

**Bienvenue au cours  
d'algorithmes et  
structures de données !**

HE<sup>VD</sup>  
IG



Quel est ce lieu?



**Khwarezm (Khiva), en Ouzbékistan**  
**Pourquoi vous en parler ?**

HE VD  
IG



# *MONGOL CONQUEST: 1206 TO 1260*

- Mongol Territories 1206
- Chingis Khan's conquests 1209-27
- Mongol conquests 1227-59
- Indirect Mongol Control
- Border 1200

- Campaigns under Ghenghis
- Mongol campaigns, 1228-60
- ⊗ Mongol victory
- ⊗ Mongol defeat
- 🔥 City sacked by the Mongols

an	<i>ONGUTS</i>	Mongol tribe united by Chingis Khan
	<i>XIXIA</i>	State
	<i>Turks</i>	Nomadic Group



# البیرونی

Abū Rayhān Muḥammad ibn Aḥmad  
**Al-Bīrūnī** (973-1048)

- Erudit khorezmien, mathématicien, astronome, physicien, encyclopédiste, philosophe, astrologue, voyageur, historien, pharmacologue et précepteur
- Connu pour avoir étudié la thèse de la rotation de la Terre autour de son axe et sa révolution autour du Soleil.
- Calcula le rayon de la Terre à 6 339,6 km, à seulement 0.5% de la valeur exacte.



# الخوارزمي

Muhammad Ibn Mūsā al-Khuwārizmī  
(780-850)

- Mathématicien, géographe, astrologue et astronome perse, membre de la Maison de la sagesse de Bagdad.
- Ses écrits en arabe, traduits en latin à partir du XIIe siècle, ont permis l'introduction de l'algèbre en Europe.



الكتاب المختصر في  
حساب الجبر والمقابلة

Kitāb al-mukhtaṣar fī ḥisāb al-jabr  
wa-l-muqābala

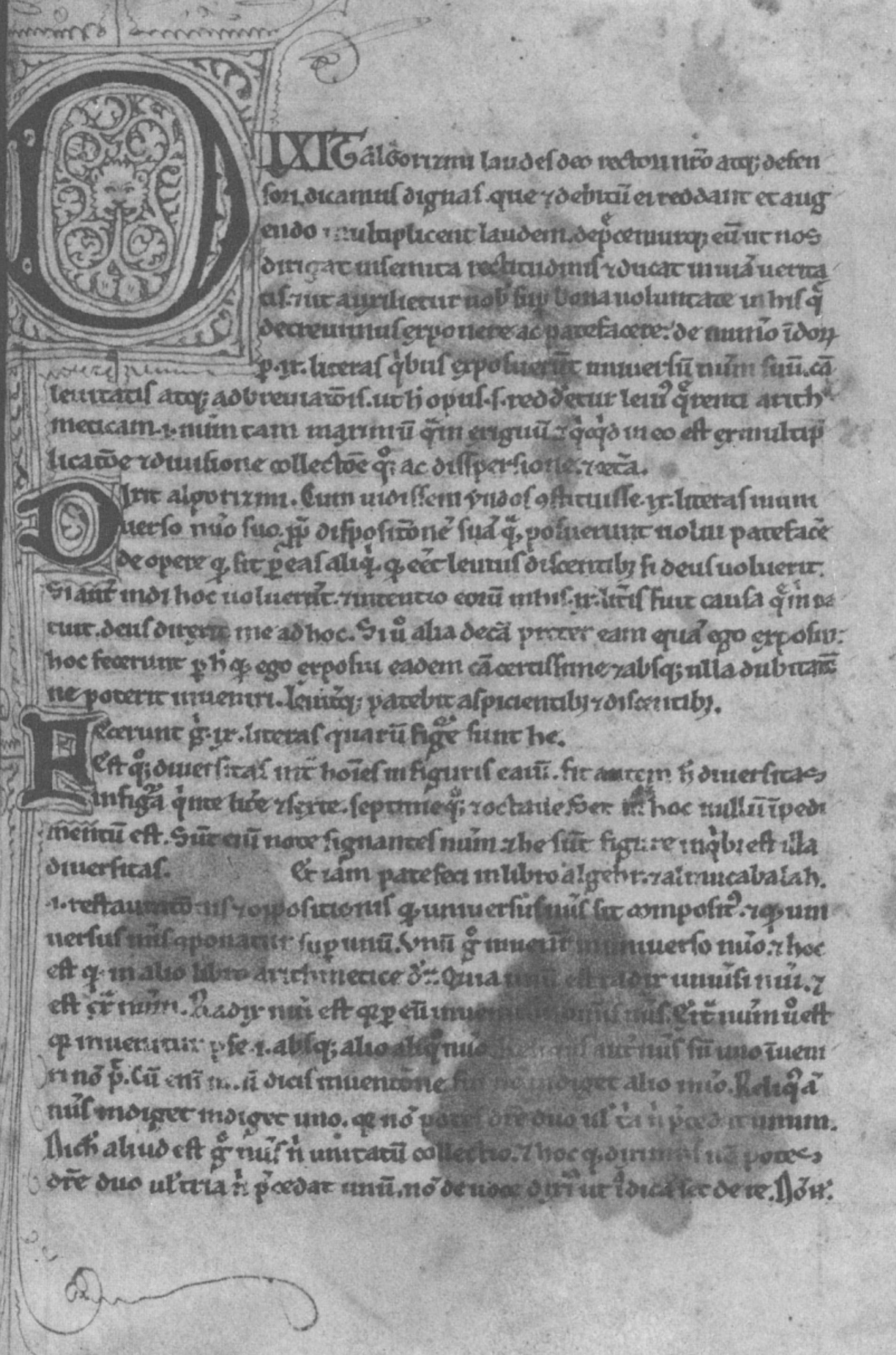
- « Abrégé du calcul par la restauration et la comparaison »
- Fondations de l'algèbre, étude des équations du 1<sup>er</sup> et 2<sup>ème</sup> degré
- al-jabr donne algebra en latin, puis algèbre en français



# Dixit Algorizmi

kitāb al-ḥisāb al-hindī

- « Traité de calcul avec les chiffres indiens », dont il ne reste pas de version originale en arabe
- La traduction latine commence par « dixit Algorizmi », qui latinise le nom **الخوارزمي**.
- Origine étymologique du mot français « **algorithme** »



# Algorithmes et structures de données

Olivier Cuisenaire

Laura Raileanu

Olivier Lemer

Olivier Tischhauser



الخوارزمي



“ I will, in fact, claim that the **difference between a bad programmer and a good one** is whether he considers his code or his data structures more important. Bad programmers worry about the **code**. Good programmers worry about **data structures** and their relationships. ”

