DAA432C
ASSIGNMENT-01

Presented by GROUP 7
JAIDEV DAS (IIT2019197)
NISCHAY NAGAR (IIT2019198)
PRIYANSHU (IIT2019196)

CONTENTS

PROBLEM STATEMENT

Given an array representing n positions along a straight line. Find k (where k <= n) elements from the array such that the minimum distance between any two (consecutive points among the k points) is maximized.

For example:

Input : arr[ ] = {1, 2, 8, 4, 9}

k = 3

Output :

Largest minimum distance = 3

3 elements arranged at positions 1, 4 and 8

ALGORITHM

1. First sort given array in ascending order
2. Take minDist = -1
3. Take left = 0 and right = maximum distance between given elements + 1
4. Find midpoint of (left + right) (let's call it mid)
5. Check if k elements can be placed with minimum distance between any two consecutive elements equal to mid. If this is true, update left to mid + 1 and minDist to mid, store the k elements, and repeat from step 4. Else, update right to mid and repeat from step 4.
6. Keep repeating above steps while left is less than right
7. When above condition fails, return minDist and the stored k elements

PSEUDO CODE

```
maxMinDist(arr,k):
    sort(arr)
    minDist = -1
    points = []
    left = 0
    right = arr[last] + 1
    while left < right:
        mid = (l + r) / 2
        r = findPoints(arr,k,mid)
        if (r):
           minDist = mid
           points = r
           left = mid + 1
        else:
           right = mid

    print(minDist)
    print(points)
```

```
findPoints(arr,k,mid):
      curr = arr[first]
      count = 1
      points = [[arr[first]]
      for el in arr[2nd to last]:
            if el - curr >= mid:
                  curr = el
                  points.append(curr)
                  count = count + 1
                  if count = k:
                        return points

      return False
```

TIME COMPLEXITY

Here, arr is the array of given elements, n is number of elements in arr

In the while loop we are halving the maximum distance between 2 points in given array (let's call it d) in each iteration. (d = arr[last element] - arr[first element] if arr is sorted in ascending order)

So the while loop runs log(d) times.

Inside this loop, there's a call to another function which employs a loop which runs for maximum n-1 times. So this function runs for n-1 times in worst case.

So T(n) = (n-1)log(d) = O(nlgd)

Where d is the maximum distance between given elements.

RESULTS

```
Enter number of elements in array (n):
5
Enter value of k (k <= n):
3

Given array is:
[2, 39, 88, 47, 53]
Maximum minimum distance between 3 points is 41
The placements of k points is given below:
[2, 47, 88]
```

```
Enter number of elements in array (n):
10
Enter value of k (k <= n):
5

Given array is:
[18, 28, 92, 35, 61, 73, 36, 89, 24, 76]
Maximum minimum distance between 5 points is 15
The placements of k points is given below:
[18, 35, 61, 76, 92]
```

THANK YOU

...