Angry Children 2

by amititkgp

Problem Submissions Leaderboard Discussions Editorial

Bill Gates is on one of his philanthropic journeys to a village in Utopia. He has **N** packets of candies and would like to distribute one packet to each of the **K** children in the village (each packet may contain different number of candies). To avoid a fight between the children, he would like to pick **K** out of **N** packets such that the unfairness is minimized.

Suppose the **K** packets have (x_1, x_2, x_3,....x_k) candies in them, where x_i denotes the number of candies in the i^{th} packet, then we define *unfairness* as

$$\sum_{1 \le i < j \le k} |X_i - X_j|$$

where |a| denotes the absolute value of a.

Input Format
The first line contains an integer N.
The second line contains an integer K.
N lines follow each integer containing the candy in the i^{th} packet.

Output Format
A single integer which will be minimum unfairness.

Constraints
2<=N<=10^5
2<=K<=N
0<= number of candies in each packet <=10^9

Sample Input #00

```
7
3
10
100
300
200
1000
20
30
```

Sample Output #00

```
40
```

Explanation #00

Bill Gates will choose packets having 10, 20 and 30 candies.So unfairness will be |10-20| + |20-30| + |10-30| = 40. We can verify that it will be minimum in this way.

Sample Input #01

```
10
4
1
2
3
4
10
20
30
40
100
200
```

Sample Output #01

```
10
```

Explanation #01

Bill Gates will choose 4 packets having 1,2,3 and 4 candies. So, unfairness will be |1-2| + |1-3| + |1-4| + |2-3| + |2-4| + |3-4| = 10

Submissions: 2564
Max Score: 50
Difficulty: Hard

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 8

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) throws IOException{

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        int N = Integer.parseInt(br.readLine());
        int K = Integer.parseInt(br.readLine());

        long[] arr = new long[N];
        long[] sum = new long[N];


        arr[0] = Long.parseLong(br.readLine());

        for(int i = 1 ; i < N ; i++){
            arr[i] = Long.parseLong(br.readLine());

        }

        Arrays.sort(arr);
        sum[0] = arr[0];

        for(int j = 1 ; j < N ; j++){
            sum[j] = arr[j] + sum[j - 1];
        }

        long currentSum = 0;

        for(int j = 0 ; j < K - 1 ; j++){
            currentSum += (sum[K - 1] - sum[j] - ((K - 1 - j) * arr[j]));
        }

```

```
37
38          long temp = currentSum;
39
40          for(int j = 1 ; j <= N - K ; j++){
41              temp = temp + ( (K - 1) * (arr[j - 1] + arr[j + K - 1])) - (2 * (sum [j + K - 1 - 1] - sum[j - 1]) ) ;
42
43              if(temp < currentSum){
44                  currentSum = temp;
45              }
46          }
47
48          System.out.println(currentSum);
49
50      }
51  }
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code