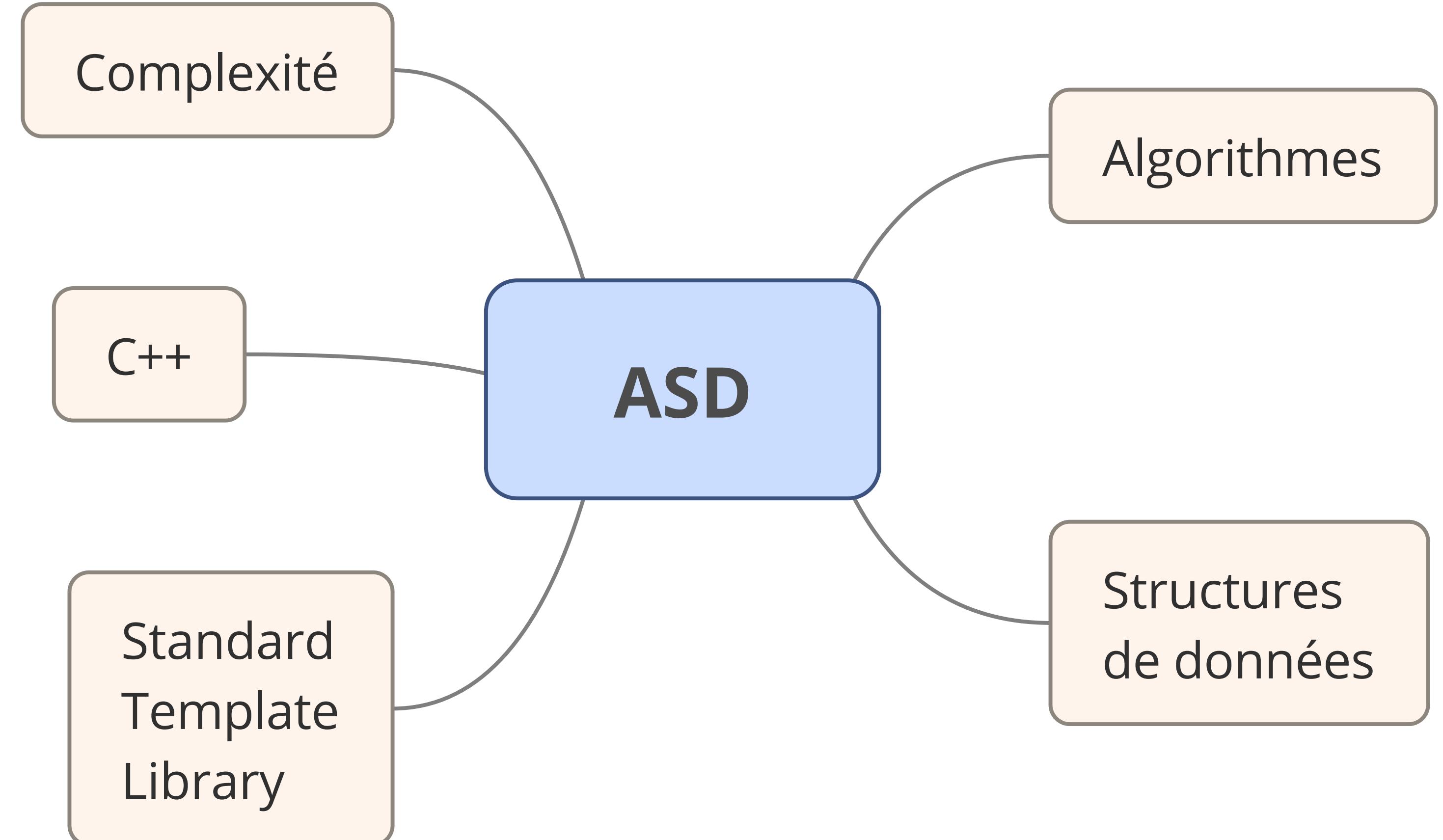
*Linus Torvalds*

# Au menu d'ASD...

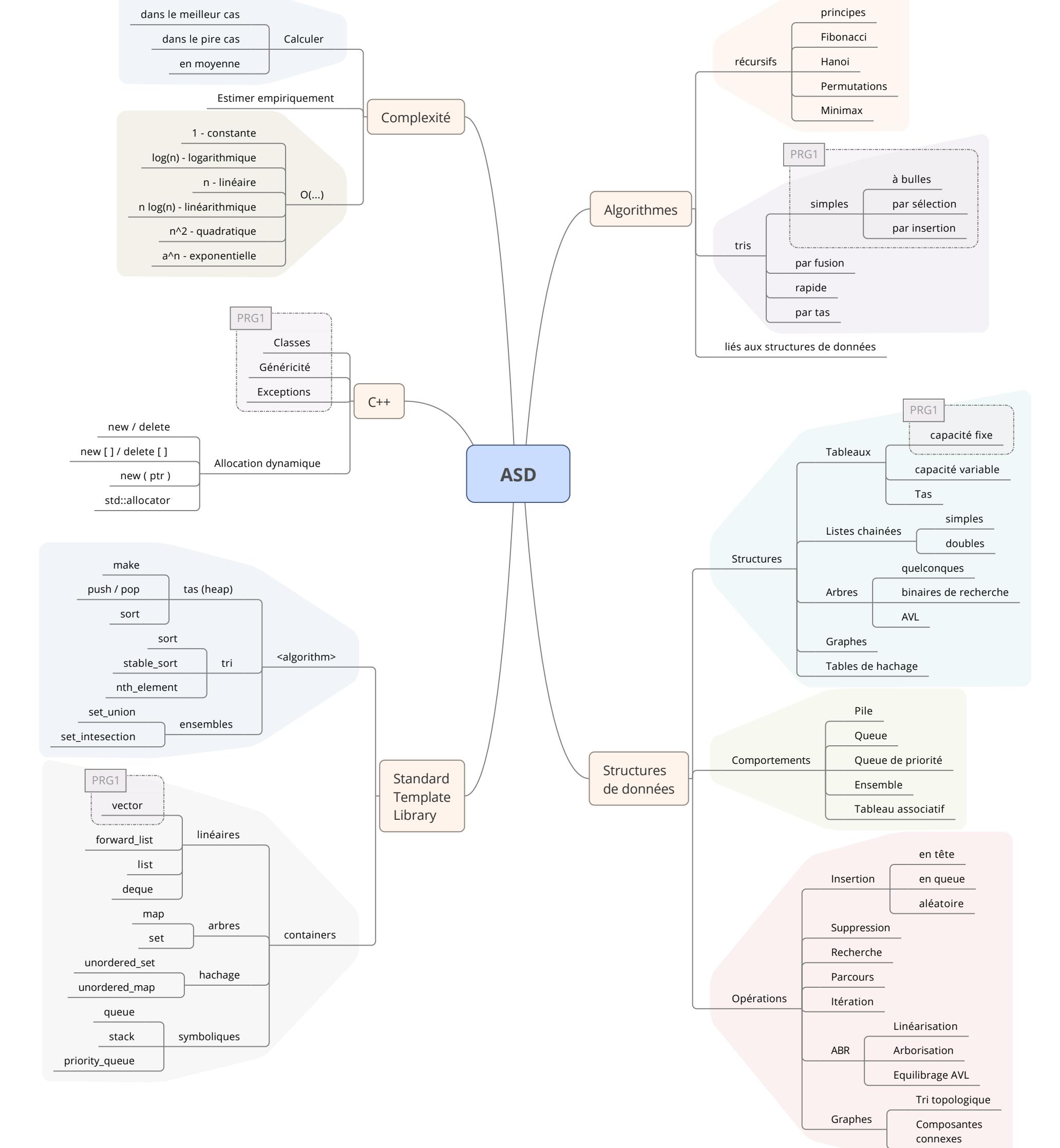


الخوارزمي

