DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP2

Algoritmos y Estructuras de Datos III

Integrante	LU	Correo electrónico
Ezequiel Aguerre	246/07	`ezeaguerre@gmail.com`
Juan Vanecek	169//10	`juann.vanecek@gmail.com`
Santiago Camacho	110/09	`santicamacho90@gmail.com`
Tomas Rodriguez	527/10	`tomirodriguez.89@gmail.com`

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

1. Problema 1: Robanúmeros

1.1. Presentación del problema

1.2. Resolución

1.2.1. 1era solucion

En cada mano, agarrar la cantidad de cartas que sumen mas.

No funciona. Contraejemplo: $[-1, -6, -10]$

1.2.2. 2da solucion

Sea v un vector de n enteros que son los valores de las cartas, y dada la funcion recursiva:

$$f(i,j,p) = \max\left(\underbrace{\max_{1\leq k\leq j}\left\{ \left(\sum_{t=i}^{k} v_t\right) p + f\left(k, j, (p+1) \bmod 2\right)\right\}}_{\text{Lo mejor de la izquierda}}, \underbrace{\max_{1\leq k\leq j}\left\{ \left(\sum_{t=k}^{j} v_t\right) p + f\left(i, k, (p+1) \bmod 2\right)\right\}}_{\text{Lo mejor de la derecha}} \right)$$

La solución al problema es $f(1, n, 0)$ donde el 3er parámetro puede ser 0 o 1, y representa al jugador

1.2.3. 3ra solucion

La solucion es el valor $\max(f_{\rightarrow}(1, n, yo), f_{\leftarrow}(1, n, yo))$ donde las funciones internas se definen como:

$$f_{\rightarrow}(i,j,p) = \begin{cases} \max\limits_{1\leq k\leq j}\left\{ \sum\limits_{t=i}^{k} v[t] + \min(f_{\rightarrow}(k+1, j, p.next()), f_{\leftarrow}(k+1, j, p.next()))\right\} & \text{si } p = yo \\ \min\limits_{1\leq k\leq j}\left\{ \max(f_{\rightarrow}(k+1, j, p.next()), f_{\leftarrow}(k+1, j, p.next())\right\} & \text{si } p = tu \end{cases}$$

$$f_{\leftarrow}(i,j,p) = \begin{cases} \max\limits_{1\leq k\leq j}\left\{ \sum\limits_{t=k}^{j} v[t] + \min(f_{\rightarrow}(i, k-1, p.next()), f_{\leftarrow}(i, k-1, p.next()))\right\} & \text{si } p = yo \\ \min\limits_{1\leq k\leq j}\left\{ \max(f_{\rightarrow}(i, k-1, p.next()), f_{\leftarrow}(i, k-1, p.next())\right\} & \text{si } p = tu \end{cases}$$

1.2.4. 4ta solucion

Defino la funcion $f(i,j)$ como la solucion optima usando de las cartas i a j.

Esto es lo maximo que puedo agarrar con las cartas de la izquierda o de la derecha.

Supongamos que por la izquierda lo mejor que puedo hacer es usando las primeras k cartas. Significa que el valor que puedo tomar es $\sum_{t=i}^{k} carta[t]$ más lo que me deja tomar el otro jugador (que va a jugar optimamente) en la mitad $[k+1,j]$. Esto es el total que suma las cartas en la mitad ya dicha, menos $f(k+1,j)$, ya que es el valor optimo. La función queda asi:

$$
\begin{aligned}
f(i,j) &= \max\left(\max_{i\le k\le j}\left\{\sum_{t=i}^{k} v[t] + \left(\sum_{t=k+1}^{j} v[t] - f(k+1,j)\right)\right\}, \max_{i\le k\le j}\left\{\sum_{t=k}^{j} v[t] + \left(\sum_{t=i}^{k-1} v[t] - f(i,k-1)\right)\right\}\right)\\
&= \max\left(\max_{i\le k\le j}\left\{\sum_{t=i}^{j} v[t] - f(k+1,j)\right\}, \max_{i\le k\le j}\left\{\sum_{t=i}^{j} v[t] - f(i,k-1)\right\}\right)\\
&= \sum_{t=i}^{j} v[t] + \max\left(\max_{i\le k\le j}\{-f(k+1,j)\}, \max_{i\le k\le j}\{-f(i,k-1)\}\right)\\
&= \sum_{t=i}^{j} v[t] - \min\left(\min_{i\le k\le j}\{f(k+1,j)\}, \min_{i\le k\le j}\{f(i,k-1)\}\right)\\
&= \sum_{t=i}^{j} v[t] - \min_{i\le k\le j}\{\min\left(f(k+1,j), f(i,k-1)\right)\}
\end{aligned}
$$

1.3. Demostración

1.4. Análisis de complejidad

1.5. Test de complejidad

1.6. Compilar y ejecutar

2. Problema 2: La centralita (de gas)

2.1. Presentación del problema

2.2. Resolución

2.3. Demostración

2.4. Análisis de complejidad

2.5. Test de complejidad

2.6. Compilar y ejecutar

3. Problema 3: Saltos en *La Matrix*

3.1. Presentación del problema

3.2. Resolución

3.3. Demostración

3.4. Análisis de complejidad

3.5. Test de complejidad

3.6. Compilar y ejecutar