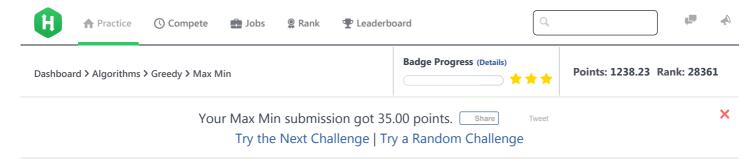
3/21/2018 HackerRank







Problem Submissions Leaderboard Discussions Editorial 🔒 **Topics**

You will be given a list of integers, arr, and a single integer k. You must create an array of length k from elements of arr such that its unfairness is minimized. Call that array *subarr*. Unfairness of an array is calculated as

max(subarr) - min(subarr)

Where:

- max denotes the largest integer in subarr.
- min denotes the smallest integer in subarr.

As an example, consider the array [1,4,7,2] with a k of 2. Pick any two elements, test subarr = [4,7]. unfairness = max(4,7) - min(4,7) = 7 - 4 = 3

Testing for all pairs, the solution [1, 2] provides the minimum unfairness.

Note: Integers in arr may not be unique.

Input Format

The first line contains an integer n, the number of elements in array arr.

The second line contains an integer k.

Each of the next n lines contains an integer arr_i where $0 \leq i < n$.

Constraints

 $2 \le n \le 10^5$

 $2 \le k \le n$

 $0 \leq arr_i \leq 10^9$

Output Format

An integer that denotes the minimum possible value of unfairness.

Sample Input 0

7

3 10

100

300

200

1000

30

Sample Output 0

20

Explanation 0

3/21/2018 HackerRank

Here $\emph{\textbf{K}}=\emph{\textbf{3}}$; selecting the $\emph{\textbf{3}}$ integers $\emph{\textbf{10}},\emph{\textbf{20}},\emph{\textbf{30}}$, unfairness equals

```
\max(10,20,30) - \min(10,20,30) = 30 - 10 = 20
```

Sample Input 1

10

1

2 3

ر 4

20

30

40

100

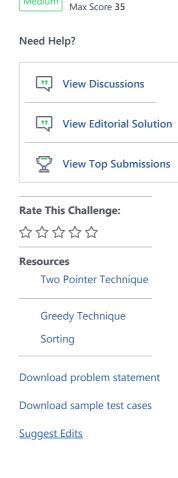
Sample Output 1

3

Explanation 1

Here K=4; selecting the **4** integers **1, 2, 3, 4**, unfairness equals

$$\max(1,2,3,4) - \min(1,2,3,4) = 4 - 1 = 3$$



f ⊌ in

Submitted 41203 times

Medium

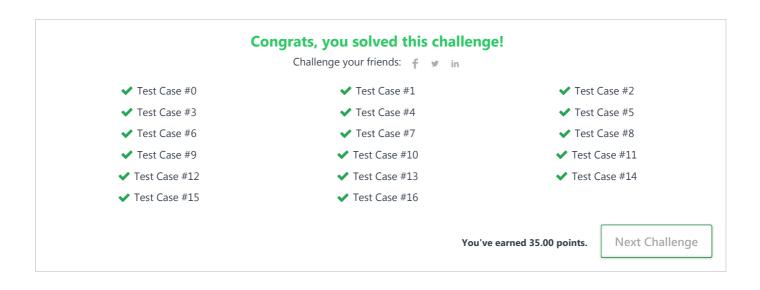
3/21/2018 HackerRank

```
1 ▼ /** Terminated due to timeout */
 2 ▼ #include <iterator>
 3 #include <vector>
 4 #include <iostream>
 5 #include <algorithm>
   #include <limits>
 7 #define speed std::ios_base::sync_with_stdio(false); std::cin.tie(nullptr); std::cout.tie(nullptr)
 8 typedef unsigned long long int uint64;
 9
10 int main()
11 ▼ {
        speed;
12
13
        uint64 N; std::cin>>N;
14
        uint64 k; std::cin>>k;
15
        std::vector<uint64> vec;
16
        vec.reserve(N);
17
        std::copy_n(std::istream_iterator<uint64>(std::cin), N, back_inserter(vec));
18
        std::sort(vec.begin(), vec.end());
19
20
        uint64 answer = std::numeric_limits<uint64>::max();
        for(size_t i=0; i<=N-k; ++i)</pre>
21
22 ▼
        {
            auto MAX = vec[i+k-1];
23 ▼
24 ▼
            auto MIN = vec[i];
            answer = std::min( answer, (MAX - MIN) );
25
            //std::cout<<answer<<" "<<MIN<<" "<<MAX<<" "<<i<<"\n";
26
27
28
        std::cout<<answer<<std::endl;</pre>
29
        return 0;
   }
30
31
                                                                                                             Line: 31 Col: 1
```

<u>Upload Code as File</u> Test against custom input

Run Code

Submit Code



 $Contest\ Calendar [Blog|Scoring] Environment [FAQ|About\ Us|Support|Careers|Terms\ Of\ Service|Privacy\ Policy|Request\ a\ Feature$