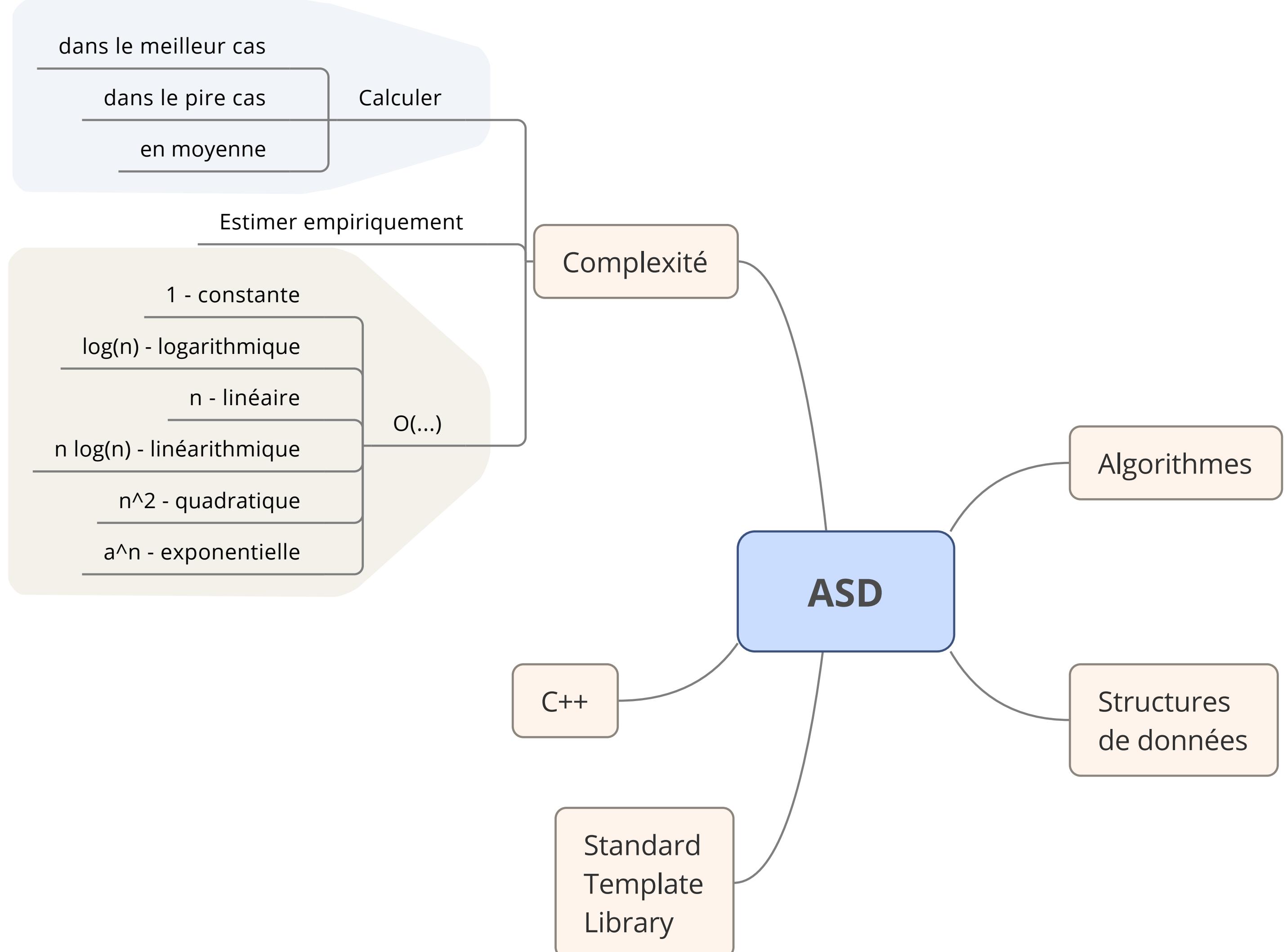
# Thèmes principaux



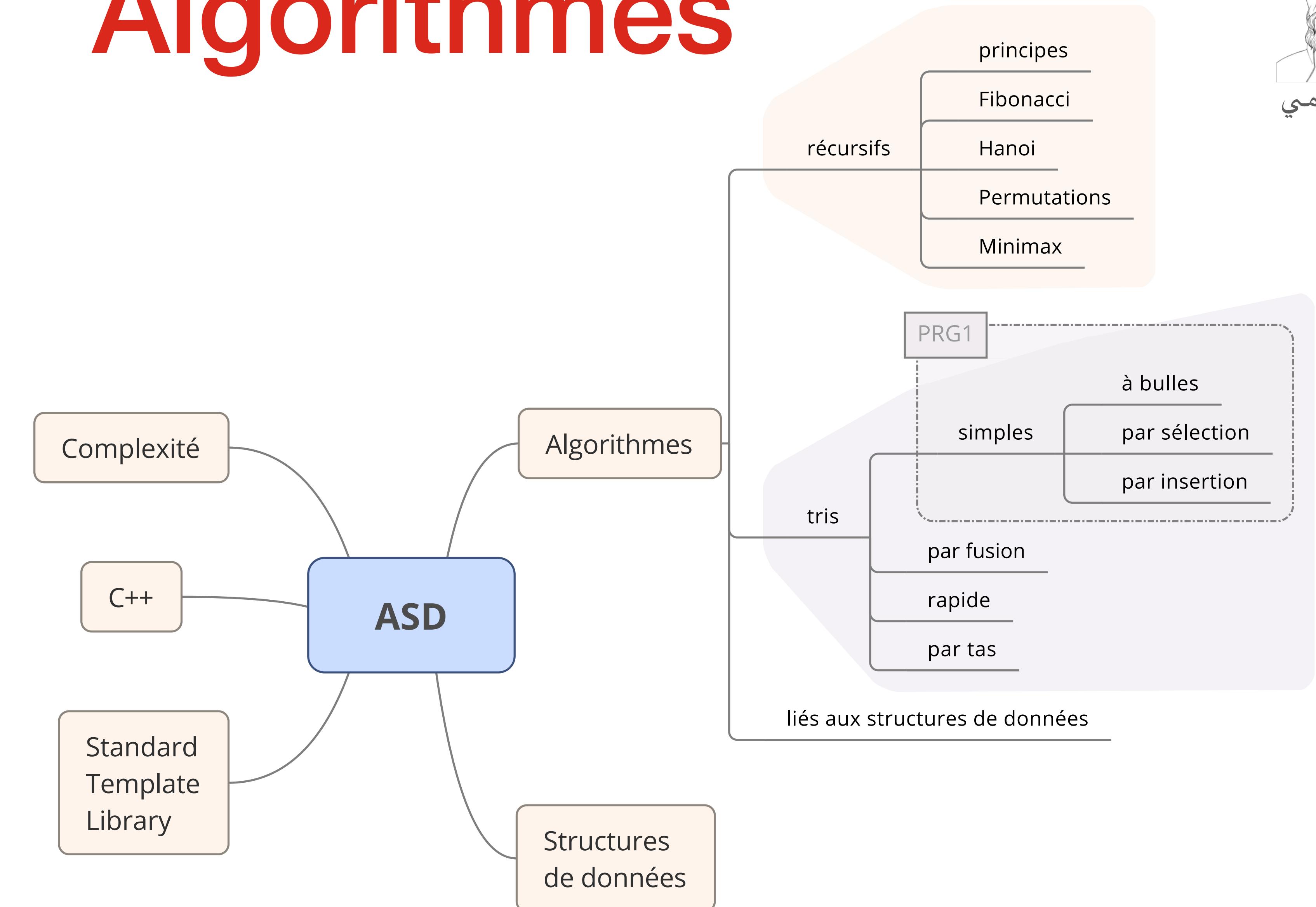
# Avec plus de détails...



