ZITOUNI
Manhal
INFO 1

Algorithmique devoir

3) Travail demandé :

① Algorithme glouton

a) Algo glouton

```
Entrée : S un entier positif
         L un tableau d'entiers positifs

Sortie   T un tableau d'entiers positifs

i ← 0
rendu ← 0
tant que (rendu < S) faire
    si rendu + L[i] > S alors
        i ← i+1
    sinon
        T[i] ← T[i] + 1
        rendu ← rendu + L[i]
    finsi

retourner T
```

b) Complexité en $O(n)$, car au pire des cas, on parcourt tout le tableau L des pièces t_i.

c) Les indépendants sont les solutions intermédiaires de cet algo. Ça représente les cardinalités des ~~pi~~ pièces choisi pour les sous problème.

d) Ce n'est pas un matroïde car ~~il n'y~~ il n'est pas optimal

Par exemple : pour ~~ra~~ rendre 14 euro avec l'ensemble

L(6,4,3,1)

La solution glouton propose les pièces suivantes

{6,6,1,1}

mais la solution optimale est {6,4,4}

② une approche par programmation dynamique

a) L'expression de la valeur optimale du problème à résoudre est $V(n, S)$

b) $\forall i \in \mathbb{N}^*$, on a $V(i, 0)$ est égal à 0 car ça représente le nombre minimal de pièces de types choisis dans $\{t_1, \ldots, t_i\}$ mais de somme 0. C'est évident, il faut 0 pièce pour rendre 0.

Pour $i \geq 1$, $V(i, j) \leq V(i-1, j)$

* Si $V(i, j)$ représente la valeur optimale du problème et si t_i fait partie des pièces choisis afin de construire la solution optimale, alors si on enlève la pièce t_i, la nouvelle expression $V(i-1, j)$ peut être optimal ou pas.

un exemple : Soit $L = \{9, 6, 2, 1\}$ ensemble de pièces et S la somme à rendre est égale à 10 alors

$V(4, 10) = 2$ dont 1 pièce de 9 et 1 pièce de 1

$V(3, 10) = 3$ dont 1 pièces de 6 et 2 pièces de 2.

$V(4, 10) < V(3, 10)$

c) $V(i, j) = \min\left(V(i-1, j),\ m + V(i-1, k)\right)$

pour $(k, m) \in K_{ij}$

Afin de construire cette expression, on commence par dire que si $V(i, j)$ représente la solution optimale alors on a 2 hypothèses :

* si la pièce t_i ne fait pas partie de la solution optimale alors $V(i, j)$ devient $V(i-1, j)$.

* Si la pièce fait partie de la solution optimale alors on prend le minimum ~~de la~~ des valeur des sous problèmes qui ont l'expression $m + V(i-1, k)$ avec tous les combinaisons possible de k, m.

On en déduit que $V(i,j)$ est le minimum de la valeur des deux hypothèse.

d) Algo dynamique

Entrée : S un entier positif (Somme à rendre)
n un entier positif (nombre de pièces)
V un tableau d'entiers relatifs V[0 ... n ; 0 ... S]
initialisé à Nil.
L un tableau d'entiers positifs qui contient les pièces
Sortie : un entier qui représente le minimum de pièce

```
Calcul V(n, S, V, L)
    si V(n, S, V, L) ≠ Nil alors
        retourner V(n, S, V, L)
    Sinon
        si (S == 0 || n == 1)
            si (n == 1 et L(n) > S et S > 0) alors
                V(n, S) ← ∞
            finsi
            sinon si (n == 1 et L(n) <= S)
                si (S % L(n) == 0)
                    V(n, S) = S / L(n)
                Sinon
                    V(n, S) ← ∞
                finsi
            finsi
            Sinon
                V(n, S) ← 0
            finsi
        retourner V(n, S)
    Sinon
        a1 ← Calcul V(n-1, S, V, L)
        min1 ← ∞
```

pour k allant de 0 à S-1 faire
pour i allant de 1 à n faire
si ((S-k) % L(i) == 0) alors
m ← S-k / L(i)
a2 ← calcul V(n-1, k, V, L) + m
sinon
a2 ← ∞
finsi
si (a₂ < min2) alors
min2 = a2
finsi

si (a₁ < min2) alors
min1 = a1
finsi
sinon
min1 = min2
finsi
V(n, S, V, L) ← min1
retourner V(n, S)
finsi
finsinon
fin algo

Complexité en O(nS)