



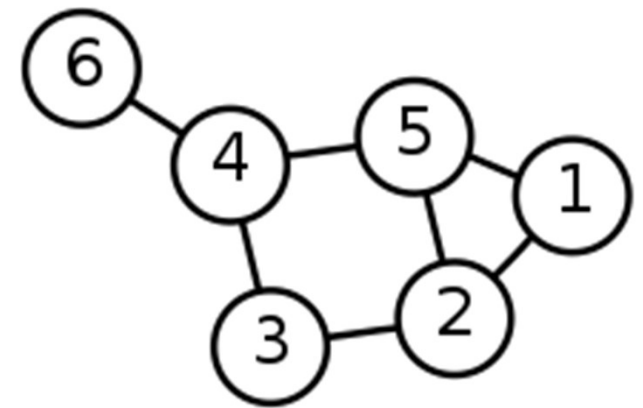
ALGORITHMIE

Théorie des graphes

THÉORIE DES GRAPHES

La théorie des graphes est la discipline mathématique et informatique qui étudie les graphes, lesquels sont des modèles abstraits de dessins de réseaux reliant des objets. Ces modèles sont constitués par la donnée de sommets (aussi appelés nœuds ou points, en référence aux polyèdres), et d'arêtes (aussi appelées liens ou lignes) entre ces sommets ; ces arêtes sont parfois non-symétriques (les graphes sont alors dits orientés) et sont appelés des flèches.

Wikipédia





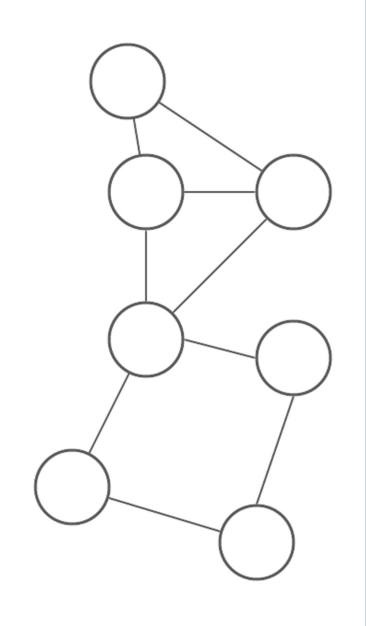
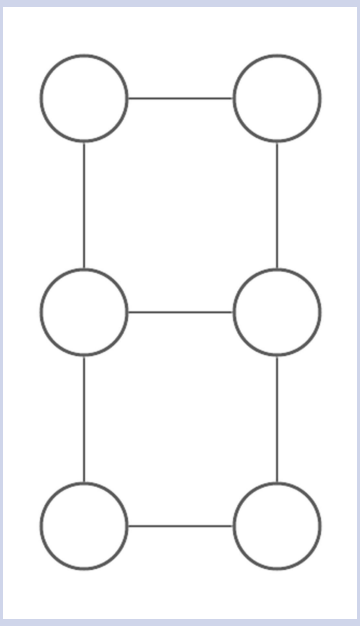
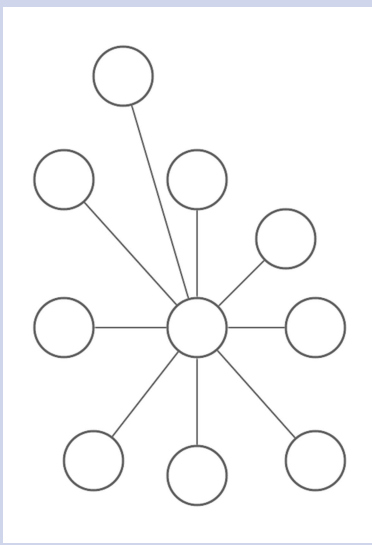
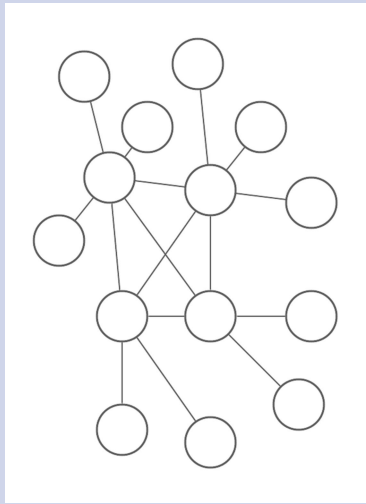
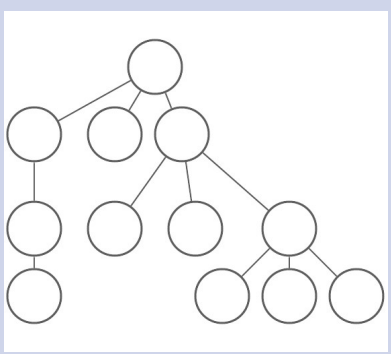
GRAPHES - VOCABULAIRE