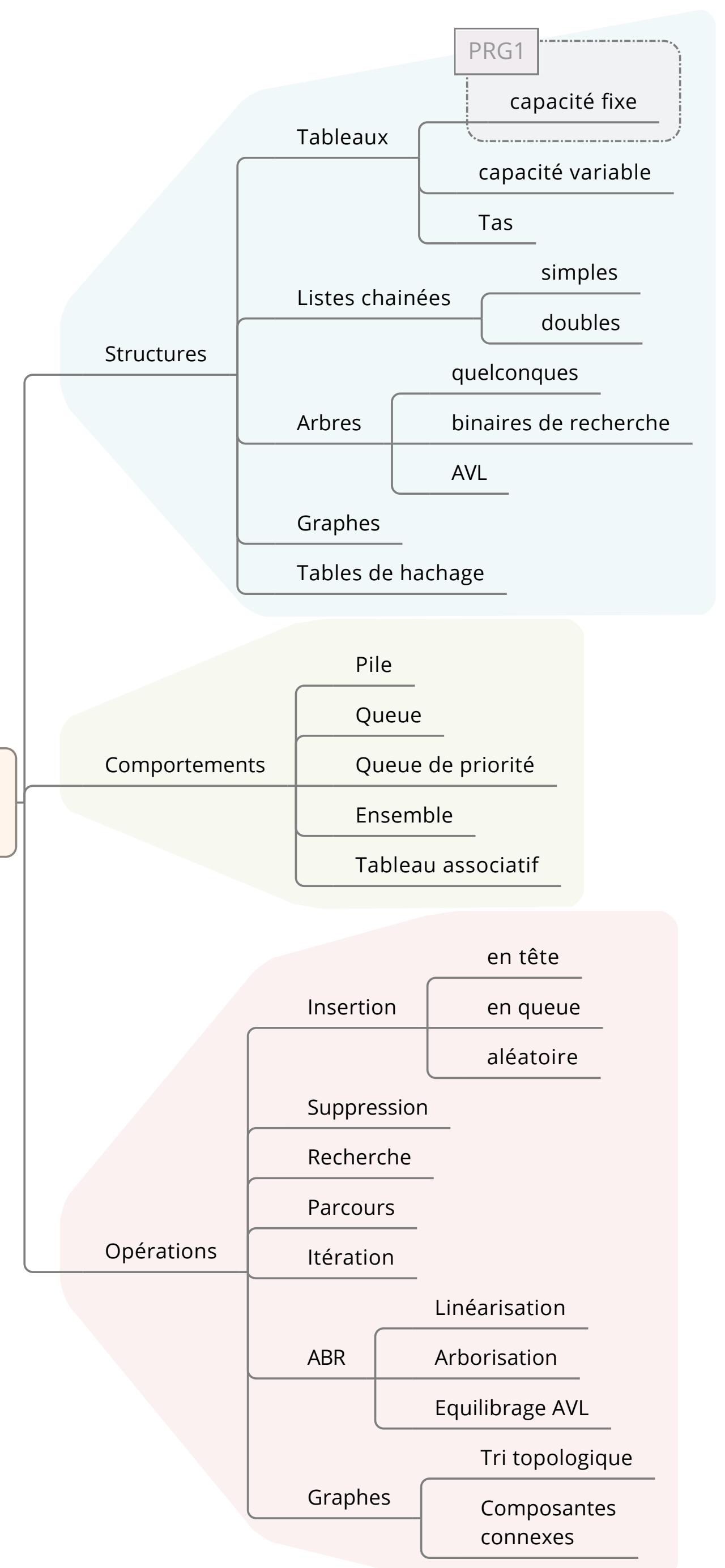
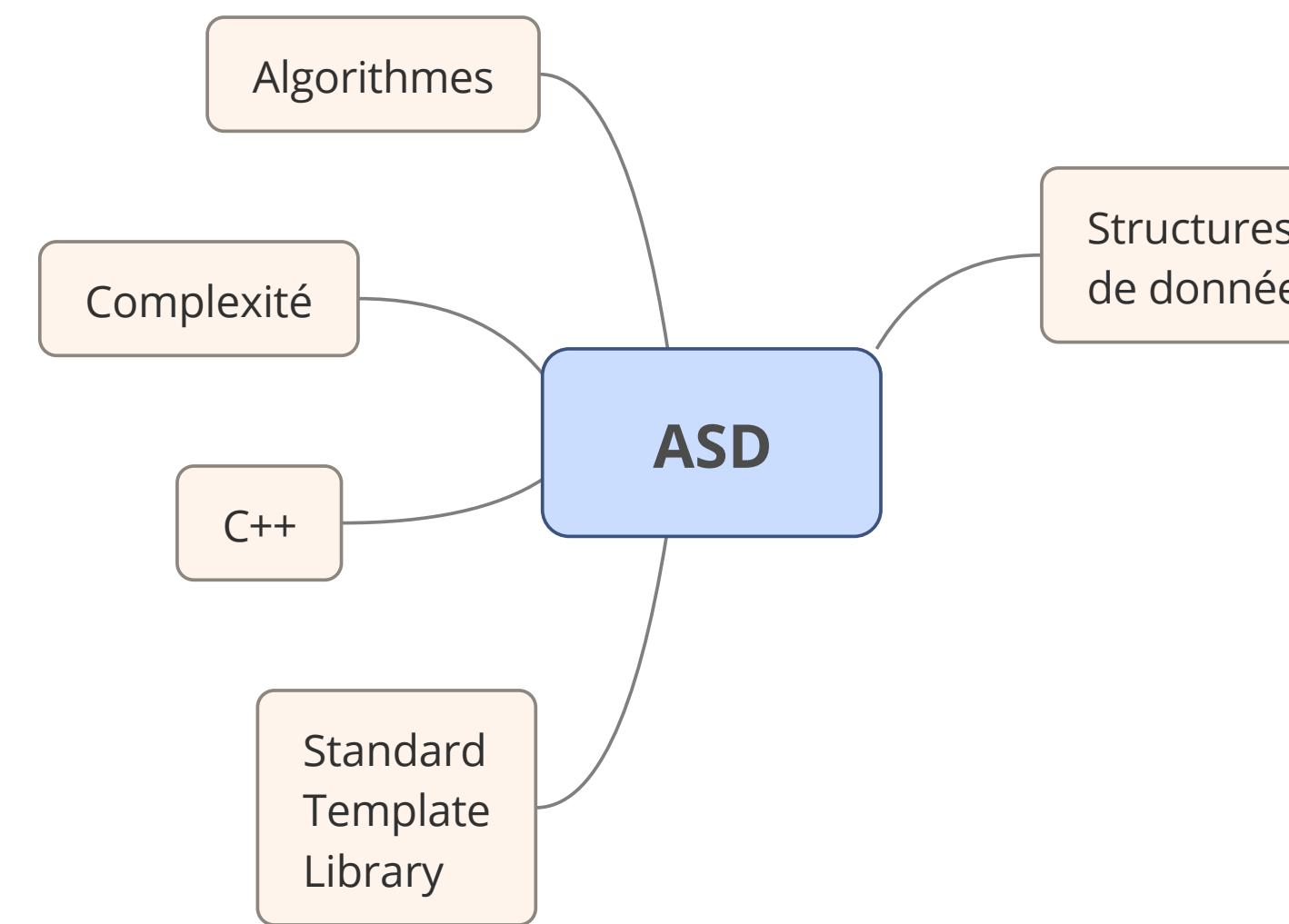
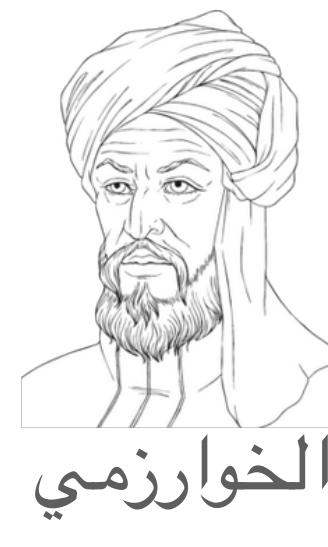
# Complexité



