

## Algorithmique

### TD n° 10 : Algorithmes gloutons

#### Exercice 1 : le voyageur pressé

Un voyageur sur l'autoroute fait le plein d'essence seulement quand son réservoir ne contient plus assez d'essence pour atteindre la station-service suivante. Cette stratégie minimise-t-elle le nombre d'arrêts de notre voyageur ?

#### Exercice 2 : le Flexit

Les ministres de diverses nations européennes doivent se voir le 31 décembre 2024 pour préparer le Flexit qui aura lieu à 23h59. Or dans la grande effervescence engendrée par l'évènement, ils ne restent que quelques heures à Bruxelles, entre deux avions (ou TGVs). On ne peut pas faire une réunion avec tout le monde. Mais si beaucoup de réunions ont lieu, les conclusions risquent d'être divergentes et cela va être dur de faire la synthèse avant minuit... Il faut donc essayer de faire le moins de réunions possible (sachant qu'une réunion dure au moins une heure) et que tout ministre participe à au moins une réunion, pour exprimer la position de son pays.

*Donc* : vous avez  $n$  ministres ; le ministre  $M_i$  atterrit à l'heure  $a_i$  et repart à l'heure  $d_i$  ( $d_i - a_i > 1h$  sinon le problème est impossible). Une réunion à l'heure  $r_j$  peut réunir tous les ministres présents à ce moment-là pour une durée d'au moins une heure, donc tous ceux tels que  $a_i \leq r_j < r_j + 1h \leq d_i$ . Il faut proposer des heures  $r_1 \dots r_k$  de début des différentes réunion sachant que l'on veut  $k$  minimal.

1. Résoudre le problème sur l'exemple suivant (donner le nombre de réunions minimal, et l'heure des réunions).

Ministre	arrivée	départ	Ministre	arrivée	départ
BE	6h30	9h45	GR	9h01	16h30
DE	15h27	21h03	IT	15h42	19h17
DK	7h12	13h08	LU	13h33	18h22
ES	7h56	13h08	PL	13h33	20h07
FR	10h01	14h21	PT	19h07	23h02

2. Proposez un algorithme glouton qui résout le problème.  
Le choix glouton doit être fait après un certain *pré-traitement* des données, lequel ?
3. Montrez que votre algorithme minimise bien le nombre de réunions.

#### Exercice 3 :

On considère le processus suivant : À chaque instant on a un entier  $x$  initialement égal à 1. À chaque étape on peut l'incrémenter avec 1 ou le doubler. Le but est de produire une valeur cible  $n$ . Par exemple, à partir de 1 on peut obtenir 10 en quatre étapes :

$$1 \xrightarrow{+1} 2 \xrightarrow{*2} 4 \xrightarrow{+1} 5 \xrightarrow{*2} 10$$

Évidemment, on peut toujours obtenir  $n$  uniquement en incrémentant par 1, mais ce n'est pas très efficace.

1. Donnez un algorithme glouton pour obtenir le nombre *minimal* d'opérations permettant d'obtenir  $n$  à partir de 1.
2. Prouvez qu'il est correcte.

**Exercice 4 :**

Supposons qu'on a  $n$  skieurs avec leur tailles données par  $t_1, \dots, t_n$  et  $n$  paires de skis avec leur tailles données par  $s_1, \dots, s_n$ . Donner un algorithme qui assigne une paire de skis à chaque skieur de sorte que la différence moyenne entre la taille du skieur et la taille du ski soit minimal. Formellement, l'algorithme devrait donner une permutation  $\sigma$  de sorte que

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

soit minimal. Par exemple pour les tailles des skieurs 1.7, 1.9, 1.8 et les skis 2.1, 1.8, 1.6 la solution est la permutation  $\sigma$  avec  $\sigma(1) = 3$ ,  $\sigma(2) = 1$  et  $\sigma(3) = 2$ . Prouvez que votre algorithme est correcte.

**Exercice 5 :**

Un mot de parenthèses  $w$  est *équilibré* si et seulement si une des conditions suivantes est remplie :

- $w$  est le mot vide ;
- $w = (v)$ , où  $v$  est un mot équilibré ;
- $w = uv$ , où  $u, v$  sont des mots équilibrés.

Par exemple, le mot  $w = ((()))(())(())()$  est équilibré.

1. Proposez un algorithme pour déterminer si un mot de parenthèses est équilibré.
2. Proposez un algorithme glouton en temps linéaire pour déterminer un des plus long sous-mots équilibrés d'un mot de parenthèses. Prouvez-le et donnez la complexité. Par exemple, pour le mot  $)()()()$  ça peut être  $(())$  ou  $()()$ . Indication : Adaptez l'algorithme pour 1.