

Reverse And Add

Diego Alfonso Prieto Torres - Sebastian Camilo Martinez Reyes

10 de diciembre de 2012

Índice

1. Contextualizacion	1
1.1. Objetivos	2
1.2. Precondicion	2
1.3. Poscondicion	2
2. Definicion del Problema	2
2.1. Definicion de Conceptos	2
2.2. Introduccion al Problema	3
3. Modelamiento de la Solucion	3
3.1. Estrategia de la Solucion	3
3.2. Leve Nocion de Estructura de Datos	3
4. Conclusiones	3

1. Contextualizacion

El problema de Reverse And Add es un problema usado en maratones de programacion cuyo enunciado puede encontrarse actualmente en el Juez en Linea de la UVA identificado con el codigo 10018. Este es un problema que tiene como objetivo jugar con los numeros, y con los conceptos de cadenas, asi que no esta clasificado dentro de la teoria de numeros, pero es excelente para revisar la programacion con cadenas y ciclos sencillos.

1.1. Objetivos

- Usar los conceptos de palindrome y un algoritmo sencillo de invertir numeros para hallar la solucion
- Invertir un numero y sumarselo a si mismo en el estado anterior, tantas veces sea necesario hasta que el resultado sea un numero palindrome.

1.2. Precondicion

- un numero N tal que ese numero tiene respuesta dentro del programa.
- El numero de sumas sucesivas que se deben hacer para hallar la solucion no sobrepasa las 1000 iteraciones.
- El numero palindrome resultante no es mas grande que 4, 294, 994, 924.

1.3. Poscondicion

Se debe informar con dos numeros I y P , tal que I es igual al numero de sumas sucesivas y P es el palindrome resultante.

2. Definicion del Problema

2.1. Definicion de Conceptos

Se entiende como numero palindrome un numero tal que tiene el mismo valor numerico si es leido de derecha a izquierda o de izquierda a derecha.

Si pensamos que cada digito del numero se encuentra dentro de un arreglo especificado como $f[0 \times M)$ interpretariamos como un numero palindrome lo siguiente:

$$(\forall i \mid 0 \leq i \leq M/2 : f_i = f_{M-1-i})$$

Tambien definimos el reverso de un numero como girar simetricamente los digitos de este; es decir. Si definimos el arreglo $r[0 \times M]$, interpretamos como reverso lo siguiente:

$$(\forall i \mid 0 \leq i < M : r_i = f_{M-1-i})$$

2.2. Introduccion al Problema

3. Modelamiento de la Solucion

La solucion consiste en un algoritmo itearativo con una unica condicion de ruptura; que los arrellos r y f sean iguales.

La forma de conseguir este logro es sumar a un numero sucesivas veces su reverso, de tal manera que en algun momento se cumpla la condicion de ruptura del ciclo. Aunque debemos ir contando cuantas veces hacemos el procedimiento de hacer el reverso de un numero, ya que este contador tambien es requisito de la informacion final de la solucion.

3.1. Estrategia de la Solucion

La estrategia es muy sencilla, haciendo uso de dos funciones, una que me calcule el reverso de un numero y otra que me informe si un numero dado es o no palindrome, lograremos hallar de forma ciclica la solucion. No hay que olvidar ademas que los valores de entrada tienen respuesta.

3.2. Leve Nocion de Estructura de Datos

Si usas un compilador como Java o C++, hay un ahorro impresionante de implementacion frente a lo que es el proceso de invertir el numero y saber si es o no palindrome, dejando la complejidad del programa en su minima expresion.

4. Conclusiones

Es muy eficiente el manejo de cadenas por medio de las clases o tag String que nos ofrecen compiladores como Java y C++, ahorrandonos mucho tiempo de ejecucion, sin embargo, ya que la complejidad algoritmica de este problema esta mas centrada en el manejo de cadenas que en los caculos recurrentes o dinamicos.