Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática

Grupo (C o F): _____

Nombres: _____

Laboratorio: _____ **Puesto:** _____ **Usuario de Exacrc:** _____

Normas de realización de la misión

1. Debes programar soluciones para cada uno de los ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección `http://exacrc/domjudge/team`.
2. Escribe comentarios que expliquen tu solución, justifiquen por qué se ha hecho así y ayuden a entenderla. Calcula la complejidad de todas las funciones que implementes.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
4. Escribid vuestro **nombre y apellidos** en un comentario en la primera línea de cada fichero que subais al juez.
5. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.

1. Supongamos dado un vector ordenado de $n \geq 1$ elementos, en el que todos los elementos aparecen repetidos dos veces, excepto uno que solamente aparece una vez (por tanto n es impar). Se pide diseñar un algoritmo recursivo eficiente que devuelva la posición de dicho elemento. Plantea la recurrencia correspondiente al coste de este algoritmo e indica a qué orden de complejidad pertenece la solución a la misma.

Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación. Cada caso de prueba se compone de dos líneas. La primera de ellas contiene el número de elementos del vector. La segunda contiene los elementos del vector separados por blancos.

Salida

Para cada caso de prueba se escribirá en una línea la posición del elemento que aparece una única vez.

Entrada de ejemplo

```
9
1
3
3
1 2 2
5
1 1 2 9 9
5
1 2 2 9 9
5
1 1 2 2 9
7
3 5 5 9 9 11 11
7
3 3 5 9 9 11 11
7
3 3 5 5 9 11 11
7
3 3 5 5 9 9 11
```

Salida de ejemplo

```
0
0
2
0
4
0
2
4
6
```

2. El dueño del casino está muy preocupado por que los jugadores de ruleta puedan llegar a detectar que el crupier controla el número que saca en cada tirada. Todas las noches estudia la lista de números que han salido ese día y comprueba que sean suficientemente dispersos. El dueño se conforma con que la diferencia entre el primer valor que se saca y el último sea mayor que una cantidad K. Además comprueba que los datos que salieron en la primera mitad de la noche y los datos que salieron en la segunda mitad sean también suficientemente dispersos, esto es que la diferencia entre el primer valor y el último tanto de la primera mitad como de la segunda sea mayor que K y además cada una de sus mitades sea también suficientemente dispersa. Estudia únicamente secuencias con un número de elementos que sea potencia de dos para poder dividirlas siempre en dos partes iguales. Considera que un solo valor siempre es suficientemente disperso. Plantea la recurrencia correspondiente al coste de este algoritmo e indica a qué orden de complejidad pertenece la solución a la misma.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de dos líneas. En la primera se muestra el número de tiradas que se considera y el valor de dispersión que se precisa. En la segunda se muestran los números que han salido en cada tirada El número de valores de cada caso de prueba es una potencia de 2. El valor de dispersión es un entero positivo mayor que cero. Los valores de cada tirada son números entre $0 \leq N \leq 1.048.576$.

Salida

Para cada caso de prueba se escribe en una línea SI si el vector está suficientemente disperso y NO si no lo está.

Entrada de ejemplo

```
4 3
6 1 3 9
4 3
3 10 12 14
8 4
20 2 0 4 14 8 5 10
```

Salida de ejemplo

```
SI
NO
SI
```