Fundamentos de Algoritmia

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

Examen Convocatoria Extraordinaria, 21 de junio de 2023.

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`.
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 v de $n \geq 0$ enteros, se desea contar el número de segmentos de longitud al menos dos que cumplen que sus elementos son consecutivos, es decir, cada elemento del segmento es uno más o uno menos que el situado en la posición anterior.

   1. Define un predicado *todosConsecutivos*(v, p, q) que devuelva cierto si y solo si todos los elementos del vector v contenidos entre las posiciones p (incluida) y q (excluida) son consecutivos.
   2. Utilizando el predicado *todosConsecutivos*, especifica una función que dado un vector de enteros de longitud ≥ 0, devuelva el número de segmentos de longitud al menos dos que cumplen que todos sus elementos son consecutivos.
   3. Diseña e implementa un algoritmo iterativo que resuelva el problema propuesto.
   4. Escribe el invariante del bucle 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 comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá el valor del número de elementos n del vector, y a continuación los elementos del vector.

Salida

Por cada caso de prueba el programa escribirá una línea con el número de segmentos solicitado en el enunciado.

Entrada de ejemplo

```
6
0
1
-7
3
1 3 5
3
6 7 6
4
1 2 3 4
10
8 2 3 4 1 2 1 2 3 -1
```

Salida de ejemplo

```
0
0
0
3
6
13
```

2.(2.5 puntos) Disponemos de un entero *ini* a partir del cual hemos generado una secuencia de enteros consecutivos estrictamente creciente. Posteriormente hemos borrado algunos de dichos enteros, y la secuencia resultante está almacenada en un vector v. Dado v y dado *ini*, se desea encontrar el menor entero de los que se han borrado.

Se pide:

1. Escribe un algoritmo recursivo eficiente que resuelva el problema.
2. Escribe la recurrencia que corresponde al coste de la función recursiva.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso se presenta en dos líneas. En la primera se indica el número de valores que quedan en la secuencia, seguido del valor inicial a partir del cual se generó la secuencia. En la siguiente línea figuran los valores.

Se garantiza que todos los valores son menores de 10^9.

Salida

Para cada caso de prueba se escribe el menor valor entero que se ha borrado.

Entrada de ejemplo

```
5
0 3
1 5
5
1 5
8
4 2
2 3 4 9
6 -2
-2 0 1 5 8 20
```

Salida de ejemplo

```
3
6
5
5
-1
```

3. (4 puntos) Una empresa de consultoría informática desea distribuir m equipos de trabajo en n proyectos. El número de personas disponibles en cada equipo viene dado por un vector E. Cada equipo ha de ser asignado exactamente a un proyecto pero un proyecto puede requerir de varios equipos para cubrir sus necesidades. De cada proyecto se conoce el número de personas que se necesitan como mínimo para poder llevarlo a cabo y el máximo número de personas admisibles en dicho proyecto. Esta información viene dada en dos vectores L y G (se puede asumir $G[i] \geq L[i]$).

El sueldo de los trabajadores dependerá del proyecto en la que desempeñarán su labor, y dicha información vendrá dada en un vector de sueldos S (dado un proyecto j, $S[j]$ es el sueldo por persona en dicho proyecto). Aunque el número de personas supere las necesidades mínimas de un proyecto, los equipos se mandan completos así que se pagará el sueldo a todas las personas que conforman los equipos asignados según el proyecto en el que se encuentren destinadas.

Se pide diseñar e implementar un algoritmo de vuelta atrás que resuelva el problema de asignar equipos a proyectos de forma que todos los equipos estén asignados a un proyecto, todos los proyectos tengan sus necesidades mínimas cubiertas pero sin pasarse de los máximos admitidos por los responsables, y se minimice el sueldo total a pagar. Plantea al menos una función de poda de optimalidad e impleméntala en tu algoritmo.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá inicialmente el valor del número de equipos m y de proyectos n. A continuación una línea con el número de personas de cada equipo, y otras tres conteniendo respectivamente las necesidades mínimas y máximas, y los sueldos de cada proyecto.

Salida

Por cada caso de prueba el programa escribirá el sueldo mínimo a pagar, o NO en caso de que no sea posible llevar a cabo la asignación de equipos a proyectos.

Entrada de ejemplo

```
3
3 2
2 5 3
4 6
7 9
5 10
3 2
4 5 6
4 6
5 7
2 5
4 2
2 4 3 5
4 5
10 8
2 5
```

Salida de ejemplo

```
NO
NO
43
```