Fundamentos de Algoritmos

Grados en Ingeniería Informática

Convocatoria ordinaria, 4 de febrero de 2021. Grupos B y D

1. (3.5 puntos) Dados dos números enteros, A y B, se dice que B es *múltiplo* de A cuando existe un tercer entero C tal que $A = B \times C$. Por ejemplo, 10 es múltiplo de 5 y 40 es múltiplo de 20.

Por su parte, una *secuencia de múltiplos* es una sucesión de números enteros en las que todos los números, a excepción del primero, son múltiplos del anterior. Un ejemplo de este tipo de sucesiones sería

```
5 10 20 40 80 160
```

Diseña un algoritmo iterativo eficiente que, dado un vector de enteros, que *no* contiene ceros, y un número natural $K > 0$ determine *cuántas* secuencias de múltiplos de longitud K tiene el vector. Debes, asimismo, determinar justificadamente el orden de complejidad del algoritmo.

La implementación deberá ir acompañada de un programa de prueba, que lea desde la entrada estandar casos de prueba, los ejecute, e imprima por la salida estándar el resultado. Cada caso de prueba consistirá en dos líneas, la primera con el número de elementos del vector (hasta 300.000) y el número K y la segunda con los elementos del vector en sí. Para cada caso de prueba el programa imprimirá el número de secuencias de múltiplos de longitud K que contiene el vector. La entrada finalizará con una línea con `-1`. A continuación se muestra un ejemplo de entrada/salida:

Entrada	**Salida**
8 4 9 5 10 20 40 80 160 320 4 2 2 4 6 12 4 1 2 4 6 12 -1	4 2 4

2. (3 puntos) Sobre los dígitos del 0 al 9 se definen las dos siguientes operaciones:

$$subir(d) = \begin{cases} d+1 & \text{si } d < 9 \\ 0 & \text{si } d = 9 \end{cases}$$

$$bajar(d) = \begin{cases} d-1 & \text{si } d > 0 \\ 9 & \text{si } d = 0 \end{cases}$$

Un número natural se llama *escalonado* cuando cada dígito, a excepción del primero, se obtiene a partir del anterior mediante una operacion *subir* o una operación *bajar*. Por ejemplo, 565432109 es escalonado, mientras que 565432102 no lo es (ya que el último dígito, 2, no puede obtenerse desde el penúltimo dígito, 0, con ninguna de las dos operaciones).

Diseña un algoritmo recursivo eficiente que, dado un número N, determine cuántos números escalonados menores que N hay. Debes, así mismo, determinar la complejidad de dicho algoritmo, planteando y resolviendo las recurrencias apropiadas.

El algoritmo irá acompañado por un programa que lee casos de prueba por la entrada estándar, e imprime el resultado por la salida estándar. Cada caso de prueba consistirá en el número N (de tipo `long long`). El programa escribirá, entonces, cuántos números escalonados menores que N hay. La entrada finalizará con una línea con `-1`. A continuación se muestra un ejemplo de entrada/salida:

Entrada	Salida
6 17 257 5678 19876 345678 565432102 -1	6 12 36 103 152 375 3449

3. (3.5 puntos) La empresa de mudanzas *YoTePorto* nos ha encargado automatizar algunos de sus procesos logísticos.

En particular, tenemos que implementar una aplicación que, dada una serie de objetos que necesitan transportarse, nos permita determinar el número mínimo de cajas necesarias para realizar la mudanza. Todas las cajas tienen la misma capacidad C (número natural positivo). Asimismo, cada objeto tiene un determinado tamaño T (número natural positivo menor o igual que C). La suma de los tamaños de los objetos almacenados en una caja no puede exceder la capacidad máxima C de la caja.

Debemos diseñar un algoritmo "vuelta atrás" que, dados (i) la capacidad C de las cajas; y (ii) los espacios requeridos por cada uno de los objetos a portar, determine el mínimo número de cajas necesarias para realizar la mudanza.

El algoritmo irá acompañado por un programa que lee casos de prueba por la entrada estándar, e imprime el resultado por la salida estándar. Cada caso de prueba consistirá en dos líneas. En la primera línea aparecerá la capacidad de las cajas C y el número de objetos a portar N ($0 \leq N \leq 20$). En la segunda línea aparecerán los N tamaños de los objetos. La entrada termina con una línea que contiene únicamente -1.

A continuación se muestra un ejemplo de entrada / salida:

Entrada	Salida
10 5 2 6 3 8 9 15 6 3 10 10 3 4 13 -1	3 4