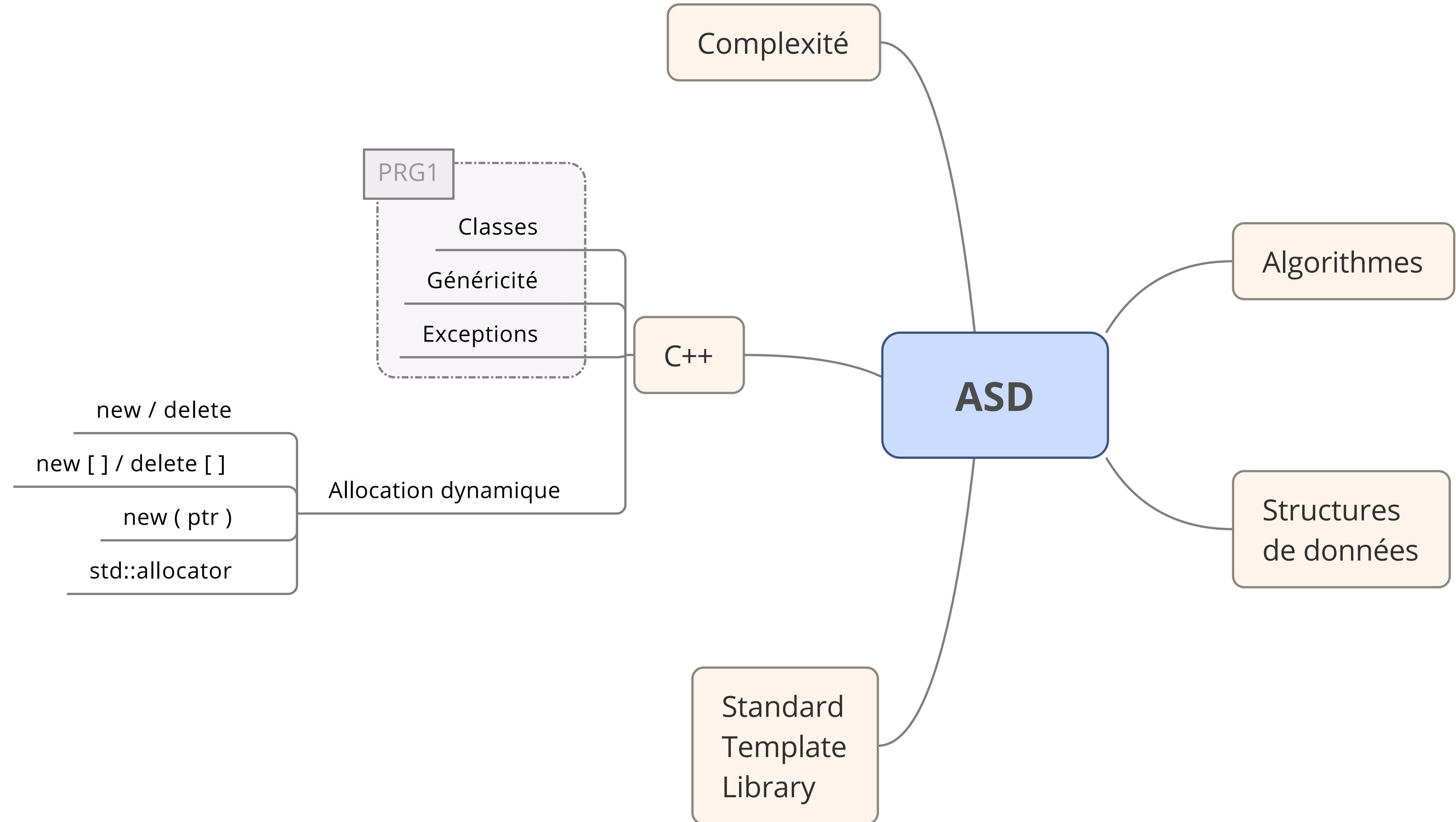
# Algorithmes



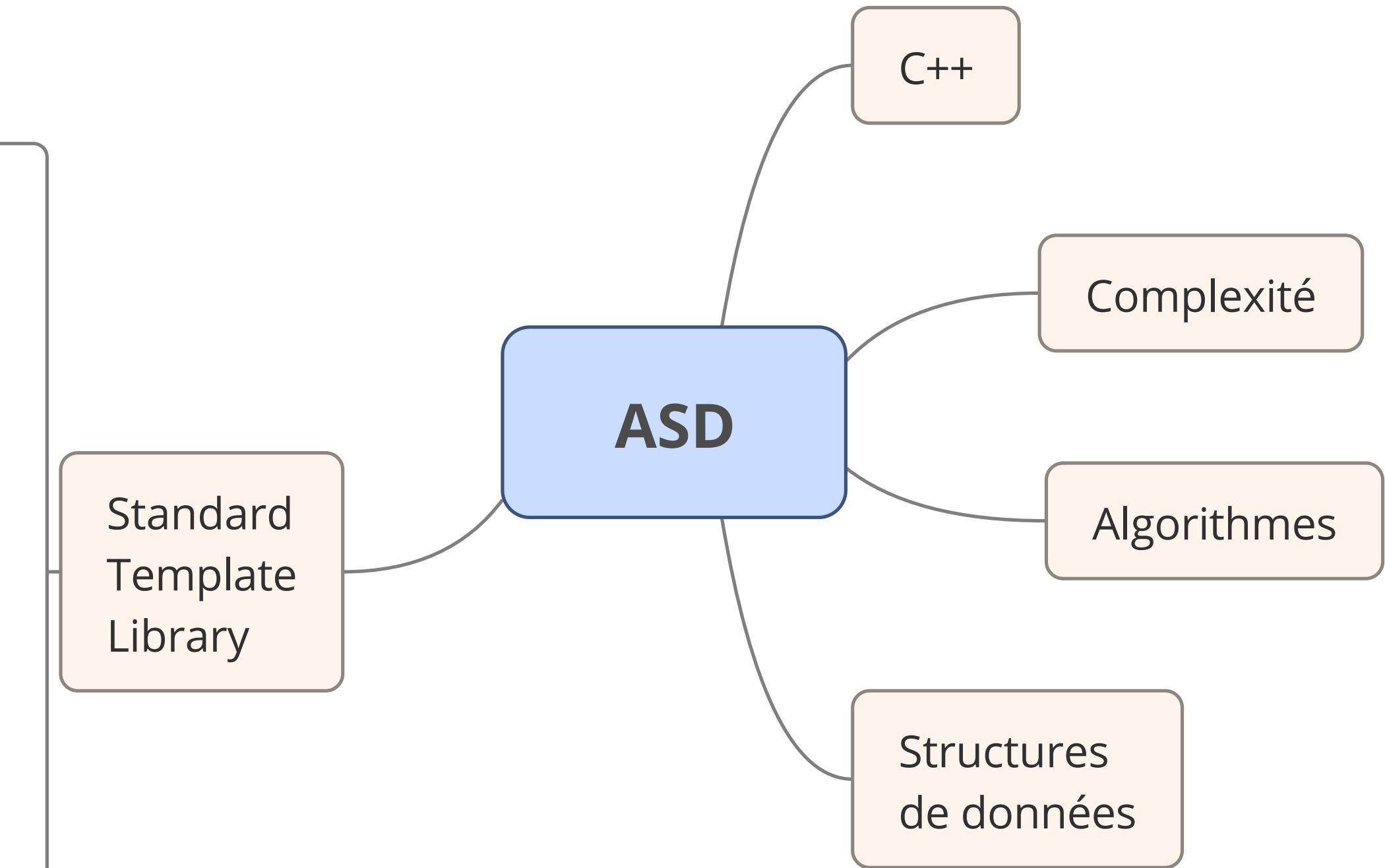
