

Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática

Examen Primer Cuatrimestre, 9 de Febrero de 2017.

Nombre: _____

Laboratorio: _____ Puesto: _____ Usuario de DOMjudge: _____

Para la realización de los ejercicios que siguen utilizaremos las siguientes definiciones:

- Se dice que un vector es *creciente por los pelos* cuando, además de ser creciente, la diferencia entre un elemento y el elemento siguiente es como mucho uno.

Por ejemplo, los siguientes vectores de tamaño 4 cumplen la definición:

1	2	3	4
---	---	---	---

1	2	2	3
---	---	---	---

1	1	1	1
---	---	---	---

Fijate que al no exigirse que el vector sea *estrictamente creciente*, el último vector con todos 1's cumple la definición.

Por su parte el vector

1	2	1	2
---	---	---	---

no es creciente por los pelos, pues no es creciente.

- Los vectores aburridos son aquellos en los que hay elementos que aparecen repetidos muchas veces. Se dice que un vector es *d-divertido* cuando ningún elemento se repite más de *d* veces.

Ejemplos de vectores *1-divertido* son

1	2	3	4
---	---	---	---

4	3	7	0
---	---	---	---

pues ningún elemento aparece más de una vez (fíjate que, en realidad, los vectores anteriores son también *10-divertidos*, pues ningún elemento aparece más de 10 veces).

Otros ejemplos de vectores y su grado de diversión más pequeño:

6	7	6	6
---	---	---	---

3-divertido

7	3	7	0
---	---	---	---

2-divertido

7	3	7	3
---	---	---	---

2-divertido

9	9	9	9
---	---	---	---

4-divertido

Las dos definiciones pueden combinarse. Por ejemplo, el vector

1	1	2	2	2	3	4	5
---	---	---	---	---	---	---	---

es *creciente por los pelos* y *3-divertido*

1. (4 puntos) Especifica, diseña e implementa una función que reciba un vector de enteros de longitud $n \geq 0$ y un parámetro $d > 0$ y devuelva si el vector es *creciente por los pelos* y *d-divertido*. Escribe el invariante y función de cota que permitan demostrar la corrección del algoritmo implementado. Por último, justifica el coste del algoritmo conseguido.

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá los valores de d , n y los elementos del vector. El programa escribirá SI si el vector es *creciente por los pelos* y *d-divertido* y NO en caso contrario.

Entrada						Salida
d	n	v				
1	4	1	2	3	4	SI
1	4	1	2	1	2	NO
1	4	1	2	3	3	NO
1	4	4	3	2	1	NO
2	4	1	2	3	4	SI
2	4	1	2	3	3	SI
2	4	1	1	2	2	SI
2	4	1	1	3	3	NO
2	4	1	1	1	2	NO
2	4	1	1	1	3	NO
5	4	1	1	1	1	SI
5	4	1	1	3	4	NO

2. (3 puntos) Diseña e implementa un algoritmo recursivo que reciba un vector de longitud $n \geq 0$ que *se sabe creciente* (no necesariamente estrictamente creciente) y devuelva si el vector es creciente por los pelos. Justifica el coste del algoritmo implementado.

Se valorarán soluciones que utilicen el esquema *divide y vencerás* y que no recorran el vector completo si no es necesario.

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá el valor de n y los elementos del vector. El programa escribirá SI si el vector es *creciente por los pelos* y NO en caso contrario.

Entrada					Salida
n	v				
4	1	1	1	1	SI
4	1	1	1	2	SI
4	1	1	1	9	NO
4	1	1	2	2	SI
4	1	2	3	4	SI
4	1	2	3	3	SI
4	1	1	3	3	NO
4	1	2	4	5	NO

3. (3 puntos) Escribe una función para generar vectores crecientes por los pelos *d-divertidos*. En concreto, la función recibirá el tamaño del vector, el valor de d y el valor para el primer elemento del vector y generará la salida en orden lexicográfico. Se admite el uso de funciones auxiliares.

La entrada comienza con una línea que contiene el número de casos de prueba. La entrada contendrá los valores de n , d , y e . El programa escribirá en líneas separadas cada uno de los vectores del resultado, con los elementos separados por un espacio.

Entrada			Salida
n	d	e	
3	3	1	1 1 1
			1 1 2
			1 2 2
			1 2 3
3	2	1	1 1 2
			1 2 2
			1 2 3

```
// Escribe todos los vectores crecientes por los pelos de tamaño n
// que sean además d-divertidos y en los que el primer elemento sea e
void escribeCrecientesPorLosPelosDivertidos(int n, int d, int e);
```