

# ALGORITHMES GLOUTONS : COURS

## Définition

Les algorithmes gloutons constituent une grande famille d'algorithmes.

- On peut les utiliser lorsqu'on a une sélection à effectuer sur un ensemble d'objets en cherchant à maximiser ou minimiser une certaine grandeur tout en respectant certaines contraintes.
- La sélection est effectuée en appliquant à chaque étape une règle de choix définie à l'avance qui effectue ce qui semble être le meilleur choix sur le moment en espérant arriver à la fin à la solution optimale.

## Exemple 1

On cherche à sélectionner cinq nombres de la liste suivante (sélection) en cherchant à avoir leur somme la plus grande possible (maximiser une grandeur) et en s'interdisant de choisir deux nombres voisins (contrainte).

15	4	20	17	11	8	11	16	7	14	2	7	5	17	19	18	4	5	13	8
----	---	----	----	----	---	----	----	---	----	---	---	---	----	----	----	---	---	----	---

Règle de choix de l'algorithme glouton :

À chacune des cinq étapes, choisir le plus grand nombre possible dans les choix restants.

[Basée sur l'idée que plus on choisit un grand nombre à chaque étape, plus on aura un grand résultat à la fin.]

*Déroulé de l'algorithme glouton :*

Étape 1 : on choisit le 20 en l'entourant et on raye le 4 et le 17 voisins (à cause de la contrainte).

Étape 2 : on choisit le 19 en l'entourant et on raye le 17 et le 18 voisins (à cause de la contrainte).

Étape 3 : on choisit le 16 en l'entourant et on raye le 11 et le 7 voisins (à cause de la contrainte).

Étape 4 : on choisit le 15 en l'entourant et on ne raye rien du tout (le 4 est déjà rayé depuis l'étape 1).

Étape 5 : on choisit le 14 en l'entourant et on raye le 2 voisin (le 7 est déjà rayé depuis l'étape 3).

*Maximisation gloutonne obtenue :*

$$20+19+16+15+14 = 84$$

*Sur cet exemple, est-ce que l'algorithme glouton est un "bon" algorithme ?*

Argument pour répondre "Non" : on peut trouver une meilleure solution (20, 18, 17, 16 et 15 qui conduisent à un total de 86)

Argument pour répondre "Oui" : sur de longues listes, les algorithmes permettant de trouver la meilleure solution (on dit *optimale*) auraient une complexité beaucoup plus grande que notre algorithme glouton. De plus, on peut démontrer que notre algorithme glouton fournit dans tous les cas une solution "*presque optimale*".

## Exemple 2

On dispose du planning d'un festival culturel qui propose des spectacles sur cinq scènes différentes.

Un festivalier, qui arrive à 10:00, cherche à choisir des spectacles dans ce planning (sélection) en cherchant à voir le plus grand nombre de spectacles possibles dans sa journée (maximiser une quantité) et en s'imposant de voir chaque spectacle en entier (contrainte).

Il peut envisager deux algorithmes gloutons différents.

Règle de choix de l'algorithme glouton A :

À chaque étape choisir, parmi les cinq scènes, le prochain spectacle qui *commence* en premier.

[Basée sur l'idée que moins on attend entre deux spectacles, plus on verra de spectacles.]

Règle de choix de l'algorithme glouton B :

À chaque étape choisir, parmi les cinq scènes, le prochain spectacle qui *fin*it en premier.

[Basée sur l'idée que plus un spectacle finit tôt, plus il y aura de la place pour les spectacles suivants.]

### Déroulé de l'algorithme

#### glouton A :

Étape 1 : spectacle 3A  
 Étape 2 : spectacle 2B  
 Étape 3 : spectacle 3B  
 Étape 4 : spectacle 2C  
 Étape 5 : spectacle 3D  
 Étape 6 : spectacle 2D  
 Étape 7 : spectacle 4D  
 Étape 8 : spectacle 3F  
 Étape 9 : spectacle 3G  
 Étape 10 : spectacle 2G

### Déroulé de l'algorithme

#### glouton B :

Étape 1 : spectacle 4A  
 Étape 2 : spectacle 2A  
 Étape 3 : spectacle 4B  
 Étape 4 : spectacle 3B  
 Étape 5 : spectacle 3C ou 2C  
 Étape 6 : spectacle 3D  
 Étape 7 : spectacle 5D  
 Étape 8 : spectacle 5E  
 Étape 9 : spectacle 5F  
 Étape 10 : spectacle 3F  
 Étape 10 : spectacle 3F  
 Étape 11 : spectacle 3G  
 Étape 12 : spectacle 2G

L'algorithme glouton A aboutit à une moins bonne solution que l'algorithme glouton B (10 spectacles contre 12 spectacles). En effet, le A choisit parfois de très longs spectacles (comme le 3A) ce qui est contradictoire avec le fait d'en voir beaucoup.

	SCENE 1	SCENE 2	SCENE 3	SCENE 4	SCENE 5	
10:00						10:00
11:00	spectacle 1A		spectacle 3A	spectacle 4A	spectacle 5A	11:00
12:00		spectacle 2A				12:00
13:00		spectacle 2B		spectacle 4B		13:00
14:00					spectacle 5B	14:00
15:00	spectacle 1B		spectacle 3B			15:00
16:00		spectacle 2C	spectacle 3C	spectacle 4C	spectacle 5C	16:00
17:00		spectacle 2D	spectacle 3D		spectacle 5D	17:00
18:00					spectacle 5E	18:00
19:00			spectacle 3E		spectacle 5F	19:00
20:00	spectacle 1C	spectacle 2E		spectacle 4D		20:00
21:00						21:00
22:00			spectacle 3F		spectacle 5G	22:00
23:00	spectacle 1D	spectacle 2F		spectacle 4E		23:00
0:00						0:00
1:00			spectacle 3G		spectacle 5H	1:00
2:00	spectacle 1E	spectacle 2G				2:00
3:00						3:00

*Remarque :* On peut démontrer que l'algorithme glouton B fournit systématiquement une solution optimale (ainsi, sur cet exemple, on ne peut pas trouver de solution permettant de voir plus de 12 spectacles en entier).

### Conclusion

**Les algorithmes gloutons s'utilisent lorsqu'on cherche à effectuer une sélection en optimisant une quantité et en respectant certaines contraintes. Ils se basent sur une règle de choix définie à l'avance qui effectue ce qui *semble* être le meilleur choix sur le moment.**

Dans certaines situations comme l'algo B de l'exemple 2, ils permettent de trouver systématiquement la meilleure solution. Dans d'autres situations comme l'exemple 1, sans trouver la meilleure solution, ils permettent toutefois de trouver une "bonne" solution sans effectuer "beaucoup de calculs".