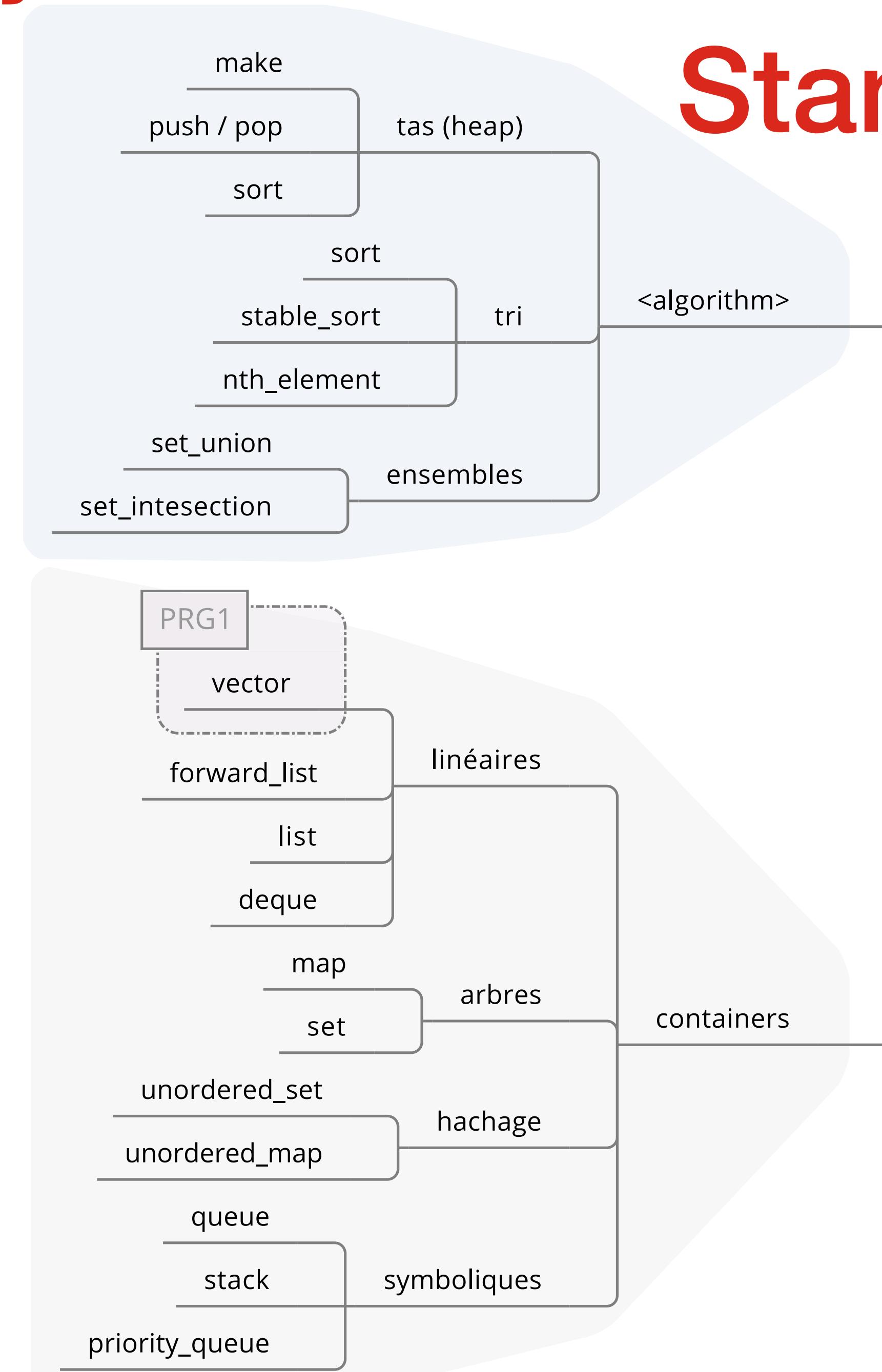
# Structures de données



