

Fundamentos de Algoritmos. Grupo E

Práctica 2: Algoritmos recursivos

7 de Noviembre de 2024

1. Debéis entregar la práctica durante la hora de laboratorio.
2. En el campus disponéis de un fichero plantilla para hacer la práctica. **Usadla, con eso evitaréis errores innecesarios.**
3. **IMPORTANTE:** asegúrate de que el Visual Studio tiene configurado el **fin de línea como CRLF**
4. Escribid el **nombre y apellidos** de los dos miembros de la pareja que estéis haciendo la práctica. Si no los incluís, el ejercicio se evaluará con un 0.
5. Debéis desarrollar e implementar una solución eficiente y entregarla en el juez automático. **Sólo es necesario que suba el código uno de los dos miembros.**
6. Vuestra solución será evaluada por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayáis realizado como pareja.
7. Durante el tiempo de laboratorio, en el juez únicamente está subido el caso de prueba del enunciado. Esto significa que un resultado de **correct** es **provisional**.
8. A lo largo del examen se os pedirá que os identifiquéis y rellenéis vuestros datos en una hoja de firma.

Dígitos obtusos

Pascual empezó muy contento en la empresa en la que hace prácticas, pero un mes después está cansado de tanto Excel así que le ha pedido a su superior que le ponga una tarea más interesante. Ante tal disposición el jefe le ha propuesto automatizar una tarea que hasta ahora hacían a mano: conocer la cantidad de dígitos *obtusos*, como son llamados, en los números identificadores de los clientes.

Un dígito de un número natural n se dice que es *obtuso* si es igual al producto, módulo 10, de los dígitos con valores impares que son más significativos que él. Además, el dígito más significativo es obtuso únicamente si es 1.

Por ejemplo, en el número 371323 hay 2 dígitos obtusos, el último $3 = (3 * 7 * 1 * 3) \bmod 10$ y el 1 $= (3 * 7) \bmod 10$. Como decíamos antes, la tarea de Pascual es la de conocer la cantidad de dígitos obtusos de un número dado.

Se pide:

1. (8.5 puntos) Escribe un algoritmo recursivo eficiente que permita resolver el problema para un número n dado. No está permitido almacenar en un vector auxiliar o una variable `string` los dígitos del número.
2. (1.5 puntos) Escribe la recurrencia que corresponde al coste de la función recursiva utilizando el número de dígitos de n como tamaño del problema. Indica también a qué orden de complejidad asintótica pertenece dicho coste.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de consta de un número n . Su valor está garantizado que cabe en una variable entera.

Salida

Para cada caso de prueba el programa escribirá la cantidad de dígitos obtusos del número n dado.

Entrada de ejemplo

```
5
371323
1111
7793113
339
22222
```

Salida de ejemplo

```
2
4
3
2
0
```