Algorithmique des graphes

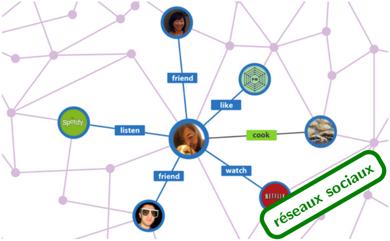
1 — Les bases

Anthony Labarre

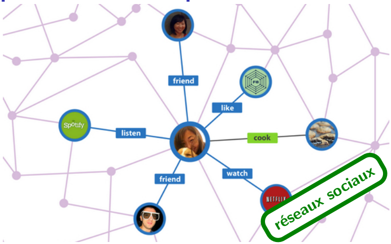
27 janvier 2021

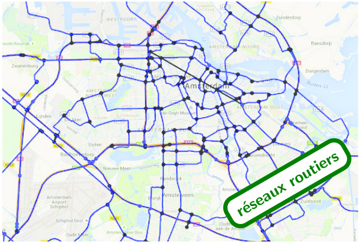
Université Gustave Eiffel

Graphes : exemples informels

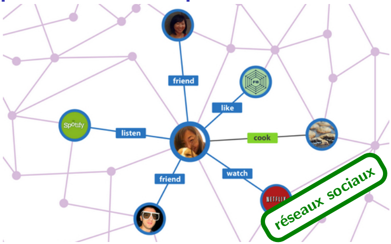


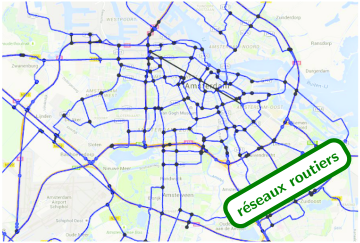
Graphes : exemples informels

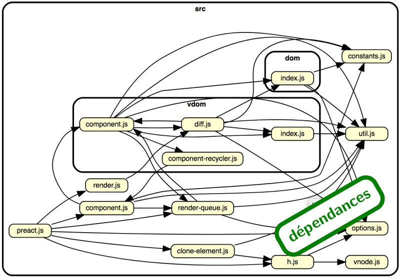




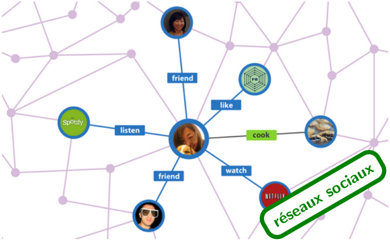
Graphes : exemples informels

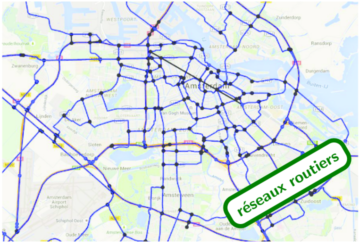


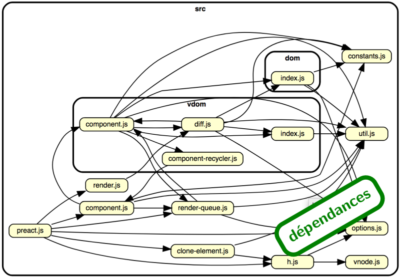


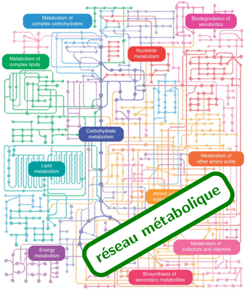


Graphes : exemples informels

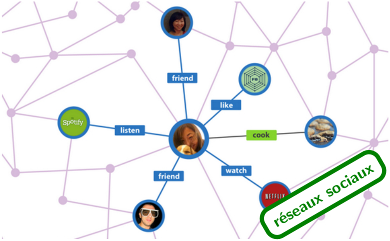


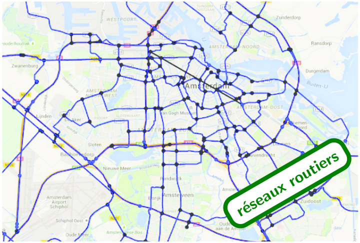


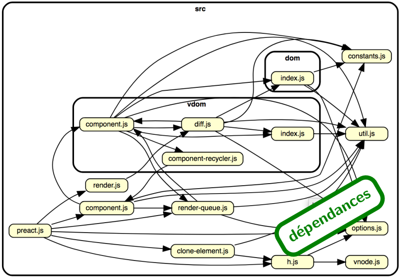


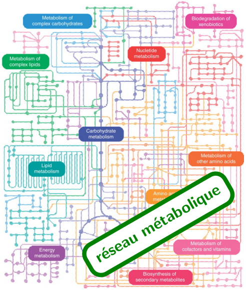


Graphes : exemples informels









⇒ Graphe = éléments reliés par une certaine relation

Concepts de base

Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.

