

Debido a la inflación, un ahorrador desea invertir todos sus ahorros en depósitos a plazo fijo anuales. Para ello analiza la oferta de este tipo de depósitos, anotando el capital máximo que cada banco permite invertir a cada cliente y el interés anual (TAE) que ofrece.

Considere las estrategias, <primero, el de mayor interés>, <primero el de menor capital máximo> y <primero el de mayor beneficio máximo>.

Aplicuelas a este ejemplo, 44.000 € a invertir en tres depósitos con, respectivamente 1%, 2%, 4% de TAE, y capitales máximos de 30.000 €, 36.000 € y 10.000 €.

¿Garantiza un algoritmo devorador que se maximice el beneficio de su inversión con algunas de estas estrategias? Diseña un algoritmo devorador óptimo.

- Se plantean bien los ejemplos.
- Se explica (si existe) la opción óptima.
- Se identifican los elementos del esquema devorador.
- Se diseña correctamente el Algoritmo.

a) <primero el de mayor interés>  $\rightarrow 4\% \text{ de } 10000 = 400$

beneficio total =  $400 + 680 = 1080$

1º (4%)  $44.000 - 10.000 = 34000$

2º (2%)  $34.000 - 34000 = 0$  (no esijo todo pgnoray suficiente dinero)

$\rightarrow 2\% \text{ de } 34000 = 680$

<primero el de menor capital máximo>  $\rightarrow 4\% \text{ de } 10000 = 400$

beneficio total =  $400 + 300 + 80 = 780$

1º (4%)  $44.000 - 10000 = 34000$

2º (1%)  $34000 - 30000 = 4000$   $\rightarrow 1\% \text{ de } 30000 = 300$

3º (2%)  $4000 - 4000 = 0$   $\rightarrow 2\% \text{ de } 4000 = 80$

<primero el de mayor beneficio máximo>

1º (2%)  $44000 - 36000 = 8000$   $\rightarrow 2\% \text{ de } 36000 = 720$

2º (4%)  $8000 - 8000 = 0$   $\rightarrow 4\% \text{ de } 8000 = 240$

benef. max =  $720 + 240 = 960$

1% de 30000 = 300

2% de 36000 = 720

4% de 10000 = 400

b) El algoritmo q te proporciona mayor beneficio es la 1ª opción.

c) candidatos: lista de bancos (tupla (interes, capitalmax))  
Fselección: banco de mayor interes  
F. objetivo: beneficio  
F. solución: ¿se ha invertido todo el dinero?  
Objetivo: maximizar  
F. factibilidad: X

inversion (C, dinero)  $\rightarrow$  S

$S \leftarrow \emptyset$

ordenar por interés (C)  $\big] O(n \log n)$   
mientras  $C \neq \emptyset \wedge \text{dinero} \neq 0$

(i, c)  $\leftarrow$  extrae-primer(C)  $\big] \Theta(1)$

si  $\text{dinero} > c$

$S \leftarrow S + \frac{i}{100}$

$\text{dinero} \leftarrow \text{dinero} - c$

sino

$S \leftarrow S + \frac{(\text{dinero} \cdot i)}{100}$

$\text{dinero} \leftarrow 0$

$O(n)$

Ejercicio 1. Se dispone de un buque portacontenedores que hay que cargar con tantos contenedores disponibles en la terminal de carga como sea posible, todos de idénticas dimensiones, pero cada uno con su propio peso.

El buque podría cargar todos los contenedores de estar vacíos, pero al no ser el caso, su peso total podría exceder la capacidad de carga del buque, lo que no debe suceder.

Diseñe un algoritmo devorador para intentar resolver el problema. Mejore su eficiencia con preordenación, y luego, con montículo.

- Identificar los elementos del esquema
- Diseñar el algoritmo
- Analizar la eficiencia
- Explicar cómo mejorarlo con las técnicas y analizar las mejoras

a)

Candidatos: contenedores

F. selección: más pequeños

F. objetivo: n° contenedores

F. solución: ¿se alcanza la capacidad de carga?

Objetivo: maximizar

F. factibilidad: ¿se puede poner el contenedor sin hundirse?

b)

Portacontenedores  $(C, \text{capacarga}) \rightarrow S$

$S \leftarrow \emptyset$

mientras

$C \neq \emptyset \wedge \text{capacarga} > 0$

$p \leftarrow \text{selecciona}(C)$

$C \leftarrow S - \{p\}$

si  $p < \text{capacarga}$

$\text{capacarga} \leftarrow \text{capacarga} - p$

$S \leftarrow S \cup \{p\}$

selecciona  $(C) \rightarrow aux$   
 $aux \leftarrow +\infty$

para todo  $p \in C$

si  $p < +\infty$

$aux \leftarrow p$

c)

$O(n^2)$

d)

1 ordena  $(C)$

2  $p \leftarrow \text{extrae\_primero}(C)$

1 crear montículo  $(C)$

2  $p \leftarrow \text{extrae\_primero}(C)$

Ejercicio 2. Suponga que dispone de información acerca de qué usuarios siguen a otros en una red social. Los bulos se propagan a través de cadenas de seguidores. Diseña un algoritmo que determine cuantos usuarios bastan para propagar un bulo entre los usuarios conectados por la cadena de seguidores más larga. Analice su eficiencia espaciotemporal en el peor de los casos.

