Notations : $G=(V,E)$

Graphe orienté $G=(V,E)$

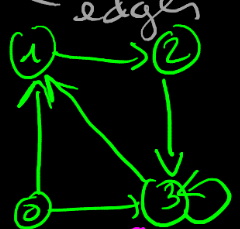
vertices

V ensemble de sommets non vide

$E \subseteq V^2$ ensemble d'arêtes peut être vide

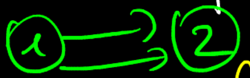
edges

Autoriser $V=\emptyset$ pose plein de problèmes. Par exemple $G=(\emptyset,\emptyset)$ est connexe et acyclique pour autant ce n'est pas un arbre car $|E| \neq |V|-1$.



$V=\{0,1,2,3\}$

$E=\{(0,1),(0,3),(1,2),(2,3),(3,3),(3,1)\}$

Pb de cette notation : pas possible de représenter . comme celui-là

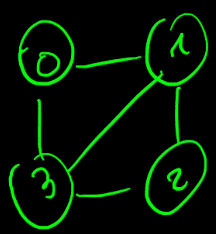
On appelle graphe simple un graphe sans arête multiple et sans boucle sur un sommet.

Graphe non orienté $G=(V,E)$

V idem

$E \subseteq V^2$ idem

On convient que les arêtes ne sont représentées qu'une fois dans E (afin que $|E|$ soit toujours le nombre d'arêtes) et que l'opérateur $(x,y) \in E$ est redéfini pour retourner vrai ssi (x,y) ou (y,x) est dans E. Ceci permet d'écrire du pseudo-code qui marche aussi bien sur des graphes orientés que non orientés.



$V=\{0,1,2,3\}$

$E=\{(0,1),(0,3),(1,3),(1,2),(2,3)\}$

avec $(2,1) \in E$ mais $(2,0) \notin E$.

Note certains auteurs utilisent la notation $\{x,y\} \in E$ avec $E \subseteq \binom{V}{2}$

pas d'ordre

paires non ordonées de deux sommets

avec l'inconvénient d'une notation différente selon le cas orienté ou non.

Sur graphe orienté $G=(V,E)$

Pour un sommet $x \in V$:

$\delta^+(x) = \{y \in V \mid (x,y) \in E\}$ successeurs

$\delta^-(x) = \{y \in V \mid (y,x) \in E\}$ prédécesseurs

$\deg^+(x) = |\delta^+(x)|$ degré sortant

$\deg^-(x) = |\delta^-(x)|$ degré entrant

Sur un graphe non-orienté

$\delta^+(x) = \delta^-(x) = \delta(x) = \{y \in V \mid (x,y) \in E\}$ voisins

$\deg(x) = |\delta(x)|$ degrés (tout court)

On a toujours

(orienté) $$\sum_{x \in V} \deg^+(x) = \sum_{x \in V} \deg^-(x) = |E|$$

(non orienté) $$\sum_{x \in V} \deg(x) = |E| \times 2$$

IMPORTANT

pour chaque sommet x faire :
pour chaque y voisin de x, faire :

$\boxed{\text{ce code fait } |E| \text{ itérations}}$ ou $2 \times |E|$ $\Big\}\ \Theta(|E|)$

La complexité d'un algo de graphe se calcule en fonction de $|V|$ et $|E|$.

P.ex. L'algo de Dijkstra (avec la bonne implémentation) tourne en $O(|E|+|V|\log|V|)$

Un algo en $O(|V|+|E|)$ est dit linéaire, car $|V|+|E|$ représente la taille du graphe.

Certaines simplifications sont possibles:

sur un graphe simple $|E| = O(|V|^2)$

sur un arbre $|E| = \Theta(|V|)$

sur un graphe connexe $|V| = O(|E|)$

IMPORTANT

Ainsi, la complexité d'un algo linéaire $O(|V|+|E|)$ peut être simplifiée en $O(|E|)$ si l'on ne travaille que sur des graphes connexes.

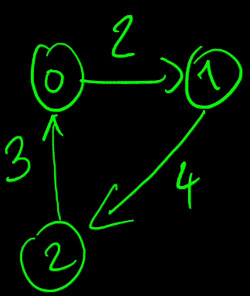
Graphe pondéré (orienté ou non)

$$G = (V, E, w)$$

weight

$$w : E \to W$$

ensemble à spécifier au cas par cas.



$V = \{0, 1, 2\}$

$E = \{(0,1), (1,2), (3,0)\}$

$w : E \to \mathbb{N}$

$(0,1) \mapsto 2$

$(1,2) \mapsto 4$

$(2,0) \mapsto 3$

Stocker cela demande donc $\Theta(|E|)$ espace du calcul on continue les complexités en fonction de $|V|$ et $|E|$.