Concepts de base

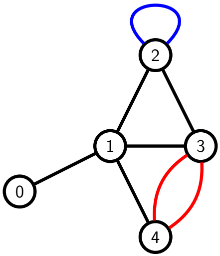
Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.

Exemple 1



Concepts de base

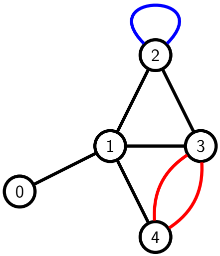
Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.
G est **simple** s'il ne contient ni **arêtes parallèles** ni **boucles**.

Exemple 1



Concepts de base

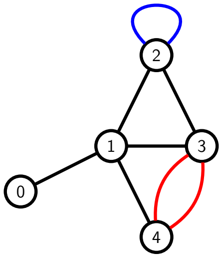
Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.
G est **simple** s'il ne contient ni **arêtes parallèles** ni **boucles**.

Exemple 1



- Une arête $\{u, v\}$ relie deux sommets **adjacents** ou **voisins** ; elle est **incidente** à u et v, qui sont ses **extrémités** ;

Concepts de base

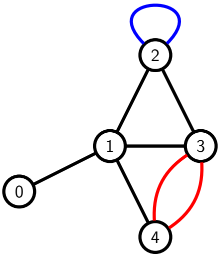
Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.
G est **simple** s'il ne contient ni **arêtes parallèles** ni **boucles**.

Exemple 1



- Une arête $\{u, v\}$ relie deux sommets **adjacents** ou **voisins** ; elle est **incidente** à u et v, qui sont ses **extrémités** ;
- Le **voisinage** d'un sommet v est l'ensemble de ses voisins :

$$N_G(v) = \{u \mid \{u, v\} \in E(G)\}$$

Concepts de base

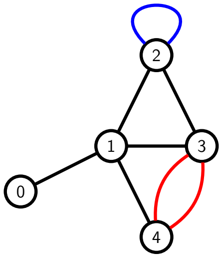
Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.
G est **simple** s'il ne contient ni **arêtes parallèles** ni **boucles**.

Exemple 1



- Une arête $\{u, v\}$ relie deux sommets **adjacents** ou **voisins** ; elle est **incidente** à u et v, qui sont ses **extrémités** ;
- Le **voisinage** d'un sommet v est l'ensemble de ses voisins :

$$N_G(v) = \{u \mid \{u, v\} \in E(G)\}$$

- Le **degré** du sommet v est la taille de son voisinage, noté $\deg_G(v)$;

Concepts de base

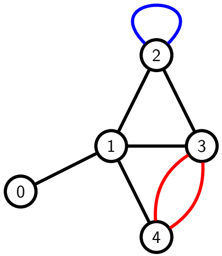
Définition 1

Un **graphe** est un couple $G = (V, E)$, où :

- V est un ensemble de **sommets** ;
- $E \subseteq V \times V$ un ensemble d'**arêtes** ;

On utilisera aussi les notations $V(G)$ et $E(G)$ pour bien distinguer le graphe G d'un autre graphe.
G est **simple** s'il ne contient ni **arêtes parallèles** ni **boucles**.

