P. Gunasri

Analytical Assignment - 5 CSA0669 192324103

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

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

Given

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

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

Index = 0 1 2 3 4

k = 1 2 3 4 5

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

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

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

Index = 0 1 2 3 4 5 6

k = 1 2 3 4 5 6 7

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

median = 7

As given, k = 2,

the value of (k = 3) = 5 //

(iii) arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] k = 6

Sol: 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 = $\frac{\text{low + high}}{2} = \frac{0+9}{2} = 4.5 \approx 5$

median = 6

As, given k = 6, The value of (k = 6) = 6

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

(i) arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] k = 6

Sol:- 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 //

closest pair of points:-

1) Given an array of points where points[i]=[xi, yi] represents a p

(i) arr= [23, 17, 31, 44, 55, 21, 20, 18, 19, 27] k=5

Given,

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

Arrange the order in ascending order

index 0 1 2 3 4 5 6 7 8 9

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

k= 1 2 3 4 5 6 7 8 9 10

median = $\frac{\text{low + high}}{2} = \frac{0+9}{2} = 4.5 \approx 5$ ✓

$4.5 \approx 4$

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

closest pair of points

1) Given an array of points where points[i]=[xi, yi] 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 = $\sqrt{x^2+y^2}$

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

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

= 8 //

$[5,8] = 5^2+8^2$ $\quad [0,1] = 0^2+1^2$

$= 25+64$ $\quad = 1$

$= 89$

Distance = $[10, 8, 89, 1]$

Arrange the points in that order close to the origin by 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,

points = $[[1,3], [-2,2]]$

Distance = x^2+y^2

$[1,3] = 1^2+3^2$ $\quad [-2,2] = (-2)^2+2^2$

$= 10$ // $\quad = 4+4 = 8$ //

Distance = $[10, 8]$

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

As the value, $k=1$

Consider first points, so the closet pair

$= [-2,2]$

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

Given

points = [[3,3], [5,-1], [-2,4]]

Distance = x^2+y^2

[3,3] = 3^2+3^2
= 9+9
= 18

[5,-1] = $5^2+(-1)^2$
= 25+1
= 26

[-2,4] = $[-2]^2+[4]^2$
= 4+16
= 20

Distance = [18, 26, 20]

As the arrangement 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]

2) Given four lists A, B, C, D of integer values, 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 defaultdict
def fourlists(A, B, C, D):
    AB = fourlists ~~count~~ = 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 = [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(four_sum_count(A, B, C, D))
```