

Fundamentos de Algoritmia

Grados en Ingeniería Informática. Grupos A, DG, C, F

Examen Convocatoria Extraordinaria, 21 de junio de 2024.

Nombre: _____ Grupo: _____

Laboratorio: _____ Puesto: _____ Usuario de DOMjudge: _____

Normas de realización del examen

1. Debes programar soluciones para cada uno de los tres 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. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas 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. (3.5 puntos) Dado un vector de números enteros, se quiere calcular el número de intervalos de un cierto tamaño dado k que no tengan ningún subsegmento (posiciones consecutivas del vector) con 3 o más elementos adyacentes estrictamente crecientes.

1. Define un predicado $noMásDeDos(v, p, q)$ que se evalúe a cierto si y solo si entre las posiciones p (incluida) y q (excluida) no hay 3 o más elementos adyacentes estrictamente crecientes.
2. Utilizando el predicado $noMásDeDos$, especifica una función que dado un entero $k \geq 3$ y un vector v de enteros de longitud $\geq k$, devuelva el número de segmentos de longitud k que no tengan ningún subsegmento (posiciones consecutivas del vector) con 3 o más elementos adyacentes estrictamente crecientes.
3. Diseña e implementa un algoritmo iterativo que resuelva el problema propuesto. Para ello es importante llevar registrada la última aparición de tres elementos adyacentes estrictamente crecientes.
4. Escribe el invariante del bucle principal que permite demostrar la corrección del mismo y proporciona una función de cota.
5. Indica el coste asintótico del algoritmo en el caso peor y justifica adecuadamente tu respuesta.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de dos líneas, en la primera se indica el número de valores del vector y el tamaño k del intervalo. En la línea siguiente se muestran los valores del vector. La entrada termina con un valor cero que no debe procesarse.

La longitud del intervalo k es mayor o igual que 3 y menor o igual que la longitud del vector.

Salida

Para cada caso de prueba se muestra en una línea el número de segmentos de tamaño k que no tienen ningún subsegmento con 3 o más valores adyacentes estrictamente crecientes.

Entrada de ejemplo

```
6 4
5 8 9 4 6 10
7 5
-8 -3 5 8 15 20 25
5 3
8 6 3 -1 -5
7 4
3 5 2 6 4 8 5
5 3
2 2 2 2 2
0
```

Salida de ejemplo

```
1
0
3
4
3
```

2.(2.5 puntos) Doraemon ha construido un nuevo invento para Novita. Se trata de un saltador con el que se puede subir escaleras saltando muchos escalones al tiempo. Novita no ha perdido el tiempo y se ha dirigido rápidamente al parque, donde hay una escalera muy larga, para probarlo. Mientras tanto Gigante, que se encuentra en lo alto de la escalera, se dispone bajar. Nunca lo hace de escalón en escalón sino que se apoya en el pasamanos para dar grandes saltos que le permitan llegar abajo en menos tiempo. Dados los escalones que pisará cada uno cuando suban o bajen la escalera, ¿podrías decirnos si se encontrarán en un escalón y podrán hablar un rato, o si se cruzarán mientras saltan? Suponemos que todos los saltos consumen el mismo tiempo independientemente del número de escalones que se salten y de quién de los saltos.

Se cumple que las dos secuencias de escalones tienen la misma longitud. La primera secuencia empieza siempre en cero y termina en lo alto de la escalera. La segunda secuencia empieza en lo alto de la escalera y termina en cero. Si Novita o Gigante utilizan menos escalones uno que el otro, se repetirá el primer o último valor, dependiendo del momento en que empiezan a bajar o subir.

En el primer ejemplo, Novita parte de la base de la escalera, valor 0. Da un salto en la base de la escalera antes de empezar a subir (segundo 0) y luego pisa los escalones 3, 4 y 6, que es la cima de la escalera. Por su parte Gigante parte de la cima de la escalera, valor 6, y salta al escalón 4 cuando Novita todavía está en la base. Después va al escalón 3, donde se encuentra con Novita que ha saltado a este escalón en este momento. Por último salta a la base de la escalera y se va para su casa (los dos valores cero). En el segundo ejemplo, Novita salta a los escalones 3, 4 y 6. Por su parte Gigante pasa al escalón 4 cuando Novita está en el 3 y al escalón 3 cuando Novita salta al 4. En este momento se cruzan, Novita subiendo del 3 al 4 y Gigante bajando del 4 al 3.