Exemple 1



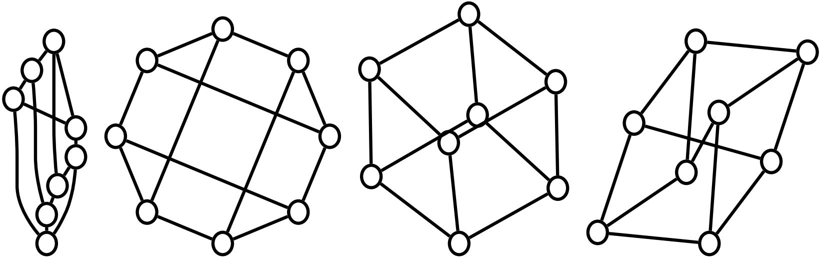
- Une arête $\{u, v\}$ relie deux sommets **adjacents** ou **voisins** ; elle est **incidente** à u et v, qui sont ses **extrémités** ;
- Le **voisinage** d'un sommet v est l'ensemble de ses voisins :

$$N_G(v) = \{u \mid \{u, v\} \in E(G)\}$$

- Le **degré** du sommet v est la taille de son voisinage, noté $\deg_G(v)$;
- Deux sommets qui ne sont pas voisins sont **indépendants** ;

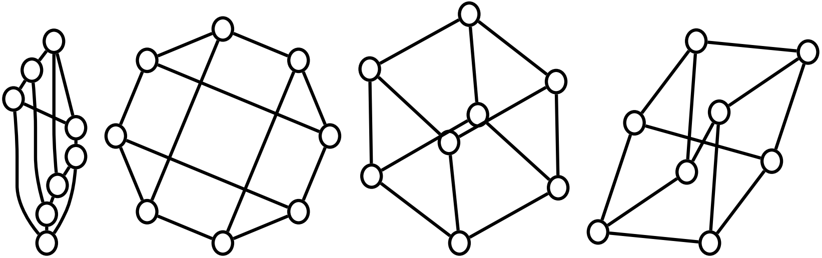
Étiquetages et dessins

Les sommets et les arêtes peuvent être étiquetés, mais ce n'est pas obligatoire. La manière dont on dessine les graphes n'importe pas :



Étiquetages et dessins

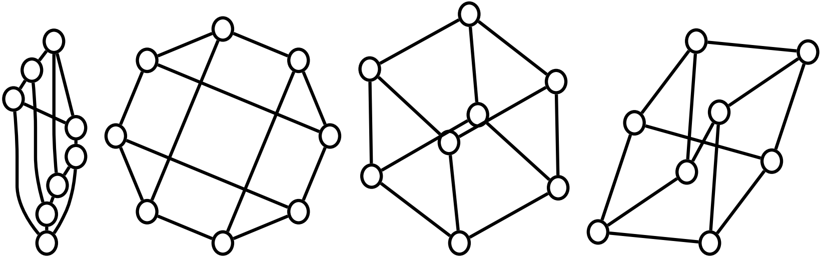
Les sommets et les arêtes peuvent être étiquetés, mais ce n'est pas obligatoire. La manière dont on dessine les graphes n'importe pas :



Les graphes G et H sont **isomorphes** si l'on peut numéroter $V(G)$ et $V(H)$ de telle sorte que $E(G) = E(H)$.

Étiquetages et dessins

Les sommets et les arêtes peuvent être étiquetés, mais ce n'est pas obligatoire. La manière dont on dessine les graphes n'importe pas :

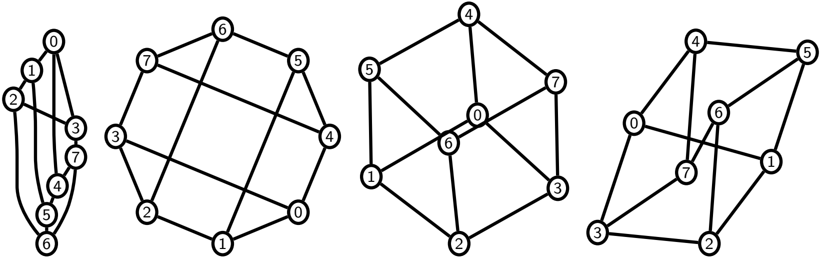


Les graphes G et H sont **isomorphes** si l'on peut numéroter $V(G)$ et $V(H)$ de telle sorte que $E(G) = E(H)$.

On ne connaît pas la complexité du problème de décision associé ; le meilleur algorithme connu est en $\exp((\log n)^{O(1)})$ [1].

Isomorphisme

Voici des étiquetages prouvant que les graphes précédents sont tous isomorphes :



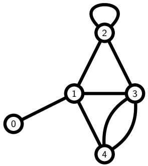
Les arêtes deviennent $\{\{0,1\},\{0,3\},\{0,4\},\ldots,\{6,7\}\}$.

