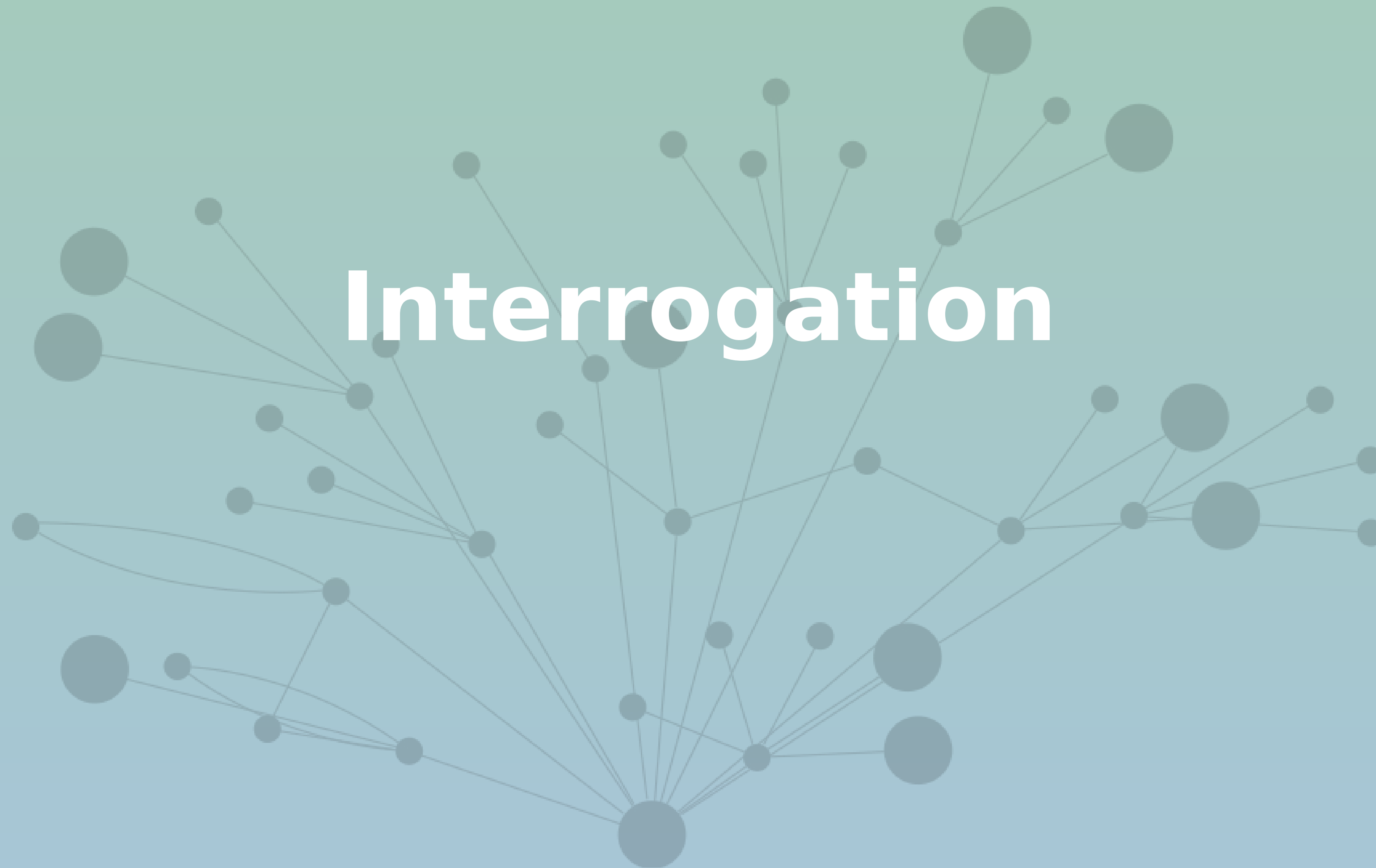


# Algo et prog

Benoit Simard



# Interrogation



# Agenda

- 5 exercices
- 10 min par exercice
- **~50 minutes**

# Enoncé

- Ecrire un algo permettant à un utilisateur de saisir les notes d'une classe (la taille d'une classe peut varier ^^), et afficher la moyenne des notes.
- Demander à l'utilisateur de saisir 5 nombres, et afficher tous les nombres supérieurs à la moyenne.
- Ecrire l'algo qui permet d'obtenir le résultat suivant : 1, 12, 123, ..., 1234567