Se pide:

1. Escribe un algoritmo recursivo eficiente del tipo divide y vencerás que resuelva el problema. No está permitido usar memoria adicional proporcional o superior a n .
2. Escribe la recurrencia que corresponde al coste de la función recursiva e indica a qué orden de complejidad asintótica pertenece dicho coste.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de tres líneas: en la primera se indica el número de escalones que se muestran en las dos líneas siguientes. En la línea siguiente se indican los escalones que pisará Novita al subir la escalera (valores iguales al principio indican que Novita tarda más en empezar a subir que Gigante en empezar a bajar y valores iguales al final de la secuencia indican que llega al final de la escalera antes que Gigante llegue abajo). En la tercera línea se indican los escalones que pisará Gigante al bajar la escalera. Igual que en el caso de Novita pueden aparecer valores iguales al comienzo o al final de la secuencia.

El número de escalones es un valor mayor que cero y menor que 100.000.

Salida

Para cada caso de prueba se escribe en una línea **HABLAN** si existe un escalón en que se encuentran los dos, seguido del escalón en que se encuentran, y **SE CRUZAN** si se cruzan mientras saltan, seguido de los dos escalones en los que pisa Novita antes y después de cruzarse.

Entrada de ejemplo

```
5
0 0 3 4 6
6 4 3 0 0
4
0 3 4 6
6 4 2 0
4
0 4 10 10
10 0 0 0
4
0 0 0 10
10 0 0 0
9
0 0 0 4 8 10 14 14 14
14 12 10 9 7 4 2 0 0
0
```

Salida de ejemplo

```
HABLAN 3
SE CRUZAN 3 4
SE CRUZAN 0 4
HABLAN 0
SE CRUZAN 4 8
```

3. (4 puntos) Llega el verano y con él las brigadas antiincendios. Tenemos n montes que vigilar y para ello hemos contratado a m vigilantes. Ahora debemos formar las brigadas antiincendios, una por cada monte a vigilar. Numeraremos las brigadas según la numeración del monte al que son asignadas, así la brigada 1 irá al monte 1, la brigada 2 al monte 2, etc. Cada monte necesita un número mínimo de vigilantes, aunque la brigada encargada del monte puede estar formada por más vigilantes de los estrictamente necesarios.

Todos los vigilantes reciben el mismo salario, pero tienen derecho a un plus de movilidad que es mayor cuanto mayor es el desplazamiento del vigilante desde su casa hasta el monte que le toque vigilar. Por ello, no permitiremos que un vigilante forme parte de una brigada asignada a un monte que esté más lejos de su casa que un cierto valor.

Se pide diseñar e implementar un algoritmo de vuelta atrás que resuelva el problema de formar las brigadas de forma que todos los vigilantes estén asignados a una brigada y se minimice la suma de los desplazamientos que deban realizar. Describe claramente el espacio de soluciones y los marcadores utilizados. Se valorará realizar poda de optimalidad.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número de brigadas que necesitamos, n , y el número de vigilantes que hemos contratado m . A continuación en una línea se muestra el tamaño mínimo de cada una de las brigadas. En la línea siguiente, la distancia máxima que permitiremos en un desplazamiento. Y por último, en las m líneas siguientes se muestran n valores correspondientes a la distancia de cada vigilante a cada uno de los montes. El final de los datos se marca con dos ceros.

El número de brigadas es mayor que cero y menor que 10, y el número de vigilantes es mayor que cero y menor que 18. El tamaño mínimo de una brigada es un valor mayor que cero y menor o igual que el total de vigilantes. Las distancias son todas ellas valores enteros positivos menores de 1000.

Salida

Para cada caso de prueba se escribe la suma de todos los desplazamientos que deben realizar los vigilantes para llegar a su lugar de trabajo. Si no es posible formar las brigadas se escribirá IMPOSIBLE.



Entrada de ejemplo

```
2 3
1 1
20
10 5
5 15
8 6

2 3
1 1
6
10 5
5 15
8 7

2 5
1 3
10
5 5
10 20
1 5
5 5
3 9
0 0
```

Salida de ejemplo

```
16
IMPOSIBLE
28
```