Fair Cut

by tunyash

Problem Submissions Leaderboard Discussions Editorial

Li and Lu have n integers, $a_1, a_2, \ldots, a_n$, that they want to divide fairly between the two of them. They decide that if Li gets integers with indices $I = \{i_1, i_2, \ldots, i_k\}$ (which implies that Lu gets integers with indices $J = \{1, \ldots, n\} \setminus I$), then the measure of unfairness of this division is:

$$f(I) = \sum_{i \in I} \sum_{j \in J} |a_i - a_j|$$

Find the minimum measure of unfairness that can be obtained with some division of the set of integers where Li gets exactly k integers.

Note $A \setminus B$ means Set complement

Input Format

The first line contains two space-separated integers denoting the respective values of n (the number of integers Li and Lu have) and k (the number of integers Li wants).
The second line contains n space-separated integers describing the respective values of $a_1, a_2, \ldots, a_n$.

Constraints

- $1 \leq k < n \leq 3000$
- $1 \leq a_i \leq 10^9$
- For 15% of the test cases, $n \leq 20$.
- For 45% of the test cases, $n \leq 40$.

Output Format

Print a single integer denoting the minimum measure of unfairness of some division where Li gets k integers.

Sample Input 0

```
4 2
4 3 1 2
```

Sample Output 0

```
6
```

Explanation 0
One possible solution for this input is $I = \{2, 4\}$; $J = \{1, 3\}$. $|a_2 - a_1| + |a_2 - a_3| + |a_4 - a_1| + |a_4 - a_3| = 1 + 2 + 2 + 1 = 6$

Sample Input 1

```
4 1
3 3 3 1
```

Sample Output 1

```
2
```

Explanation 1

The following division of numbers is optimal for this input: $I = \{1\}; J = \{2, 3, 4\}$.

Submissions: 910
Max Score: 40
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 8

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {
    public static void main(String args[] ) throws Exception {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String[] num = br.readLine().split("\\s");

        int n = Integer.parseInt(num[0]);
        int k = Integer.parseInt(num[1]);

        num = br.readLine().split("\\s");

        long[] arr = new long[n];
        long sum = 0;

        for(int i = 0 ; i < n ; i++){
            arr[i] = Long.parseLong(num[i]);
            sum += arr[i];
        }

        Arrays.sort(arr);
        ArrayList<Integer> listA = new ArrayList<Integer>();
        ArrayList<Integer> listB = new ArrayList<Integer>();

        if(k > n / 2){
            k = n - k;
        }

        if(n % 2 == 0){

            int index;
            long value = sum / n;

            index = (n/2);


            if(k % 2 == 0)
            {
                listB.add(index);
                int a = 1;

                while(true)
                {
                     if(listB.size() == k )
                     {
```

```
                    break;
                }

                if(listB.size() == k -1){

                    int intA,intB;

                    if((index + (2*a)) < n){
                        intA = index + (2*a);
                    }
                    else{
                        intA = Integer.MAX_VALUE;
                    }

                    if((index - (2*a)) >= 0){
                        intB = index - (2*a);
                    }
                    else{
                        intB = Integer.MAX_VALUE;
                    }

                    if(intB < intA){
                        listB.add(index - (2*a));
                    }
                    else{
                        listB.add(index + (2*a));
                    }

                    break;

                }

                if((index + (2*a)) < n){
                    listB.add(index + (2*a));
                }

                if((index - (2*a)) >= 0){
                    listB.add(index - (2*a));
                }

                a++;
            }

        }
        else
        {

            int a = 1;
            listB.add(index);

            while(true)
            {
                 if(listB.size() == k)
                 {
                     break;
                 }

                 if((index + (2*a)) < n){
                     listB.add(index + (2*a));
                 }

                 if((index - (2*a)) >= 0){
                     listB.add(index - (2*a));
                 }

                 a++;

            }
        }
    }
    else{

        if(k % 2 == 0)
```

```
            {
                int index = n/2;
                int a = 0;
                while(true)
                {
                    if(listB.size() == k)
                    {
                        break;
                    }

                    if((index + (2*a + 1)) < n){
                        listB.add(index + (2*a + 1));
                    }

                    if((index - (2*a + 1)) >= 0){
                        listB.add(index - (2*a + 1));
                    }

                    a++;
                }
            }
            else
            {
                int index = n/2 ;
                listB.add(index);
                int a = 1;
                while(true)
                {
                    if(listB.size() == k)
                    {
                        break;
                    }

                    if((index + (2*a)) < n){
                        listB.add(index + (2*a));
                    }

                    if((index - (2*a)) >= 0){
                        listB.add(index - (2*a));
                    }

                    a++;
                }
            }
        }

        for(int i = 0 ; i < n ; i++){

            if(!listB.contains(i)){
                listA.add(i);
            }
        }


        long output = 0;

        for(int i = 0 ; i < listA.size() ; i++){

            for(int j = 0 ; j < listB.size() ; j++){
                output += (Math.abs(arr[listA.get(i)] - arr[listB.get(j)]));
            }
        }

        System.out.println(output);

    }
}

```

Line: 1 Col: 1

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