Pour faire simple, un graphe est un ensemble :

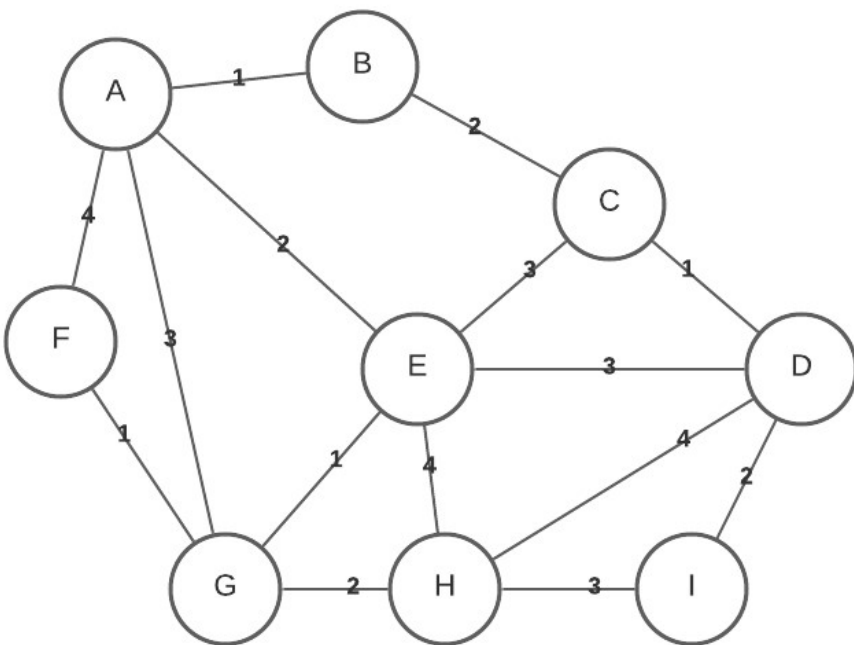
- De nœuds aussi appelés sommets. Ils sont représentés par un rond dont l'étiquette représente le nom du nœud (graphe étiqueté).
- D'arrêtes aussi appelés chemins ou liens. Ils sont représentés par un trait.
 - Une arrête peut avoir un poids (un nombre est écrit sur le trait et représente le coût que représente le fait d'emprunter ce chemin). Le graphe est dit alors pondéré.
 - Une arrête peut être représentée par une flèche. Cela signifie que le chemin ne peut être emprunté que dans le sens de la flèche. Le graphe est dit alors orienté.

Un graphe connexe est un graphe où tous les nœuds sont accessibles quelque soit le nœud de départ.