# Enoncé 2

Que fait le code suivant

---

# Enoncé 3

Que fait l'algo ? Bref, c'est quoi A et B ?

---



# Exercice sur les tableaux

# Exercice tab 1

Etant donné deux tableaux, trouver leurs éléments communs

Pour mémoire `int[] tab = { 1, 2, 3 , 4}`



# Exercice tab 2

Idem, mais permettre a l'utilisateur de saisir les tableaux

# Exercice tab 3

Etant donné deux tableaux, créer un tableau contenant tous les éléments unique des tableaux (sans répétition).

$$[1, 2, 3, 4] + [3, 4, 5] = [1, 2, 3, 4, 5]$$

# Exercice tab 4

Ecrire un algo qui permet de donner le nombre median d'un tableau (de taille impaire).

# Exercice tab 5

Ecrire un algo qui permet de trier un tableau par order croissant.

# Complexité



# Définition

C'est le cout d'un algorithme, c'est a dire une mesure de sa rapidité d'exécution.  
Cela permet de comparer la performance de deux algos résolvant le même problème.

# Exemple 1

Cout de  $2 \times n$  additions

---

# Exemple 2

Cout de  $n^2$  opérations

---



# Exemple 2 bis

Cout de n opération

---

# Evaluation

Pour évaluer la complexité d'un algo, il faut déterminer le nombre d'opérations effectué en fonction de la taille  $n$  des entrées

Il n'existe pas un ensemble standardisé pour évaluer la complexité d'un algo

# La notation

- $O(1)$  : temps constant
- $O(\log(n))$  : temps proportionnel mais logarithmique a la taille de l'entrée
- $O(n)$  : temps proportionnel a la taille de l'entrée
- $O(n^2)$  : temps proportionnel mais exponentiel a la taille de l'entrée

# Les algos de tri de tableaux

# Principe

Etant donné un tableau de taille, trier ses éléments par ordre croissant.

# tri sort

Il consiste en effet à parcourir 'bêtement' le tableau à la recherche du plus petit élément. Lorsque celui-ci est trouvé, il est placé en première position. A nouveau, l'algorithme recherche le plus petit élément dans le reste du tableau et le place en seconde position. Cette procédure est répétée jusqu'à ce que le tableau soit trié.

# Exemple

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7

# Spseudo-code

---



# Pour le fun

**<https://www.youtube.com/watch?v=Ns4TPTC8whw>**

# Bubble sort

Le principe du tri à bulles (bubble sort ou sinking sort) est de comparer deux à deux les éléments  $e_1$  et  $e_2$  consécutifs d'un tableau et d'effectuer une permutation si  $e_1 > e_2$ . On continue de trier jusqu'à ce qu'il n'y ait plus de permutation.

