Date :- 25/06/24

B. Laranya

CSA0672 - DAA

192311131

① To implement the median of median algorithm ensures that you handle the worst-case time complexity efficiently while finding the k^{th} smallest element in an unsorted array.

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

Given

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

Index = 0 1 2 3 4

Arrange the array in ascending order = [3, 5, 7, 12, 19]

K = 1 2 3 4 5

$$\text{Median} = \frac{\text{low+high}}{2} = \frac{0+4}{2} = 2$$

Median = 7

As given K = 2, the value of (K=2) = 5

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

Given

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

Index = 0 1 2 3 4 5 6

Arrange the array in ascending order = [3, 4, 5, 7, 12, 19, 26]

K = 1 2 3 4 5 6 7

$$\text{Median} = \frac{\text{low+high}}{2} = \frac{0+6}{2} = 3$$

Median = 7

As given, K = 2

The value of (K = 3) = 5

(iii) Given

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

Arrange the order 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 = $\frac{low+high}{2} = \frac{0+9}{2} = 4.5 \approx 5$

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 ≈ 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 = $\frac{0+9}{2}$ = 4.5 ≈ 5

≈ 4.5 ≈ 4

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

(3) Closest pair of Paths

Given an array of points where points[i] = [x_i, y_i] represents a point an 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 in that order close to the origin by ~~ascending~~ dis. considering distances

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

As the value K=2,

Consider first 2 points, so the closet pair

= [[0,1], [-2,2]]

(ii) Points = [[1,3], [-2,2]], K=1

Given

Distance = x^2+y^2

$[1,3] = 1^2+3^2$

$= 10$ //

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

$= 8$ //

Distance = [10, 8]

Arrange the points in such a order that are close to the origin by considering distances = [[-2,2], [1,3]]

As the value, K=1

Consider, first points, so the closest pair = [-2, 2]

(iii) Points = [[3,3], [5,-1], [-2,4]] K = 2

Distance = x^2+y^2

[3,3] = 3^2+3^2 = 18

[5,-1] = $5^2+(-1)^2$ = 26

[-2,4] = $[-2]^2+[4]^2$ = 20

Distance = [18, 26, 20]

Arrange of points should be done in such a way that are close to origin considering distances.

As the value K = 2, Take two points into consideration, [3,3], [-2,4].

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

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

```
From collections import default dict
def fourlists (A, B, C, D):
AB_fourlists = default dict(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 = [1, 2]
B = [-2, -1]
C = [-1, 2]
D = [0, 2]
Print (four_Sum_count(A,B,C,D))
```

(ii) 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 ( for_Sum_Count(A,B,C,D)) .9
```