using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

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));

/\* la idea es ordenar el array, y asociarle un array paralelo

\* para almacenar los indices, hago un array paralelo porque,

\* si lo ordeno y luego lo recorro de punta a punta

\* para saber cual es la minima diferencia entre p[i] - p[i-1]

\* entonces el indice p[i-1] puede estar despues o antes en el

array no ordenado, y yo necesito que p[i] sea menor que p[i-1]

y ADEMAS que el indice i-1 sea mayor que i en el array original,

por eso hago un array paralelo almacenando los indices y lo ordeno

como se ordenan los arrays con los paralelos usando el quicksort

porque sino da que excede el tiempo limite.

\* Si hago un bucle i con un bucle j interno tambien excede el tiempo limite

en ese caso tomaria O(n^2),

pero haciendolo así toma O(nLog n) para el quicksort

y O(n) para verificar de punta a punta la minima diferencia

\* entre p[i-1] y p[i]\*/

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];

if (indices[i - 1] > indices[i])

{

min\_dif = Math.Min(min\_dif, dif);

}

}

Console.WriteLine(min\_dif);

//Console.ReadLine();

}

//static void Main(string[] args)

//{

// long[] arr = { 34,2344,long.MaxValue ,1,8,85};

// int[] indices = { 0, 1, 2, 3,4, 5 };

// quicksort(arr, indices, 0, arr.Length-1);

// for (int i = 0; i < arr.Length; i++)

// {

// Console.WriteLine(arr[i] + " " + indices[i]);

// }

// Console.ReadLine();

//}

}

}