# ITC MPSI
# Notes de cours algorithmes gloutons

Ressources :

https://eduscol.education.fr/document/30067/download https://clogique.fr/nsi/premiere/gloutons/

## I) Introduction

Les algorithmes *gloutons* sont des algorithmes souvent simples, qui donnent une solution pas forcément exacte à un problème. C'est un peu "la solution évidente qui en fait ne fonctionne pas" dans la plupart des cas.

Pour certaines applications, on va se satisfaire d'un tel algorithmes.

On va présenter deux exemples d'algorithmes gloutons.

## II) Problème du rendu de monnaie

### II.1) Présentation

*Exemple 1*

> Vous souhaitez rendre (ou payer) 34€ pour un achat. Quelles pièces et billets utilisez-vous ? Attention, on demande d'utiliser le moins de coupures possibles !
>
> Instinctivement, quelle méthode utilisez-vous ? Essayez de la transposer en un algorithme.

Idée : toujours rendre la plus grande pièce possible d'abord. C'est un algorithme "glouton", car il essaye de prendre "le plus possible" à chaque moment, sans anticiper la suite ni revenir en arrière.

*Problème 2* [du rendu de monnaie]

> On a $n$ pièces $v_1 < v_2 < ... < v_n \in \mathbb{N}$, et une somme à rendre $S \in \mathbb{N}$. On pose $v_1 = 1$.
>
> On cherche un n-uplet $T = (x_1, x_2, ..., x_n)$ tel que $S = \sum_{i=1}^{n} x_i v_i$, et qui minimise $\sum_{i=1}^{n} x_i$.

On cherche à écrire une fonction Python qui prend en argument
- une liste d'entiers `[v1, ..., vn]`
- un entier S

Et qui renvoie la liste d'entiers `[x1, ..., xn]` obtenue par la méthode gloutonne.

lien vers un notebook contenant le code

### II.2) Non optimalité

Sur notre exemple des €, il se trouve que l'algorithme glouton donne toujours la solution optimale. On dit que c'est un système monétaire *canonique*. Mais ce n'est pas toujours le cas !

*Exemple 3*

> On prend le système monétaire qui a des pièces de 1, 3, et 4. Si on essaie de rendre 6, que donne l'algorithme glouton ? Est-ce optimal ?

L'exemple a peut-être l'air un petit peu "construit", mais des systèmes non canoniques ont bien été en circulation dans l'histoire.

*Exemple 4*

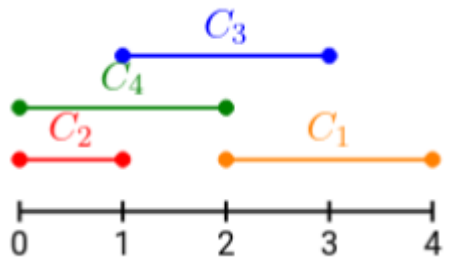
Avant 1971, le système britannique avait des pièces de [1, 2, 6, 12, 24, 30, 60, 240]. Pouvez-vous trouver un exemple où le glouton n'est pas optimal ?

Réponse : 48

On ne connait de condition nécessaire et suffisante pour qu'un système soit canonique.

# III) Construction d'emploi du temps

Vous allez dans une convention / séminaire / etc, où un certain nombre d'événements et de conférences se produisent. Vous souhaitez assister a un maximum de conférences.




Par exemple, ici, on peut assister à 2 conférences maximum.

Comment choisir à quelles conférences assister ?

On peut imaginer plusieurs algorithmes gloutons différents dans ce cas ! Dans tous les cas, il s'agira de trier les conférences selon un certain critère, et de prendre la "meilleure" conférence selon ce critère de manière répétitive.

## III.1) Première solution : on les prend dans l'ordre de *début*.

Contre exemple : une très longue conférence dés le début, puis plein de petites qui lui sont superposées :

```
|---------------------------------------| C1
   |--| C2     |--| C3    |--| C4
```

## III.2) Deuxième solution : on prend les plus courtes d'abord.

A priori, si une conférence est courte, elle doit être plus facile à caser dans un emploi du temps. Mais ce n'est pas optimal non plus !

```
|---------------| C1 |---------------------| C3
                |-------| C2
```

Ici, C2 est la plus courte.

## III.3) Celles qui interfèrent avec le moins d'autre conférences

A priori, si une conférence a peu de conflits, elle est plus intéressante.

```
          |----------C1------------|
|--| |---C2--|                    |---C3---| |--|
  |---|                                 |---|
  |---|                                 |---|
  |---|                                 |---|
```

Ici, C1 a le moins de conflit. Mais elle empêche de choisir C2 et C3, ce qui nous mène a un choix avec 3 conférences (alors qu'on peut faire 4 ! )

## III.4) Troisième solution : on les prend dans l'ordre de *fin*.

Essayez de trouver un contre exemple. Compliqué, non ?

Et bien en fait, c'est optimal !

Pourquoi ?

Soit $S$ l'ensemble des conférences. Considérons une allocation optimal $C_1^{\text{opt}}, ..., C_n^{\text{opt}}$. On considère les $C_1, ..., C_n$ triés temporellement. (Comme ils ne peuvent pas s'intersecter, ce n'est pas important si on trie par heure de début ou de fin.)

Soit $C_1^g$ la première activité choisie par l'algorithme glouton (soit celle qui termine en dernier). Alors, $C_1^g, C_2^{\text{opt}}, ..., C_n^{\text{opt}}$ reste une allocation valide, et donc elle est aussi optimale.

On peut aussi noter que $C_2^{\text{opt}}, ..., C_n^{\text{opt}}$ est une allocation optimale pour les conférences $S \setminus \left\{\text{conférences intersectant avec } C_1^{\text{opt}}\right\}$. On peut donc appliquer le même argument n fois, et $C_g^1, ..., C_g^n$ est une allocation valide, et donc optimale.