Sous-graphes

On s'intéressera régulièrement à un "morceau" particulier d'un graphe G.

- Un **sous-graphe** d'un graphe $G = (V, E)$ est un graphe $H = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$.

Exemple 2



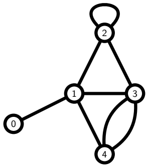
G

Sous-graphes

On s'intéressera régulièrement à un "morceau" particulier d'un graphe G.

- Un **sous-graphe** d'un graphe $G = (V, E)$ est un graphe $H = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$.
- H est **induit** par V' si E' contient toutes les arêtes de E reliant deux sommets de V' dans G ; on écrit alors $H = G[V']$.

Exemple 2



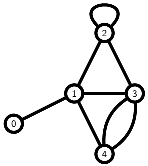
G

Sous-graphes

On s'intéressera régulièrement à un "morceau" particulier d'un graphe G.

- Un **sous-graphe** d'un graphe $G = (V, E)$ est un graphe $H = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$.
- H est **induit** par V' si E' contient toutes les arêtes de E reliant deux sommets de V' dans G ; on écrit alors $H = G[V']$.
- H est **induit** par F' si V' contient seulement les sommets de V reliés par F' dans G ; on écrit alors $H = G[F']$.

Exemple 2



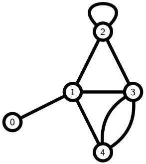
G

Sous-graphes

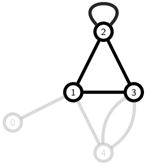
On s'intéressera régulièrement à un "morceau" particulier d'un graphe G.

- Un **sous-graphe** d'un graphe $G = (V, E)$ est un graphe $H = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$.
- H est **induit** par V' si E' contient toutes les arêtes de E reliant deux sommets de V' dans G ; on écrit alors $H = G[V']$.
- H est **induit** par F' si V' contient seulement les sommets de V reliés par F' dans G ; on écrit alors $H = G[F']$.

Exemple 2



G



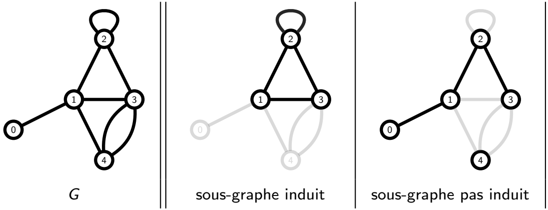
sous-graphe induit

Sous-graphes

On s'intéressera régulièrement à un "morceau" particulier d'un graphe G.

- Un **sous-graphe** d'un graphe $G = (V, E)$ est un graphe $H = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$.
- H est **induit** par V' si E' contient toutes les arêtes de E reliant deux sommets de V' dans G ; on écrit alors $H = G[V']$.
- H est **induit** par F' si V' contient seulement les sommets de V reliés par F' dans G ; on écrit alors $H = G[F']$.

Exemple 2

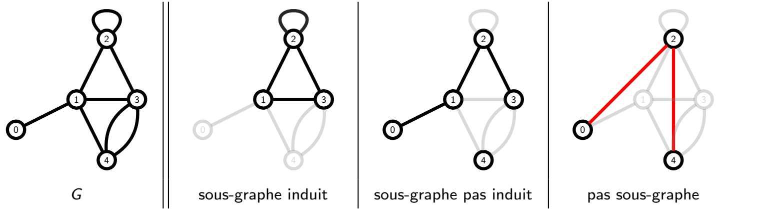


Sous-graphes

On s'intéressera régulièrement à un "morceau" particulier d'un graphe G.

- Un **sous-graphe** d'un graphe $G = (V, E)$ est un graphe $H = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$.
- H est **induit** par V' si E' contient toutes les arêtes de E reliant deux sommets de V' dans G ; on écrit alors $H = G[V']$.
- H est **induit** par F' si V' contient seulement les sommets de V reliés par F' dans G ; on écrit alors $H = G[F']$.