GRAPHES – PRINCIPAUX TYPES

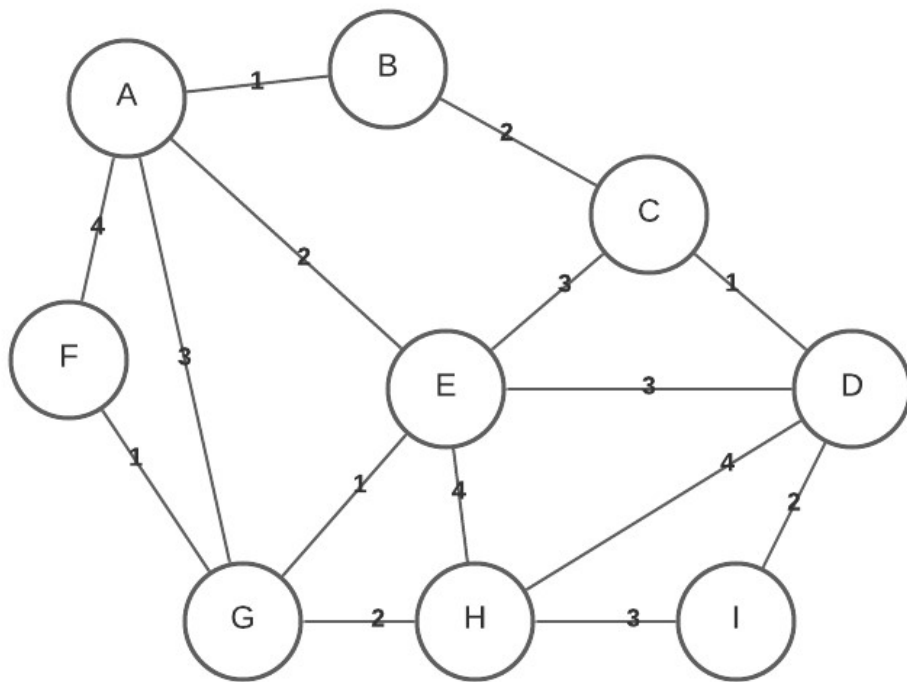
Quelconque	Homogène	Polaire	Multipolaire	Arbre
				

GRAPHES — STRUCTURES DE STOCKAGES



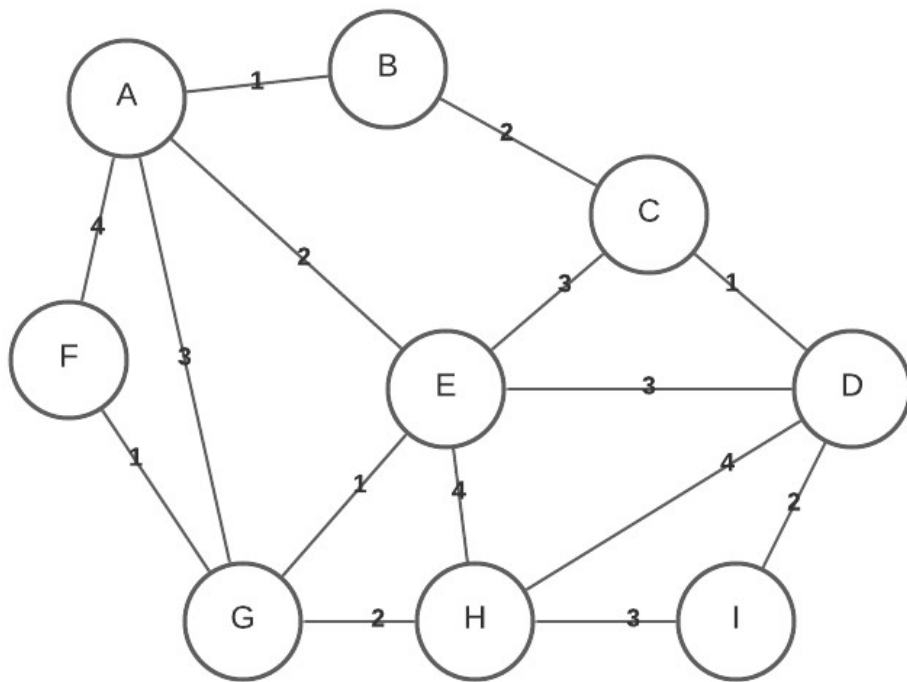
	A	B	C	D	E	F	G	H	I
A		1			2	4	3		
B	1		2						
C		2		1	3				
D			1		3			4	2
E	2		3	3			1	4	
F	4						1		
G	3				1	1		2	
H				4	4		2		3
I				2				3	

