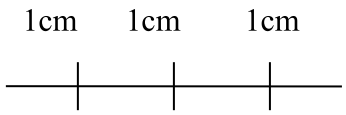
L2 Informatique

INU JF Champollion

TP Programmation dynamique

Exercice 1 : Découpage du bout de bois

On a un morceau de bois d'une longueur entière N que l'on doit couper en n morceaux de longueur 1. Le coût de chaque coupe est égal à la longueur du morceau de bois que l'on cherche a couper. Prenons un exemple pour un morceau de longueur 4 :



Si on coupe au milieu le coût pour la première coupe est de 4 puis il nous faut faire 2 coupes une a la première marque et l'autre à la troisième ; le coût de chacune de ces coupes est de 2, donc le coût total est de 8.
Si par contre on commence a couper à la première marque le premier coût est aussi de 4, puis on va couper au milieu ce qui nous donne un coût de 3, et la dernière coupe a la troisième marque qui aura un coût de 2, donc le coût total est de 9.
On note $Sol(i)$ le coût minimal pour couper un bout de bois de longueur i.

1. Trouver à la main la valeur de $Sol(i)$ pour i allant de 1 à 5.
2. Trouver une relation de récurrence permettant d'exprimer $Sol(n)$ en fonction des $Sol(i)$ avec $i<n$.
3. En déduire un programme en programmation dynamique qui permet de trouver $Sol(n)$.

Pour une longueur de 100 vous devez trouver un coût minimum de 672.

Exercice 2 : parenthésage

On dispose de n chiffres 1. On cherche en utilisant ces chiffres *un*, des additions, des multiplications et des parenthèses à obtenir une expression dont l'évaluation est la plus grande possible. On note $M(n)$ ce nombre.
Par exemple, avec 7 *uns*, on peut faire l'expression $(1+1)\times(1+1+1)+1+1$ et son évaluation est 8.
Mais on peut faire mieux.

1. Trouver à la main la valeur de $M(3)$, $M(4)$, $M(5)$, $M(6)$.
2. On remarque qu'une expression optimale pour n chiffres 1 peut toujours s'écrire de la forme $(A) + (B)$ ou $(A) \times (B)$ où A et B sont alors des expressions optimales pour les problèmes avec respectivement $|A|$ chiffres 1 et $|B|$ chiffres 1. ($|E|$ désigne le nombre de *uns* dans l'expression E.)
   En déduire alors une formule de récurrence pour $M(n)$: Exprimer $M(n)$ en fonction des $M(i)$ avec $i<n$.
3. En déduire un programme de calcul de $M(n)$. Calculer le résultat pour $n=5,6,7,8,..,15$.

Exercice 3 : Problème du sac à dos entier

Écrire alors un programme mettant en œuvre la programmation dynamique pour résoudre le problème du sac à dos entier.

On traitera précisément le cas particulier d'un sac à dos avec 3 types d'articles :

- A de poids 3Kg et de prix 18€,
- B de poids 4Kg et de prix 20€,
- C de poids 2Kg et de prix 8€.

On note `Sol(n)` la valeur maximale d'un sac à dos de capacité `n` kg.

1. Trouver à la main la valeur de `Sol(n)` pour `n` allant de 1 à 5.
2. Trouver une relation de récurrence permettant d'exprimer `Sol(n)` en fonction des `Sol(i)` avec `i<n`.
3. En déduire un programme en programmation dynamique qui permet de trouver `Sol(n)`.
4. Pour un poids du sac à dos de 10Kg vous devez trouver un prix maximum de 56€, pour un poids du sac à dos de 1000Kg vous devez trouver un prix maximum de 5996€.
5. Proposer 2 autres solutions : solution exhaustive (force brute) et solution «gloutonne». Comparer les résultats obtenus et les temps d'exécution.