# C++

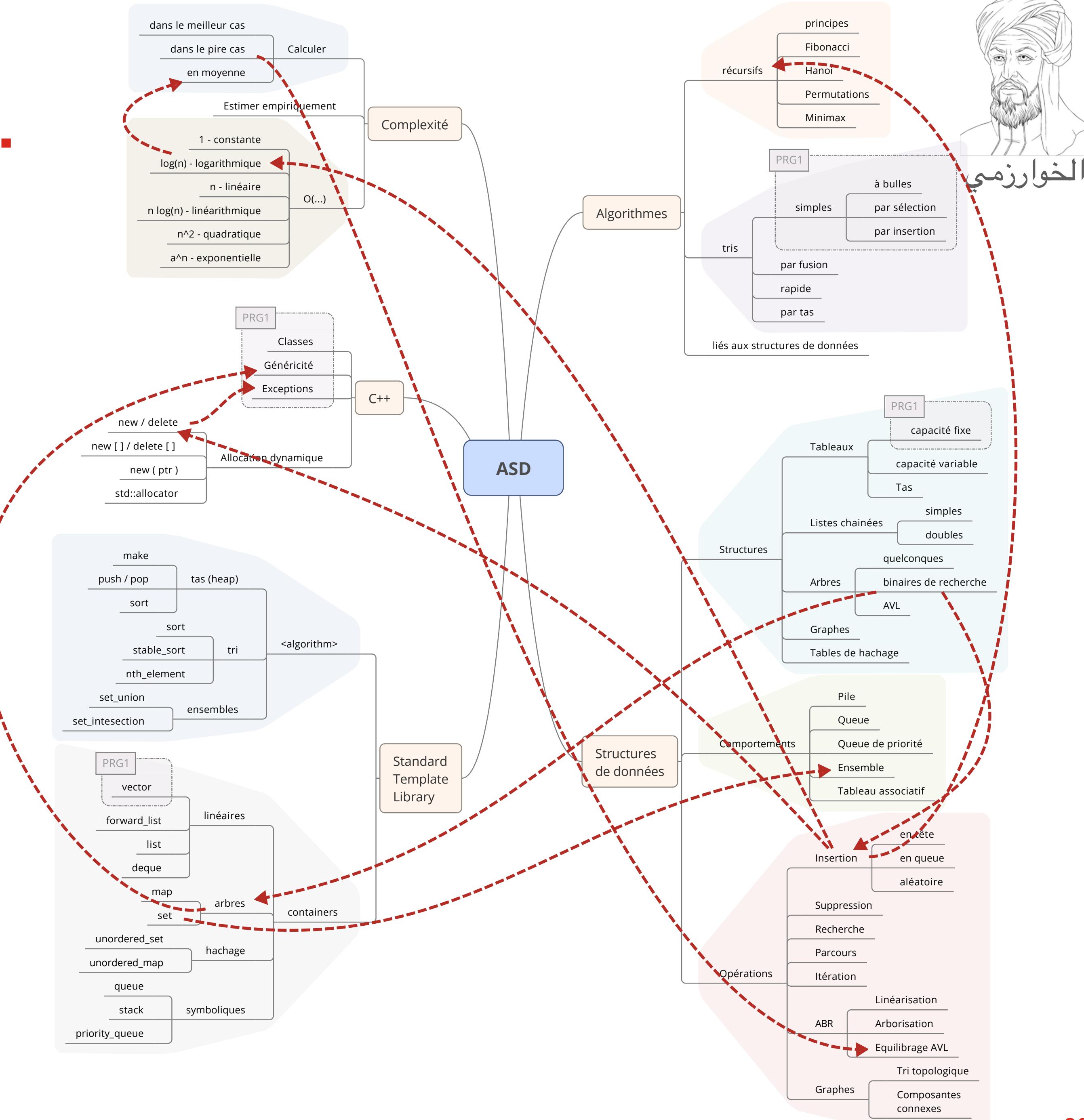


# Standard Template Library



# Insérer dans un arbre binaire de recherche ...

- Algorithme récursif
- De complexité logarithmique ... en moyenne
- Aussi dans le pire des cas si on équilibre l'arbre, par exemple avec un arbre **AVL**
- Pour le mettre en oeuvre en C++, il faut allouer dynamiquement avec `new / delete`, ce qui peut lever une **exception std::bad\_alloc**
- Cette structure est utilisée par `std::set` et `std::map`, des mises en oeuvres génériques des TDA **ensemble** et **tableau associatif**



# Table des matières



- **Introduction** : algorithmes, complexités, TDA, mise en oeuvre en C++
- **Récursivité** : Euclide, factorielle, Fibonacci, Hanoi, permutation, minimax
- **Tris** : stabilité, complexités, tri fusion, tri rapide, sélection rapide, tri optimal, tri comptage, tri par base
- **Structures linéaires** : tableaux de taille et/ou capacité variable, buffers circulaires, listes simplement ou doublement chainées, itération, tri de listes, tas, tri par tas, TDAs pile, file, file de priorité
- **Arbres** : représentation, parcours et itération sur arbres généraux, arbres binaires, binaires de recherche, AVL, TDAs ensemble et tableau associatif
- **Graphes** : représentations, parcours, composantes connexes, tri topologique, algorithmes de Kosaraju, de Dijkstra et de Kruskal
- **Tables de hachage** : fonction de hachage, collisions, chainage, adressage ouvert, redimensionnement

# En pratique



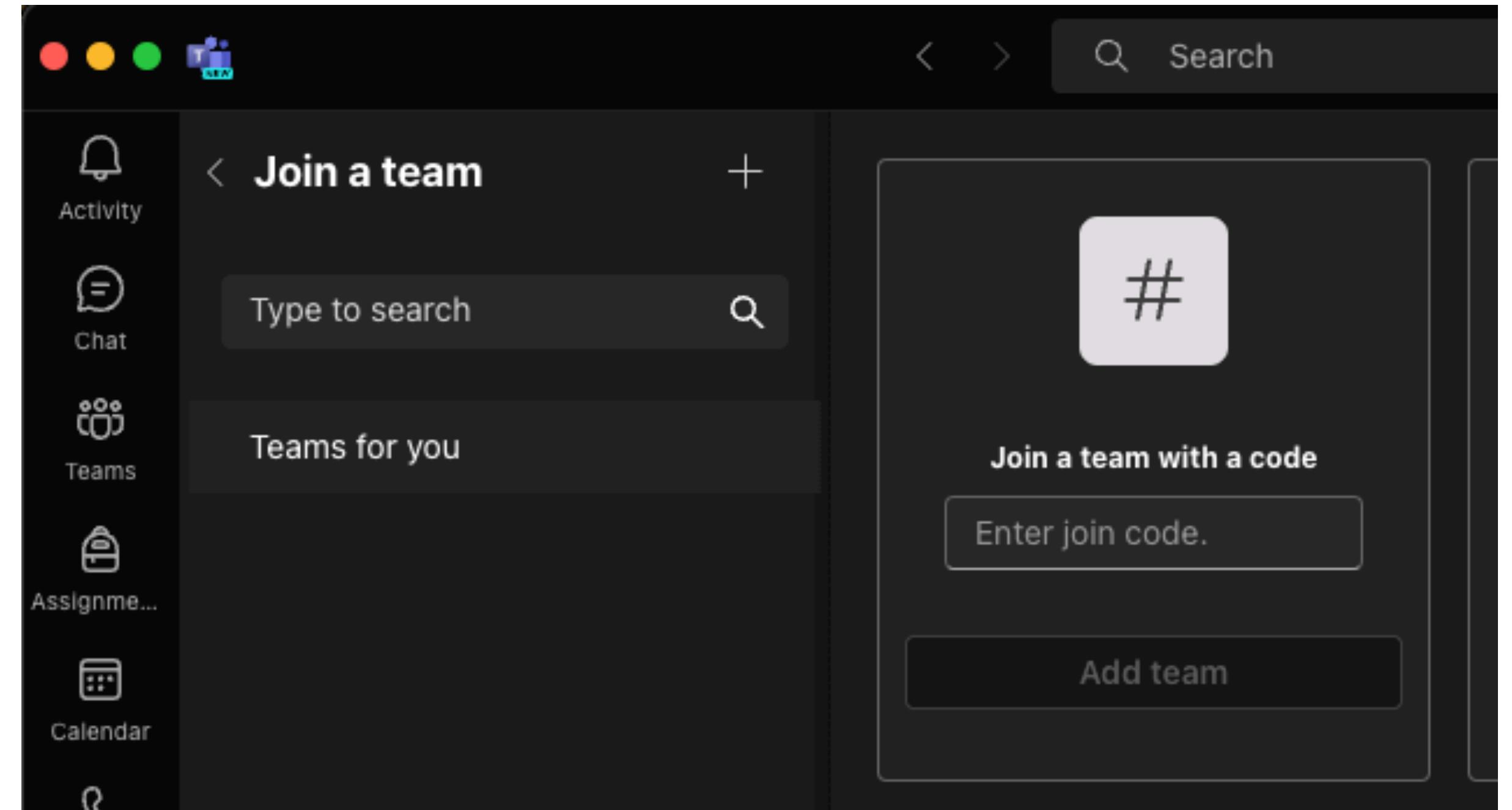
الخوارزمي

# Temps de travail

- Cours ( 4h / sem )
  - Théorie
  - Exercices
- Laboratoires ( 4h / sem )
  - Par groupe de 3 étudiants, sur 2 semaines
  - Toujours du code C++, parfois aussi un rapport
- Travail Personnel ( 6h / sem )
  - Rapports de laboratoire, révisions, ...