Exemple 2



Graphes fréquents

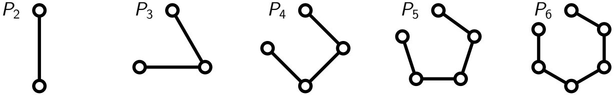
Certains graphes reviennent souvent dans les applications :

- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \ldots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p-2$;

Graphes fréquents

Certains graphes reviennent souvent dans les applications :

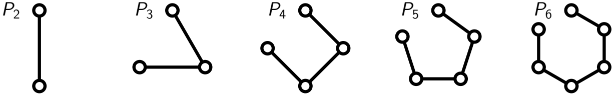
- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \ldots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p - 2$;



Graphes fréquents

Certains graphes reviennent souvent dans les applications :

- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \ldots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p-2$;

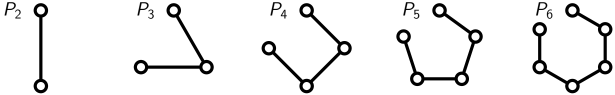


- un **cycle** dans un graphe G est une séquence $C = (u_0, u_1, u_2, \ldots, u_p)$ de sommets où $\{u_i,\ u_{i+1 \mod p}\} \in E(G)$ pour $0 \leq i \leq p-1$.

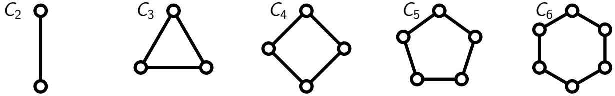
Graphes fréquents

Certains graphes reviennent souvent dans les applications :

- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \ldots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p-2$;



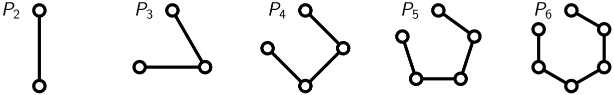
- un **cycle** dans un graphe G est une séquence $C = (u_0, u_1, u_2, \ldots, u_p)$ de sommets où $\{u_i,\ u_{i+1 \mod p}\} \in E(G)$ pour $0 \leq i \leq p-1$.



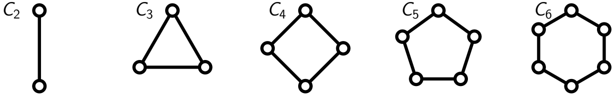
Graphes fréquents

Certains graphes reviennent souvent dans les applications :

- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \ldots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p-2$;



- un **cycle** dans un graphe G est une séquence $C = (u_0, u_1, u_2, \ldots, u_p)$ de sommets où $\{u_i,\ u_{i+1 \mod p}\} \in E(G)$ pour $0 \leq i \leq p-1$.

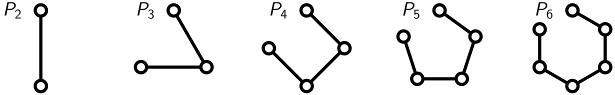


- enfin, on dit d'un graphe $G = (V, E)$ qu'il est **complet** si $E = V \times V$, c'est-à-dire que chaque sommet est adjacent à tous les autres.

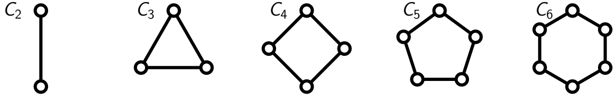
Graphes fréquents

Certains graphes reviennent souvent dans les applications :

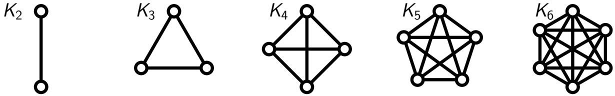
- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \ldots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p-2$;



- un **cycle** dans un graphe G est une séquence $C = (u_0, u_1, u_2, \ldots, u_p)$ de sommets où $\{u_i,\ u_{i+1 \mod p}\} \in E(G)$ pour $0 \leq i \leq p-1$.



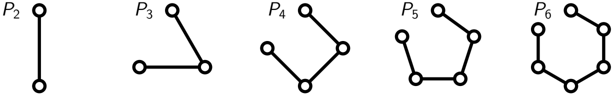
- enfin, on dit d'un graphe $G = (V, E)$ qu'il est **complet** si $E = V \times V$, c'est-à-dire que chaque sommet est adjacent à tous les autres.