Rubrica:

- Se explica la tabla de subproblemas resueltos.
- Se diseña un algoritmo que soluciona el problema.
- Además, se analiza su eficiencia temporal.  $\Theta(n^3)$
- Además, se analiza su eficiencia espacial.  $\Theta(n^2)$

TSR (booleana)  
↓  
estructura de datos que almacena las soluciones para no tener q volver a calcularlas.

b) Propagar-bulo ( $A_{dy}, n$ )

$aux \leftarrow -\infty$

desde  $i \leftarrow 1$  hasta  $n$   
desde  $j \leftarrow 1$  hasta  $n$

si  $A_{dy}[i, j]$

$mCostes[i, j] \leftarrow 1$

$Camino[i, j] \leftarrow -1$

sino

$mCostes[i, j] \leftarrow -\infty$

$Camino[i, j] \leftarrow \infty$

pg usamos siempre matrices

matriz  $ady \rightarrow$  matriz  $costes$

↓  
aplicamos Floyd

$\Theta(n^2)$

desde  $i \leftarrow 1$  hasta  $n$

$mCostes[i, i] \leftarrow 0$

$\Theta(n)$

desde  $k \leftarrow 1$  hasta  $n$

desde  $i \leftarrow 1$  hasta  $n$

desde  $j \leftarrow 1$  hasta  $n$

$mCostes[i, i] \leftarrow \max(mCostes[i, i],$

$mCostes[i, k] + mCostes[k, j])$

$Camino[i, i] \leftarrow k$

$\Theta(n^3)$

desde  $i \leftarrow 1$  hasta  $n$   
 desde  $j \leftarrow 1$  hasta  $n$   
 si  $m\text{Costes}(i, j) > aux$   
 $aux = m\text{Costes}(i, j)$   
 $(x, y) \leftarrow (i, j)$

$\Theta(n^2)$

$S \leftarrow \text{total}(\text{camino}, x, y)$

$\Theta(n)$

$\text{Total}(C, i, i) \rightarrow aux$   
 $k \leftarrow C[i, i]$   
 si  $k \neq -1$  (hay un intermediario)  
 $aux \leftarrow 1 + \text{total}(C, i, k) + \text{total}(C, k, i)$   
 sino  
 $S \leftarrow 0$



Ejercicio 2. Una empresa dispone de una serie de centros logísticos, una flota de vehículos permite transportar paquetes entre ellos, para lo que la empresa ha establecido un mapa con las rutas que los conectan.

Al realizar cualquier ruta entre dos centros logísticos, el vehículo asignado debe parar en todos los centros intermedios situados en ella, de forma que pueda procederse a la carga y descarga de los paquetes necesarios.

Diseñe un algoritmo que determine cuántas paradas ha de realizar como MINIMO un vehículo en ruta entre los dos centros más alejados de la red antes de llegar a su destino.

- Diseñar tabla de subproblemas resueltos.
- Diseñar algoritmo.
- 
- Analizar eficiencia temporal y espacial del algoritmo.

```
1 propagar-bul0 (Adj, n)
  aux ← -∞
  desde i ← 1 hasta n
    desde j ← 1 hasta n
      si Adj[i, j]
        mCostes[i, j] ← 1
        Camino[i, j] ← -1
      sino
        mCostes[i, j] ← +∞
        Camino[i, j] ← ∞
```

```
desde i ← 1 hasta n
  mCostes[i, i] ← 0
```

```
desde k ← 1 hasta n
  desde i ← 1 hasta n
    desde j ← 1 hasta n
      mCostes[i, i] ← min(mCostes[i, i],
                          mCostes[i, k] + mCostes[k, j])
      Camino[i, i] ← k
```

desde  $i \leftarrow 1$  hasta  $n$

desde  $j \leftarrow 1$  hasta  $n$

si  $m(\text{steps}(i, j)) > \text{aux}$

$\text{aux} = m(\text{steps}(i, j))$

$(x, y) \leftarrow (i, j)$

$S \leftarrow \text{total}(\text{camino}, x, y)$

$\text{Total}(C, i, 1) \rightarrow \text{aux}$

$k \leftarrow C[i, i]$

si  $k \neq -1$  (hay vno intermedio)

$\text{aux} \leftarrow 1 + \text{total}(C, i, k) + \text{total}(C, k, i)$

Sino

$S \leftarrow 0$