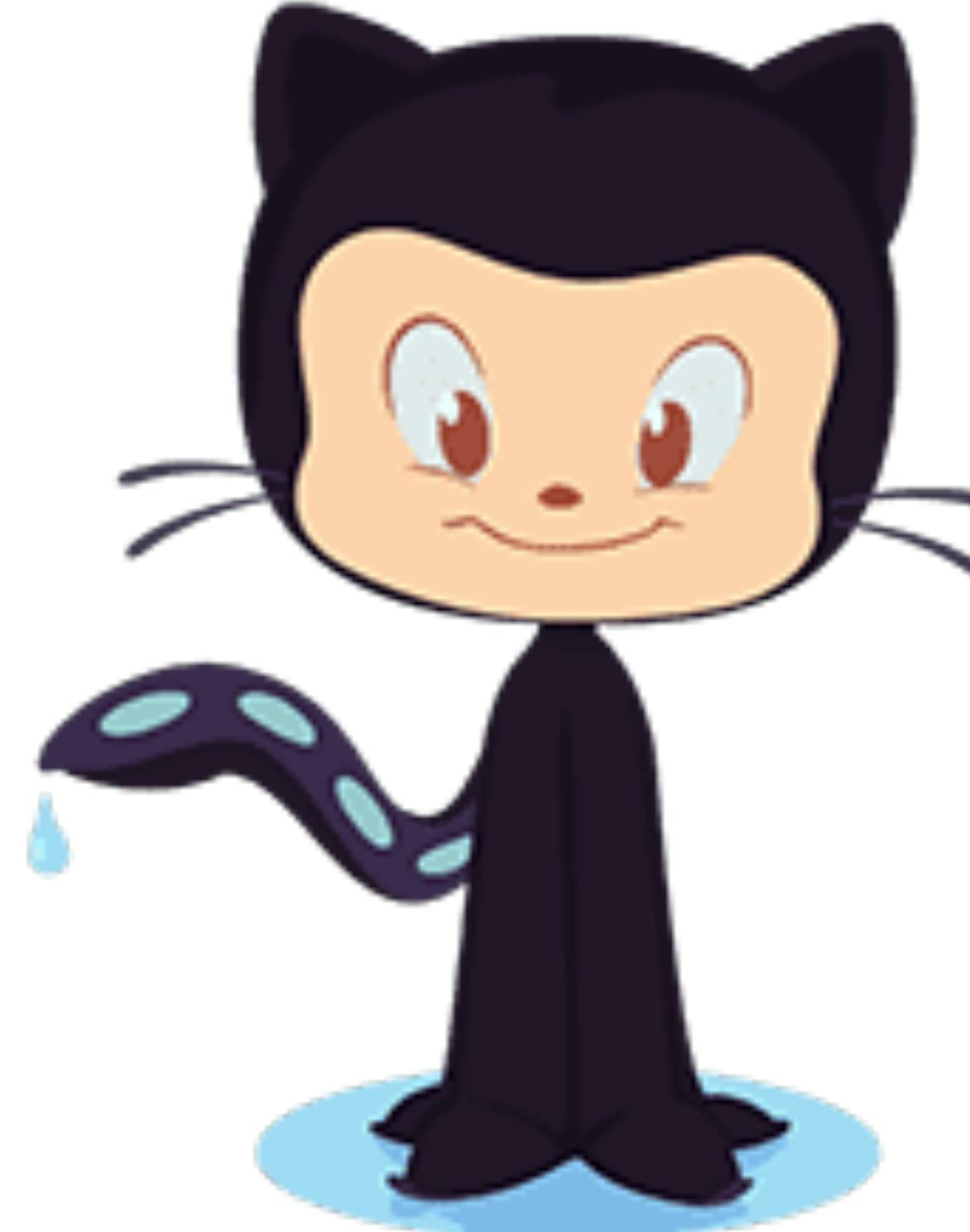


# Teams



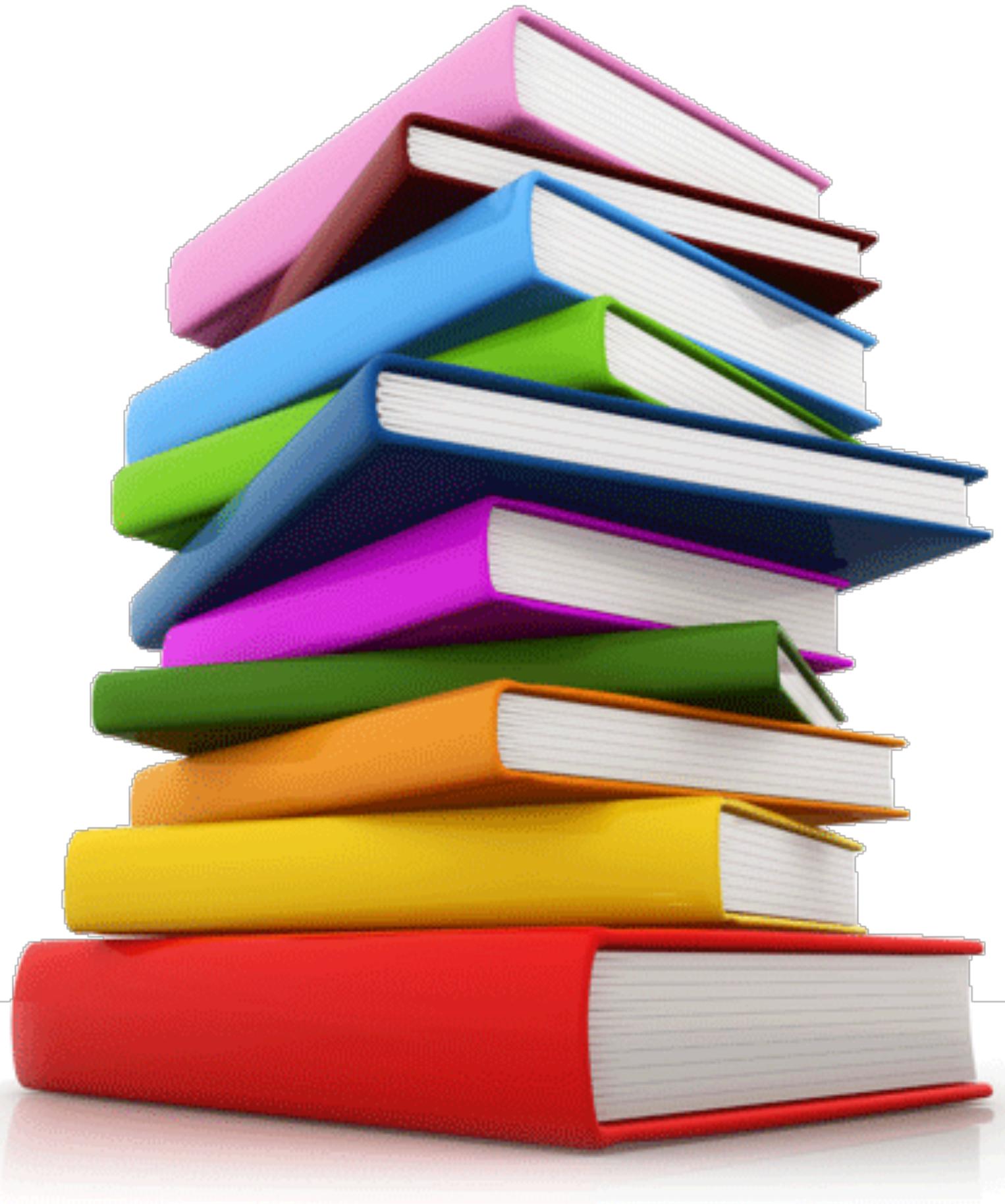
**Teams Class : ASD-D-2024**

# GitHub & GitHub Classroom



Classroom ASD-D-2024

# Bibliographie



- «Introduction to algorithms», Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein, Third Edition, MIT Press, 2009, ISBN-10: 0262033844
- «Algorithmes en C++», Robert Sedgewick, ISBN 2-7440-7056-4
- «Algorithmes et structures de données avec ADA, C++ et Java», A. Guerid, P. Breguet, H. Röthlisberger, Presses Polytechniques et Universitaires Romandes
- «Apprendre le C++», Claude Delannoy, ISBN 978-2-212-12414-9

# Evaluation



Laboratoires (\*)

20%

Tests Ecrits

30%

Examen Final

50%