# Exemple



# Complexité

Dans le pire des cas  **$O(n^2)$**

# Spseudo-code

---

# Spseudo-code optimisé

---

# Pour le fun

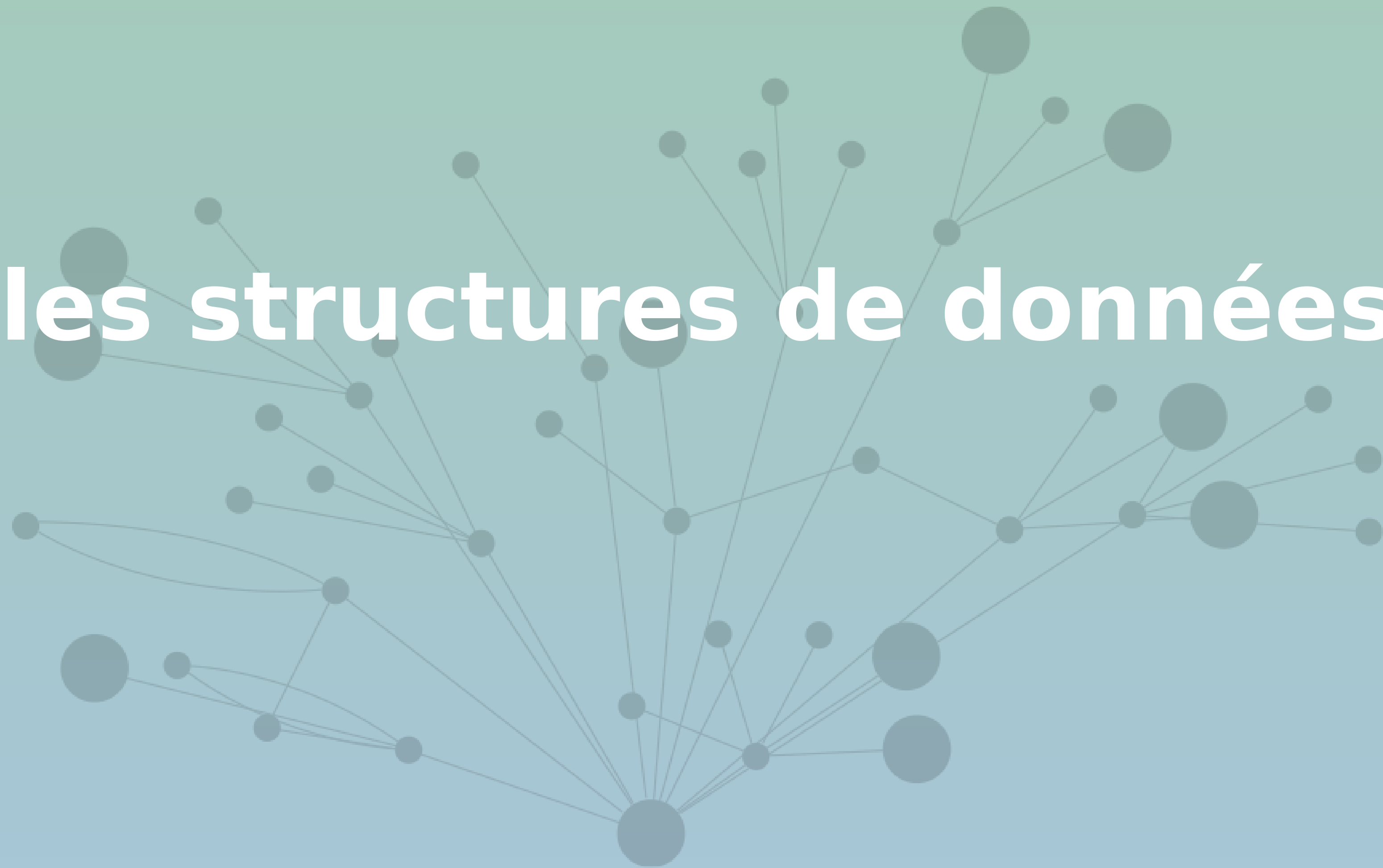
**<https://www.youtube.com/watch?v=lyZQPjUT5B4>**

# Bien d'autres algo ...

**<https://www.youtube.com/watch?v=BeoCbJPuvSE>**



# les structures de données



# Les tableaux

Déjà vu !! D'autres exo : [\*\*https://fr.scribd.com/doc/214641203/algorithme-exercices-corriges\*\*](https://fr.scribd.com/doc/214641203/algorithme-exercices-corriges)

# La File

**FIFO** : First In First Out

# La pile

**LIFO** : Last In First Out

# Les arbres

Haa