Stone Pile

Diego Alfonso Prieto Torres - Sebastian Camilo Martinez Reyes

25 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 1005. Este es un problema que esta clasificado dentro de la busqueda exahustiva, y por lo tanto, requiere de metodologias de busqueda o analisis de casos para obtener una solucion eficiente. Es un problema excelente para explotar los conocimientos basicos de las estructuras de datos como arreglos, y del manejo de ciclos y condicionales.

1.1. Objetivos

- Identificar e implementar un algoritmo de busqueda exahustiva para hallar la solucion del problema.
- Agrupar en dos grupos una serie de numeros de forma tal que la diferenacia entre estos dos grupos sea minima.

1.2. Precondicion

Un entero N tal que $1 \leq N \leq 20$ que representa el numero de piedras, y para cada piedra indican su peso W tal que $1 \leq W \leq 100000$.

1.3. Poscondicion

Un entero R que representa la minima diferencia entre los grupos de numeros.

2. Definicion del Problema

2.1. Definicion de Conceptos

Tenemos una serie de numeros, bien sea denominada como S, donde los elementos se pueden repetir.

Se define la sumatoria de la serie S como un numero T_{S_n} tal que:

$$T_{S_n} = (+i \mid i \in S_n : S_n.i)$$

Por ejemplo S puede ser igual a $13, 14, 5, 8, 27$.

2.2. Introduccion al Problema

El objetivo del programa debe ser encontrar un par de series S_1 y S_2, tal que:

$$(\forall i \mid 0 \leq i < N \land i \in S : S.i \in S_1 \Rightarrow S.i \notin S_2)$$

en donde R es la respuesta definida como:

$$R = (\downarrow i, j \mid T_{S_i} - T_{S_j})$$

3. Modelamiento de la Solucion

3.1. Estrategia de la Solucion

Como no existe una solucion polinomica que podamos aplicar para hallar la solucion, realizaremos una busqueda exhaustiva de todas las formas en que podemos dividir los pesos de las piedras en dos grupos, y asi ir buscando la diferencia minima de esos 2 grupos.

A cada elemento, le podemos asignar un 1 para indicar que se va aadir al primer grupo, y 0 para el segundo grupo:

14 27 13 8 5 Grupo1 Grupo2
0 0 0 0 0 : 67 0
0 0 0 0 1 : 5 62
0 0 0 1 0 : 8 59
0 0 0 1 1 : 13 54

Si notamos, particularmente, cada agrupamiento de numeros, se puede representar por medio de un numero binario, que es siempre igual al anterior, despues de agregar o quitar un bit. De esta forma podemos ir calculando el peso de la diferencia de la distribucion a medida que realizamos el proceso de aadir o quitar un bit; escogiendo siempre la mas baja.

3.2. Leve Nocion de Estructura de Datos

La estructura de datos es muy intuitiva, con un arreglo de tamao N de tipo boolean, podemos generar el valor de verdad sobre cada digito, que es, en escencia lo que representa in bit.

4. Conclusiones

Una virtud que tienen los algoritmos de busqueda exahustiva es que encuentran la solucion exacta, sin embargo lo hacen en un tiempo muy extenso, para evitar esto, siempre es bueno una estrategia de representacion de los datos, en los cuales se eviten los ciclos anidados y se permita llevar un control sobre el proceso, disminuyendo al minimo el proceso del programa.