

# Problema B

Valerie Parra Cortés 201619703  
Cristina Isabel Gonzalez Osorio 201520019

December 11, 2018

## 1 Identificación del problema

Para este problema se solucionó un algoritmo de programación dinámica. En cada periodo se tiene 3 opciones: invertir en la bolsa A, invertir en la bolsa B y no invertir en ninguna bolsa. Si se invierte en la bolsa A, se selecciona el máximo capital obtenido en el periodo anterior entre la bolsa B y no invertir y se calcula el nuevo capital correspondiente. Si se invierte en la bolsa B o se decide no invertir, se selecciona el máximo capital obtenido en el periodo anterior (independientemente de que en bolsa se obtuvo) y se calcula el nuevo capital. Formalmente, si  $c$  representa el capital inicial,  $i=[0, 1 \dots n]$  los periodos y  $j=[1 \ 2 \ 3]$  donde representa la bolsa A, 2 la bolsa B y 3 no invertir en ninguna bolsa, la ecuación de recurrencia del problema se muestra a continuación

$$C(i, j) = \begin{cases} c & \text{si } i = 0 \\ \lfloor \max(C(i-1, 3), C(i-1, 2)) * (1 + a[i]/100) \rfloor & \text{si } i \neq 0 \wedge j = 1 \\ \lfloor \max(C(i-1, 1), C(i-1, 2), C(i-1, 3)) * (1 + a[i]/100) \rfloor & \text{si } i \neq 0 \wedge j = 2 \\ \max(C(i-1, 1), C(i-1, 2), C(i-1, 3)) & \text{si } i \neq 0 \wedge j = 3 \end{cases}$$

El grafo de exploración del problema seria

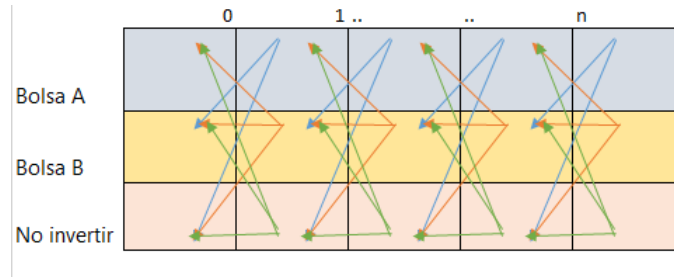


Figure 1: Grafo de necesidades

Entrada	Descripción
n: nat	Número de periodos
a: int[0,n)	Arreglo con las ganancias estimadas de la bolsa a
b:int[0,n)	Arreglo con las ganancias estimabas de la bolsa b
c: nat	Capital inicial

Salida	Descripción
g	Capital máximo en el tiempo n

En cuanto a la implementación, como los periodos sólo dependen del periodo anterior y en cada vuelta del ciclo se van actualizando, sólo se guarda dicho periodo. Para este algoritmo también se hubiera podido implementar una versión Greedy (escoger en cada periodo el que maximiza dicha ganancia), sin embargo aunque la optimización local puede llevarme a la global, no siempre es así, en particular cuando la ganancia de invertir en A en el periodo siguiente es mayor a la invertir en A en el periodo actual. Por ejemplo imagine la tabla 1, donde se muestran las ganancias en porcentajes. Para un capital inicial de 1000 y 3 periodos El grafo de necesidades resultante sería el que se muestra en la tabla 2 donde en color se muestran las decisiones tomadas. Como vemos las mejores decisiones son invertir en B en el periodo 1, no invertir en el periodo 2 e invertir en la A en el periodo 3. Sin embargo un algoritmo Greedy hubiera maximizado la ganancia en el periodo 1, invirtiendo en la bolsa B en el periodo 2 invertiría en la A y el último en la B logrando un capital final de 1348 y no de 2080.

Entrada	Descripción
n: nat	Número de periodos
a: int[0,n)	Arreglo con las ganancias estimadas de la bolsa a
b:int[0,n)	Arreglo con las ganancias estimabas de la bolsa b
c: nat	Capital inicial

Table 1: Caso ejemplo

	0	1	2	3
Bolsa A	1000	1020	1248	2080
Bolsa B	1000	1040	988	1348
No invertir	1000	1000	1040	1248

Table 2: Grafo de necesidades del ejemplo

Otra opción es implementar programación dinámica sobre la matriz completa con todas las bolsas y todos los periodos, pero esto implicaría una inversión exponencial en espacio, por lo que se prefirió la versión implementada.

### **1.1 Complejidad espacial**

La complejidad espacial del problema es  $\theta(1)$  porque solo se guarda la  $i$ -ésima columna.

### **1.2 Complejidad temporal**

En cuanto tiempo, el algoritmo es  $\theta(n)$  porque necesitaremos llegar al tiempo  $n$ .

### **1.3 Comentarios finales**

Este problema se implemento usando la técnica de programación dinámica aprendida en el curso.