

MINIMISING MAXIMUM DISTANCE B/W GAS STATIONS

* In this problem, we are given an array of sorted integers where each integer represents the coordinate of a gas station. Our job is to place K more gas stations such that the maximum distance between any 2 gas stations which are adjacent is minimized. This is to be done while maintaining a sorted order.

This minimum distance is to be returned as a long double.

Clearly, placing the new gas stations at the end or before the start doesn't make sense. That is because the distance between the already existing elements would still be the same. Hence any new addition has to be within already existing. Our first priority is to always place the gas stations at equal distances. We find where the maximum distance is at, then half it by placing a gas station between them. Repeat this process until all gas stations are placed.

$$\text{Eg : } [1 \underbrace{13}_{12} \underbrace{17}_{4} \underbrace{23}_{6}] \quad K = 5$$

12 is max, add a gas station in
b/w, K--

$$[1 \underbrace{7}_{6} \underbrace{13}_{6} \underbrace{17}_{4} \underbrace{23}_{6}]$$

6 is max, add a gas station in
b/w, K--

$$[1 \underbrace{7}_{6} \underbrace{13}_{6} \underbrace{17}_{4} \underbrace{20}_{3} \underbrace{23}_{3}]$$

Repeat this process

Pseudocode :

```

maxGasStations(arr, N, K) {
    int howMany[K-1]
    for(g = 1 → K) {
        maxVal, maxIndex = -1, -1
        for(i = 0 → N-1) {
            dist = arr[i+1] - arr[i]
            sectorLength = dist[howMany[i]+1]
            if(maxVal < sectorLength) {
                maxVal = sectorLength[i]
                maxIndex = i
            }
        }
        howMany[maxIndex] ++
    }
    maxi = -1
}

```

```
for(i = 0 → N - 1) {  
    sectorLength = (arr[i + 1] - arr[i]) /  
                    (howMany[i] + 1)  
    maxi = max(maxi, sectorLength)  
}  
return maxi  
}
```

This is brute force solution. Time Complexity is $O(NK + N)$