GRAPHES – STRUCTURES DE STOCKAGES



```
[
  {
    noeud: {
      nom: A,
      liens: [
        { lien: { sommet: B, poids: 1 } },
        { lien: { sommet: E, poids: 2 } },
        { lien: { sommet: F, poids: 4 } },
        { lien: { sommet: G, poids: 3 } }
      ]
    },
    noeud: {
      nom: B,
      liens: [
        { lien: { sommet: A, poids: 1 } },
        { lien: { sommet: C, poids: 2 } }
      ]
    },
    noeud: {
      nom: C,
      liens: [
        { lien: { sommet: B, poids: 2 } },
        { lien: { sommet: D, poids: 1 } },
        { lien: { sommet: E, poids: 3 } }
      ]
    },
    ...
  ]
]
```


GRAPHES – STRUCTURES DE STOCKAGES



```
[  
  {lien: {poids: 1, sommets: [A,B]}},  
  {lien: {poids: 2, sommets: [A,E]}},  
  {lien: {poids: 3, sommets: [A,F]}},  
  {lien: {poids: 4, sommets: [A,G]}},  
  {lien: {poids: 2, sommets: [B,C]}},  
  {lien: {poids: 1, sommets: [C,D]}},  
  {lien: {poids: 3, sommets: [C,E]}},  
  {lien: {poids: 3, sommets: [D,E]}},  
  {lien: {poids: 4, sommets: [D,H]}},  
  {lien: {poids: 2, sommets: [D,I]}},  
  {lien: {poids: 1, sommets: [E,G]}},  
  {lien: {poids: 4, sommets: [E,H]}},  
  {lien: {poids: 1, sommets: [F,G]}},  
  {lien: {poids: 2, sommets: [G,H]}},  
  {lien: {poids: 3, sommets: [H,I]}},  
]
```



ALGORITHMES DE PRIM

- Objectif :
 - Trouver l'arbre couvrant de poids minimum d'un graphe pondéré connexe
- Fonctionnement :
 - On prend un sommet au hasard et on le stocke dans les sommets connus
 - On cherche à partir des sommets connus, quel est l'arrête permettant d'accéder à un sommet inconnu dont le poids est le plus faible
 - On stocke l'arrête et le nouveau sommet
 - On recommence tant que tous les sommets n'ont pas été parcourus

ALGORITHME DE KRUSKAL

- Objectif :
 - Trouver l'arbre couvrant de poids minimum d'un graphe pondéré connexe
- Fonctionnement :
 - On trie toutes les arrêtes par poids croissant
 - On prend l'arrête de poids le plus faible qui ne crée pas un cycle. C'est-à-dire dont les deux sommets ne sont pas encore dans l'arbre couvrant minimum
 - On recommence tant que tous les sommets n'ont pas été parcourus



ALGORITHME DE BORŮVKA / SOLLIN

- Objectif :
 - Trouver l'arbre couvrant de poids minimum d'un graphe pondéré connexe
- Fonctionnement :
 - On trie toutes les arrêtes par poids croissant
 - On prend l'arrête de poids le plus faible.
 - On la stocke et on fusionne les nœuds de cette arrête.
 - On recommence tant que l'arbre n'est pas réduit à un seul sommet.



ALGORITHME DE DIJKSTRA

- Objectif :
 - Résoudre le problème du plus court chemin
- Fonctionnement :
 - Illustration avec un exemple car je pense que cela sera beaucoup plus explicite qu'une explication.
 - Sinon pour l'explication : https://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra

EXERCICE :

- Coder l'un des algorithmes d'arbre couvrant minimum et le tester avec le graphe du plateau de jeu les aventuriers du rail Europe.



jeanamerico@gmail.com