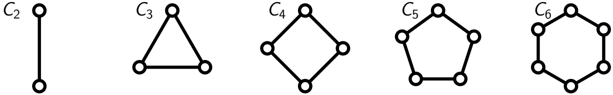
Graphes fréquents

Certains graphes reviennent souvent dans les applications :

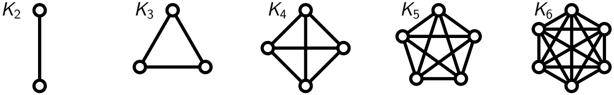
- un **chemin** dans un graphe G est une séquence $P = (u_0, u_1, u_2, \dots, u_{p-1})$ de sommets **distincts** où $\{u_i, u_{i+1}\} \in E(G)$ pour $0 \leq i \leq p-2$;



- un **cycle** dans un graphe G est une séquence $C = (u_0, u_1, u_2, \dots, u_p)$ de sommets où $\{u_i,\ u_{i+1 \mod p}\} \in E(G)$ pour $0 \leq i \leq p-1$.



- enfin, on dit d'un graphe $G = (V, E)$ qu'il est **complet** si $E = V \times V$, c'est-à-dire que chaque sommet est adjacent à tous les autres.



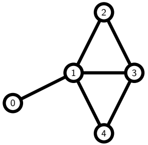
La **longueur** d'un cycle ou d'un chemin est son nombre d'arêtes.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \tag{1}$$

Exemple 3

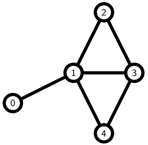


Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \tag{1}$$

Exemple 3



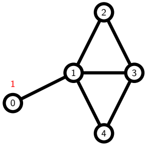
En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \qquad (1)$$

Exemple 3



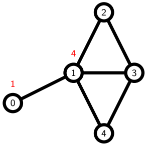
En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \qquad (1)$$

Exemple 3



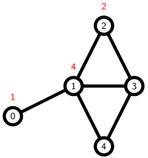
En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \tag{1}$$

Exemple 3



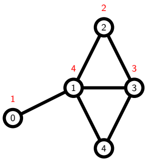
En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \qquad (1)$$

Exemple 3



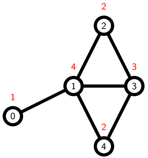
En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \qquad (1)$$

Exemple 3



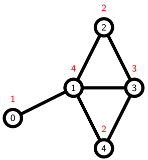
En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \qquad (1)$$

Exemple 3



En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

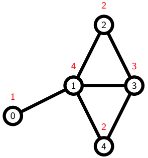
K_n contient le nombre maximum d'arêtes, à savoir $\binom{n}{2} = n(n-1)/2$.

Degrés et taille

La **taille** d'un graphe $G = (V, E)$ est son nombre d'arêtes $|E|$. Si G est simple, on peut la calculer à l'aide de la relation suivante :

$$\sum_{v \in V} \deg(v) = 2|E|. \qquad (1)$$

Exemple 3



En effet, si l'on parcourt les sommets du graphe en sélectionnant chaque arête incidente à chacun des sommets, on aura sélectionné chaque arête exactement deux fois.

K_n contient le nombre maximum d'arêtes, à savoir $\binom{n}{2} = n(n-1)/2$.

Donc pour tout graphe $G = (V, E)$: $|E| = O(|V|^2)$, et $\log |E| = O(\log |V|)$.

Représentation de graphes en mémoire

- Les deux encodages de graphes les plus fréquents se basent sur des *matrices* ou des *listes* d'adjacence ;

Représentation de graphes en mémoire

- Les deux encodages de graphes les plus fréquents se basent sur des *matrices* ou des *listes* d'adjacence ;
- Le choix de la représentation est important, car :

Représentation de graphes en mémoire

- Les deux encodages de graphes les plus fréquents se basent sur des *matrices* ou des *listes* d'adjacence ;
- Le choix de la représentation est important, car :
  1. la consommation en mémoire n'est pas la même ;

Représentation de graphes en mémoire

- Les deux encodages de graphes les plus fréquents se basent sur des *matrices* ou des *listes* d'adjacence ;
- Le choix de la représentation est important, car :
  1. la consommation en mémoire n'est pas la même ;
  2. la complexité des opérations diffère aussi ;

Représentation de graphes en mémoire

- Les deux encodages de graphes les plus fréquents se basent sur des *matrices* ou des *listes* d'adjacence ;
- Le choix de la représentation est important, car :
  1. la consommation en mémoire n'est pas la même ;
  2. la complexité des opérations diffère aussi ;
- Le "bon" choix dépend des algorithmes et des applications qui nous intéressent ;

