Maximun

Diego Alfonso Prieto Torres - Sebastian Camilo Martinez Reyes

22 de noviembre de 2012

Índice

1. Contextualizacion

El problema de Maximum es un problema usado en maratones de programacion cuyo enunciado puede encontrarse actualmente en el Juez en Linea TIMUS identificado con el codigo 1079. Este es un problema que esta clasificado dentro de la teoria de numeros, y por lo tanto, es la oportunidad perfecta para desafiar los conocimientos matematicos, combinados con las habilidades en programacion. Es un problema excelente para explotar los conocimientos de recurrencia y de sucesiones, que son, en gran medida, la base para el modelamiento de muchas de las soluciones de problemas.

1.1. Objetivos

- Comprender como se puede modelar la recurrencia para dar solucion a los problemas que puedan llegarse a plantear.
- Encontrar el valor mas grande de una secuencia de numeros en el intervalo $[0, N]$.
- implementar un algoritmo recurrente apoyado en la teoria de numeros y la matematica.

1.2. Precondicion

Un numero N tal que $0 \leq N \leq$ que indica en N *esimo* indice de la secuencia, hasta donde deberemos tenerla en cuenta.

1.3. Poscondicion

Un numero M que representa el mayor valor de la secuencia de los indices que esten en el intervalo $0 \leq i \leq N$

2. Definicion del Problema

2.1. Definicion de Conceptos

Se conoce la sucesion a trabajar definida como:

$$a_i = \begin{cases} 0, & \text{si } i = 0 \\ 1, & \text{si } i = 1 \end{cases}$$

Ademas se debe cumplir que:

$$(\forall i \mid 1 \leq i : a_i = a_{2*i} \wedge a_i + a_{i+1} = a_{(2*i)+1})$$

Se sostiene que es una sucesion divergente.

2.2. Introduccion al Problema

El objetivo del problema es encontrar el a_i mas grande de los valores de la sucesion entre $1 \leq i \leq N$. Es decir:

$$M = (\uparrow i \mid 1 \leq i : a_i)$$

3. Modelamiento de la Solucion

3.1. Estrategia de la Solucion

La estrategia de la solucion es muy sencilla y nos basamos en la recurrencia para ello.
Primero que nada debemos hallar los valores de toda la sucesion desde 1 hasta N; por lo cual hacemos uso de la siguiente funcion recurrente:

$$secuencia(i) = \begin{cases} 0, & \text{si } i = 0 \\ 1, & \text{si } i = 1 \\ secuencia(i/2), & 1 \bmod 2 = 0 \\ secuencia(i+1) + secuencia(i-1), & 1 \bmod 2 = 1 \end{cases}$$

Cuando ya obtenemos todos los valores de la sucesion desde 0 hasta el valor N indicado, porcedemos a construir un algoritmo de busqueda sencillo sobre los valores, con el objetivo de rescatar el valor mas grande de la sucesion de numeros.

3.2. Leve Nocion de Estructura de Datos

Un arreglo donde se depositaran los valores de la sucesion manteniendo el indice del vector como el indice de la sucecion.

Por otro lado la implementacion de la sucesion y del algoritmo de busqueda, la dejamos a preferencia y definicion del lector, ya que es un buen ejercicio mental a nivel matematico y como programador.

4. Conclusiones

Los algoritmos recurrentes son una herramienta muy eficaz en la solucion del problema, aunque generalmente son conocidos como implementaciones lentas y con un gasto de memoria(espacio) alto, sin embargo el beneficio es asegurar la respuesta de manera eficas y sin operaciones adicionales, ademas se toma como evidencia su certeza en la implementacion al habernos apoyado en su teoria para solucionar este problema.

Para que un programador pueda enfocar una solucion recurrente, es necesario que tenga los conocimientos matematicos ligados a su analisis cognitivo,

por ello seguimos insistiendo en el desarrollo de las habilidades matematicas junto con el analisis para la solucion de problemas como una estrategia eficaz sobre la evolucion del prgramador.