divide into sub arrays.

A_1 = [12, 3, 5, 7, 4] and A_2 = [19, 26]

From A_1 Median is = 5

From A_2 Median is = 19

Median[5, 19] = 5.

partition around 5, Arrange the values less than 5 in leftside of '5' and greater than 5 in right side. [3, 4, 5, 7, 12, 19, 26]

= 5 is the 3rd smallest element.

arr = [1,2,3,4,5,6,7,8,9,10]

Divide arr into subarrays

A_1 = [1,2,3,4,5] A_2 = [6,7,8,9,10]

Median = 3 Median = 8

Median of Medians = [3,8]

∴ partition around 3 Arrays the elements less than 3 are in left side and greater elements in right side.

iii) Given arr: [1,2,3,4,5,6,7,8,9,10] K = 6

Arrange the orders in ascending order, it is already arranged.

Index: 0 1 2 3 4 5 6 7 8 9

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

K: 1 2 3 4 5 6 7 8 9 10

Median: $\frac{\text{last+first}}{2} = \frac{0+9}{2} = 4.5 \approx 5$

Median = 6

As given K=6 the value of [K=6] = 6//

2) To implement a function median of medians (arr, k) that takes an unsorted array arr and an integer k, and returns the kth smallest element in the array.

i) Given arr: [1,2,3,4,5,6,7,8,9,10] K=6

Arrange it in ascending order, but it is already arranged: [1,2,3,4,5,6,7,8,9,10].

Median: $\frac{0+9}{2} = 4.5 \approx 5$

Median = 6

As given k = 6, the value of (k=6) = 6//.

ii) Given

arr = [23, 17, 31, 44, 55, 21, 20, 18, 19, 27]

Arrange in ascending order = [17, 18, 19, 20, 21, 23, 27, 31, 44, 55]

Median = $\frac{9+0}{2}$ = 4.5 ≈ 5

= 4.5 ≈ 4

As given k = 5, the value of (k=5) = 21//

3) Closest pair of points.

Given an array of points where points[i] = [x_i, y_i] represents a point on the x-y plane and an integer k, return the k-closest pair to the origin (0,0).

i) points = [[1,3], [-2,2], [5,8], [0,1]], k = 2

Given

points = [[1,3], [-2,2], [5,8], [0,1]]

Distance = $x^2 + y^2$

(1,3) = $1^2 + 3^2 = 10$

[-2,2] = $(-2)^2 + 2^2$

= 8

[-5,8] = $5^2 + 8^2$

= 25 + 64

= 89

[0,1] = $0^2 + 1^2$

= 1

Distance = [10, 8, 89, 1]

Arrange the points : [[0,1], [-2,2], [1,3], [5,8]].

k = 2.

: [[0,1], [-2,2]]

ii) points : [[1,3], [-2,2]], k = 1.

Given:

Distance = x^2+y^2

[1,3] : 1^2+3^2 [-2,2] : $(-2)^2+2^2$

= 10. = 8//

Distance = [10, 8]

Arrange the points : [[-2,2], [1,3]].

k = 1

[-2,2]

iii) points : [[3,3], [5,-1], [-2,4]] k = 2

Distance : x^2+y^2

[3,3] : 3^2+3^2 [5,-1] : $5^2+(-1)^2$

= 18 = 26

[-2,4] : $[-2]^2+[4]^2$

= 20

Distance = [18, 26, 20]

[3,3], [-2,4].

4) Given four DFSs A, B, C, D of integers value, write a program to compute how many types (i, j, k, l). There are such that A[i] + B[j] + C[k] + D[l] is

Zero.

A=[1,2] B=[-2,-1] C=[-1,2] D=[0,2]

```
for a in A:
    for b in B:
AB_sum_counts[a+b]+=1
count=0
  for c in C:
  for d in D:
    complement=-(c+d)
    if complement in AB_sum_counts
        count+=AB_sum_counts[complement]
  return count
A=[1,2]
B=[-2,-1]
C=[-1,2]
D=[0,2]
Print(four_sum_count(A,B,C,D))
```

Hi) A=[0], B=[0], C=[0], D=[0]

```
from collections import defaultdict
def four_sum_count(A,B,C,D):
  AB_sum_counts=defaultdict(int)
  for a in A:
    for b in B:
```

```
        AB_sum_counts[a+b] += 1
    count = 0
    for c in C:
        for d in D:
            complement = -(c+d)
        if complement in AB_sum_counts:
        count += AB_sum_counts[complement].
return count
A = [0]
B = [0]
C = [0]
D = [0]
print(f"{for_sum_count([A, B, C, D])}")
```