Minimum Loss

locked

by nabila_ahmed

Lauren has a chart of projected prices for a house over the next n years, where the price of the house in the i^{th} year is p_i. She wants to purchase and resell the house at a minimal *loss* according to the following rules:

- The house cannot be sold at a price greater than or equal to the price it was purchased at (i.e., it must be resold at a loss).
- The house cannot be resold within the same year it was purchased.

Find and print the *minimum* amount of money Lauren must lose if she buys the house and resells it within the next n years.

Note: It's guaranteed that a valid answer exists.

Input Format

The first line contains an integer, n, denoting the number of years of house data.
The second line contains n space-separated long integers describing the respective values of $p_1, p_2, \ldots, p_n$.

Constraints

- $2 \leq n \leq 2 \times 10^5$
- $1 \leq p_i \leq 10^{16}$
- It's guaranteed that a valid answer exists.

Subtasks

- $2 \leq n \leq 1000$ for 50% of the maximum score.

Output Format

Print a single integer denoting the minimum amount of money Lauren must lose if she buys and resells the house within the next n years.

Sample Input 0

```
3
5 10 3
```

Sample Output 0

```
2
```

Explanation 0

Lauren buys the house in year 1 at price $p_1 = 5$ and sells it in year 3 at $p_3 = 3$ for a minimal loss of $5 - 3 = 2$.

Sample Input 1

```
5
20 7 8 2 5
```

Sample Output 1

```
2
```

Explanation 1

Lauren buys the house in year 2 at price $p_2 = 7$ and sells it in year 5 at $p_5 = 5$ for a minimal loss of $7 - 5 = 2$.

Submissions: 1451
Max Score: 35
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
class Solution {

        static void quicksort(long[] vector, int[] paralelo, int primero, int ultimo)
        {
            int i, j, central;
            long pivote;
            central = (primero + ultimo) / 2;
            pivote = vector[central];
            i = primero;
            j = ultimo;
            do
            {
                while (vector[i] < pivote) i++;
                while (vector[j] > pivote) j--;
                if (i <= j)
                {
                    long temp;
                    temp = vector[i];
                    vector[i] = vector[j];
                    vector[j] = temp;

                    int t2 = paralelo[i];
                    paralelo[i] = paralelo[j];
                    paralelo[j] = t2;

                    i++;
                    j--;
                }
            } while (i <= j);

            if (primero < j)
            {
                quicksort(vector, paralelo, primero, j);
            }
            if (i < ultimo)
            {
                quicksort(vector, paralelo, i, ultimo);
            }
        }


        static void Main(string[] args)
        {
            int n =  int.Parse(Console.ReadLine());
            long[] p =  Array.ConvertAll(Console.ReadLine().Split(' '), e => long.Parse(e));


            //Dictionary<long, long> diccio = new Dictionary<long, long>();

            int[] indices = new int[n];
            for (int i = 0; i < n; i++)
            {
                indices[i] = i;
            }

            quicksort(p, indices, 0, n - 1);


            long min_dif = int.MaxValue;
            for (int i = 1; i < n; i++)
            {
                long dif = p[i] - p[i - 1];
```

```
67                 if (indices[i - 1] > indices[i])
68                 {
69                     min_dif = Math.Min(min_dif, dif);
70                 }
71             }
72
73             Console.WriteLine(min_dif);
74
75             //Console.ReadLine();
76
77         }
78 }
```

Line: 65 Col: 27

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

Testcase 0 ✔ Testcase 1 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run you code against all the test cases.

Input (stdin)

```
3
5 10 3
```

Your Output (stdout)

```
2
```

Expected Output

```
2
```