Matrice d'adjacence

Définition 2

La **matrice d'adjacence** $A(G)$ du graphe $G = (V, E)$ contient un 1 en ligne i et en colonne j si l'arête $\{i, j\}$ existe, et un 0 sinon :

$$A_{ij}(G) = \begin{cases} 1 & \text{si } \{i, j\} \in E, \\ 0 & \text{sinon.} \end{cases}$$

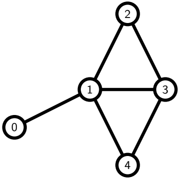
Matrice d'adjacence

Définition 2

La **matrice d'adjacence** $A(G)$ du graphe $G = (V, E)$ contient un 1 en ligne i et en colonne j si l'arête $\{i, j\}$ existe, et un 0 sinon :

$$A_{ij}(G) = \begin{cases} 1 & \text{si } \{i,j\} \in E, \\ 0 & \text{sinon.} \end{cases}$$

Exemple 4



	0	1	2	3	4
0	0	1	0	0	0
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	0	1	0	1	0

	0	1	2	3	4
0	0				
1	1	0			
2	0	1	0		
3	0	1	1	0	
4	0	1	0	1	0

010010011001010

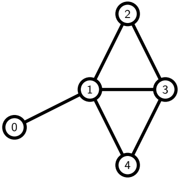
Matrice d'adjacence

Définition 2

La **matrice d'adjacence** $A(G)$ du graphe $G = (V, E)$ contient un 1 en ligne i et en colonne j si l'arête $\{i, j\}$ existe, et un 0 sinon :

$$A_{ij}(G) = \begin{cases} 1 & \text{si } \{i, j\} \in E, \\ 0 & \text{sinon.} \end{cases}$$

Exemple 4



	0	1	2	3	4
0	0	1	0	0	0
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	0	1	0	1	0

	0	1	2	3	4
0	0				
1	1	0			
2	0	1	0		
3	0	1	1	0	
4	0	1	0	1	0

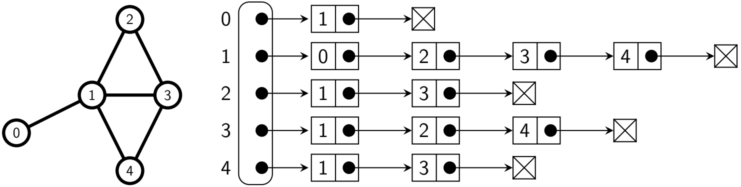
010010011001010

Les connexions sont symétriques, donc $A(G)$ aussi, et on pourrait donc n'en garder que la moitié. S'il n'y a pas de poids, une chaîne binaire suffit.

Listes d'adjacence

Les **listes d'adjacence** encodent, pour chaque sommet du graphe $G = (V, E)$, la liste des voisins de ce sommet.

Exemple 5



L'API Graphe

Les algorithmes étudiés supposeront l'existence d'une classe `Graphe` proposant les méthodes suivantes :

Méthode
`ajouter_arete(u, v)`
`ajouter_aretes(séquence)`
`ajouter_sommet()`
`aretes()`
`boucles()`
`contient_arete(u, v)`
`contient_sommet(u)`
`degre(sommet)`
`nombre_aretes()`
`nombre_boucles()`
`nombre_sommets()`
`retirer_arete(u, v)`
`retirer_aretes(séquence)`
`retirer_sommet(sommet)`
`retirer_sommets(séquence)`
`sommets()`
`sous_graphe_induit(séquence)`
`voisins(sommet)`

Exercice

Quelle est la complexité de chacune des méthodes, dans le cas des matrices et des listes d'adjacence ?

