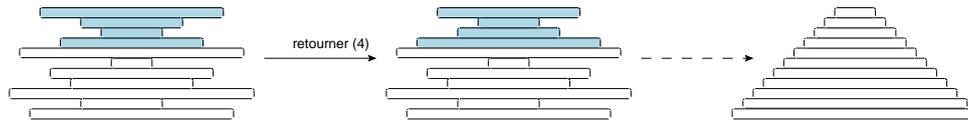


TD n°6 - Partiel 2007

1 Tri crêpes

On veut trier une pile de crêpes de façon à obtenir une pile de crêpes triées par ordre croissant de taille. On dispose pour cela uniquement d'une spatule pour retourner les crêpes. On définit l'opération *retourner*(n) qui consiste à glisser la spatule sous les n premières crêpes et à retourner le haut de la pile.



Question 1.1

Donner un algorithme permettant de trier une pile de n crêpes à l'aide de l'opération *retourner*. On donnera sa complexité au pire cas en nombre d'opérations *retourner*.

Question 1.2

Justifiez la correction de votre algorithme.

Question 1.3

Vous devez désormais trier une pile de crêpes ayant chacune une face brûlée. Pour soigner la présentation on vous demande désormais de trier les crêpes tout en vous assurant que chaque crêpe a sa face brûlée tournée vers le bas. Quelle serait alors la complexité de l'algorithme ?

2 Fanfare

Une fanfare est composée de n musiciens de tailles t_1, t_2, \dots, t_n . Pour les jours de fête l'orchestre dispose de m uniformes ($m \geq n$) de tailles u_1, u_2, \dots, u_m . Chaque année certains musiciens s'en vont et sont remplacés par d'autres, il faut alors ré-attribuer à chaque musicien le costume qui lui sied le mieux.

René, le percussionniste, pense qu'il faut chercher à minimiser la différence moyenne entre la taille d'un musicien et celle de son costume :

$$\frac{1}{n} \sum_{i=1}^n |t_i - u_{\alpha(i)}|$$

où $\alpha(i)$ est l'indice du costume attribué au musicien de taille t_i . Il propose pour cela un algorithme glouton qui consiste à chercher i et j minimisant $|t_i - u_j|$. On attribue alors l'uniforme j au musicien i et on itère jusqu'à ce que tout le monde ait reçu un uniforme.

Question 2.1

L'algorithme de René est-il optimal ?

Anne, la corniste, trouve plus équitable de chercher à minimiser le carré moyen des écarts :

$$\frac{1}{n} \sum_{i=1}^n (t_i - u_{\alpha(i)})^2$$

Question 2.2 Montrez par un exemple l'avantage de cette fonction objective par rapport à la précédente. L'algorithme glouton est-il optimal pour la nouvelle fonction objective ?

Question 2.3 Montrer que si il y a autant de costumes que de musiciens, l'algorithme consistant à classer les musiciens et le costumes par taille croissante et à attribuer au musicien i le costume i est optimal pour la seconde fonction objective.

Question 2.4 Donner un algorithme donnant une solution optimale dans le cas $m \geq n$.

3 Sous-vecteur de somme maximale

Étant donné un tableau T de n entiers relatifs, on cherche $\max\{ \forall i, j \in \{1 \dots n\} \mid \sum_{k=i}^j T[k] \}$.

Par exemple pour le tableau suivant :

2	18	-22	20	8	-6	10	-24	13	3
---	----	-----	----	---	----	----	-----	----	---

l'algorithme retournerait la somme des éléments 4 à 7 soit 32.

Question 3.1 Donner un algorithme retournant la somme maximale d'éléments contigus par une approche *diviser pour régner*.

Question 3.2 Donner un algorithme retournant la somme maximale d'éléments contigus par *programmation dynamique*.

Question 3.3 Comparer la complexité Asymptotique au pire cas des deux approches.

4 La Bibliothèque

La bibliothèque planifie son déménagement. Elle comprend une collection de n livres b_1, b_2, \dots, b_n . Le livre b_i est de largeur w_i et de hauteur h_i . Les livres doivent être rangés dans l'ordre donné (par valeur de i croissante) sur des étagères identiques de largeur L .

Question 4.1 On suppose que tous les livres ont la même hauteur $h = h_i, 1 \leq i \leq n$. Montrer que l'algorithme glouton qui range les livres côte à côte tant que c'est possible minimise le nombre d'étagères utilisées.

Question 4.2 Maintenant les livres ont des hauteurs différentes, mais la hauteur entre les étagères peut se régler. Le critère à minimiser est alors l'encombrement, défini comme la somme des hauteurs du plus grand livre de chaque étagère utilisée.

4.2.1 Donner un exemple où l'algorithme glouton précédent n'est pas optimal.

4.2.2 Proposer un algorithme optimal pour résoudre le problème, et donner son coût.

Question 4.3 On revient au cas où tous les livres ont la même hauteur $h = h_i, 1 \leq i \leq n$. On veut désormais ranger les n livres sur k étagères de même longueur L à minimiser, où k est un paramètre du problème. Il s'agit donc de partitionner les n livres en k tranches, de telle sorte que la largeur de la plus large des k tranches soit la plus petite possible. Proposer un algorithme pour résoudre le problème, et donner son coût en fonction de n et k .

5 Bonus

On considère un ensemble S de $n \geq 2$ entiers distincts stockés dans un tableau (S n'est pas supposé trié). Résoudre les questions suivantes :

Question 5.1 Proposer un algorithme en $\mathcal{O}(n)$ pour trouver deux éléments x et y de S tels que $|x - y| \geq |u - v|$ pour tout $u, v \in S$.

Question 5.2 Proposer un algorithme en $\mathcal{O}(n \log n)$ pour trouver deux éléments x et y de S tels que $x \neq y$ et $|x - y| \leq |u - v|$ pour tout $u, v \in S, u \neq v$.

Question 5.3 Soit m un entier arbitraire (pas nécessairement dans S), proposer un algorithme en $\mathcal{O}(n \log n)$ pour déterminer s'il existe deux éléments x et y de S tels que $x + y = m$.

Question 5.4 Proposer un algorithme en $\mathcal{O}(n)$ pour trouver deux éléments x et y de S tels que $|x - y| \leq \frac{1}{n-1}(\max(S) - \min(S))$.