Comparaison des complexités

Les algorithmes étudiés supposeront l'existence d'une classe `Graphe` proposant les méthodes suivantes :

Méthode	Matrice d'adjacence	Liste d'adjacence
`ajouter_arete(u, v)`	$O(1)$ si u et v existent	$O(\deg(u) + \deg(v))$ si u et v existent
`ajouter_aretes(séquence)`	$O(\lvert\text{séquence}\rvert)$	$O(\Delta \times \lvert\text{séquence}\rvert)$
`ajouter_sommet()`	$O(n)$	$O(1)$
`aretes()`	$O(n^2)$	$O(m)$
`boucles()`	$O(n)$	$O(m)$
`contient_arete(u, v)`	$O(1)$	$O(\deg(u) + \deg(v))$
`contient_sommet(u)`	$O(1)$	$O(1)$
`degre(sommet)`	$O(n)$	$O(1)$
`nombre_aretes()`	$O(n^2)$	$O(n)$
`nombre_boucles()`	$O(n)$	$O(m)$
`nombre_sommets()`	$O(1)$	$O(1)$
`retirer_arete(u, v)`	$O(1)$	$O(\deg(u) + \deg(v))$
`retirer_aretes(séquence)`	$O(\lvert\text{séquence}\rvert)$	$O(\Delta \times \lvert\text{séquence}\rvert)$
`retirer_sommet(sommet)`	$O(n^2)$	$O(m + n)$
`retirer_sommets(séquence)`	$O(n^2 \times \lvert\text{séquence}\rvert)$	$O((m + n) \times \lvert\text{séquence}\rvert)$
`sommets()`	$O(n)$	$O(n)$
`sous_graphe_induit(séquence)`	$O(\lvert\text{séquence}\rvert^2)$	$O(\Delta \times \lvert\text{séquence}\rvert^2)$
`voisins(sommet)`	$O(n)$	$O(\deg(\mathit{sommet}))$

Δ est le degré maximum du graphe ; $n = |V(G)|$, $m = |E(G)|$.

Remarques sur ces implémentations

- On peut améliorer les complexités de certaines opérations dans les deux cas ;

Remarques sur ces implémentations

- On peut améliorer les complexités de certaines opérations dans les deux cas ;
- ... mais ces choix peuvent aussi avoir un impact négatif sur d'autres opérations ou l'espace consommé ;

Remarques sur ces implémentations

- On peut améliorer les complexités de certaines opérations dans les deux cas ;
- ... mais ces choix peuvent aussi avoir un impact négatif sur d'autres opérations ou l'espace consommé ;
- En général, on implémente plusieurs classes, et on utilise “la bonne” en fonction de la situation ;

Remarques sur ces implémentations

- On peut améliorer les complexités de certaines opérations dans les deux cas ;
- … mais ces choix peuvent aussi avoir un impact négatif sur d'autres opérations ou l'espace consommé ;
- En général, on implémente plusieurs classes, et on utilise "la bonne" en fonction de la situation ;
- Il vaut toujours mieux séparer les identifiants des propriétés ;

Remarques sur ces implémentations

- On peut améliorer les complexités de certaines opérations dans les deux cas ;
- ... mais ces choix peuvent aussi avoir un impact négatif sur d'autres opérations ou l'espace consommé ;
- En général, on implémente plusieurs classes, et on utilise "la bonne" en fonction de la situation ;
- Il vaut toujours mieux séparer les identifiants des propriétés ;
- Par exemple : ne pas utiliser `"bonjour"` pour identifier un sommet, mais plutôt un naturel `k` et éventuellement une structure `noms` avec `nom[k] = "bonjour"` ;

Graphviz et dot

Graphviz est une suite d'outils permettant de visualiser des graphes exprimés dans le langage `dot`. Le format est simple :

Exemple 6

```
graph G {
    # liste de sommets avec leurs propriétés
    0;
    1;
    2;
    # ...
    # liste d'arêtes avec leurs propriétés
    0 -- 1;
    1 -- 2;
    3 -- 7;
    # ...
}
```

Visualisation

Les graphes exprimés dans le langage `dot` peuvent être convertis en images en ligne (`http://www.webgraphviz.com/`) ou à l'aide d'un des outils suivants :

- `dot` : dessine les graphes de manière hiérarchique, typiquement utilisé pour les graphes orientés (voir plus loin) ou les arbres.
- `neato` : utilisé en général pour les graphes de taille raisonnable (moins de 100 sommets).
- `fdp` : similaire à `neato`.
- `sfdp` : plus adapté aux grands graphes.
- `twopi` : dispose un sommet au centre, et les autres sur des cercles concentriques autour de ce centre.
- `circo` : dispose les sommets du graphe sur un cercle.
- `osage` : plus adapté aux graphes "en couches".

Bibliographie

[1] László Babai.

Graph isomorphism in quasipolynomial time [extended abstract].

In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016.