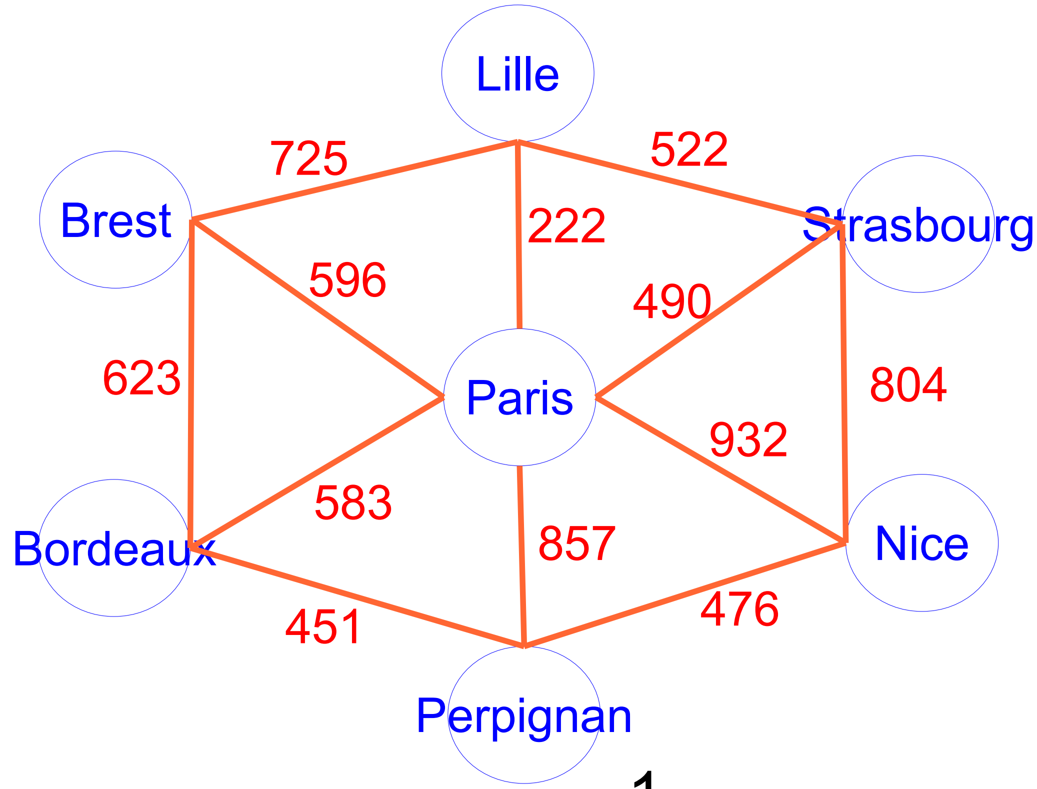
Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

$$v : E \to R^{*+}$$

Exemple. Des villes et leurs distances



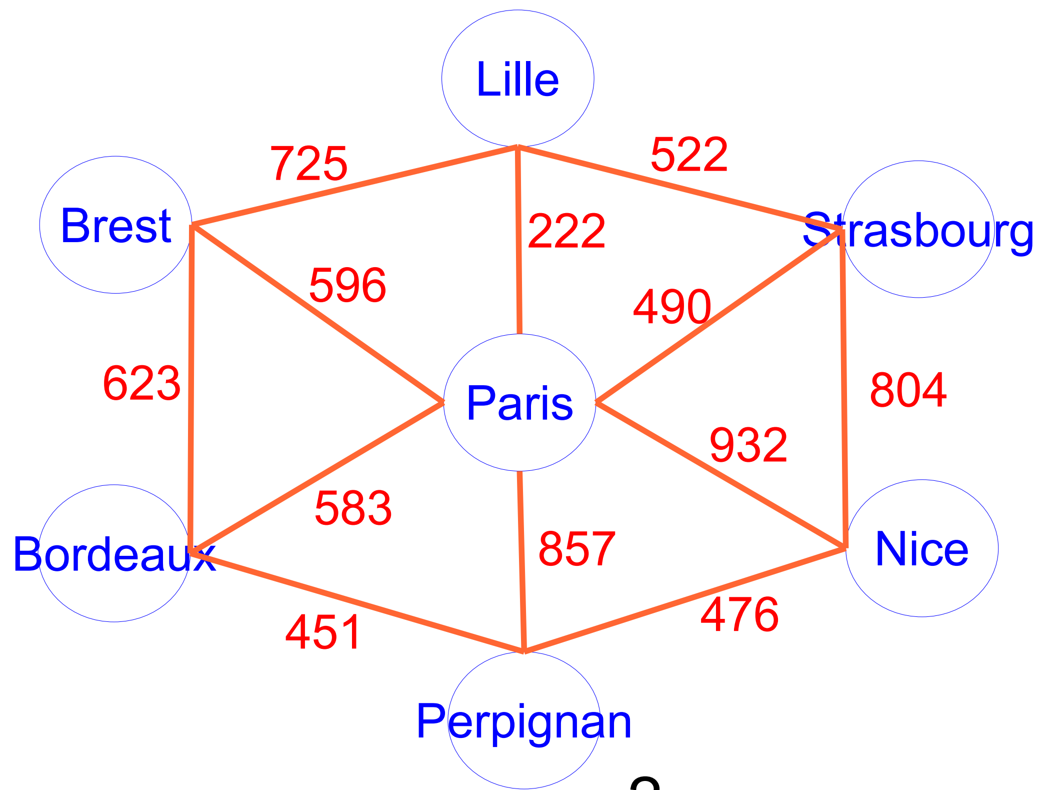
Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

$$v : E \to R^{*+}$$

Exemple. Des villes et leurs distances



Problème 1. Construire un *réseau qui permet de connecter deux villes qcq*

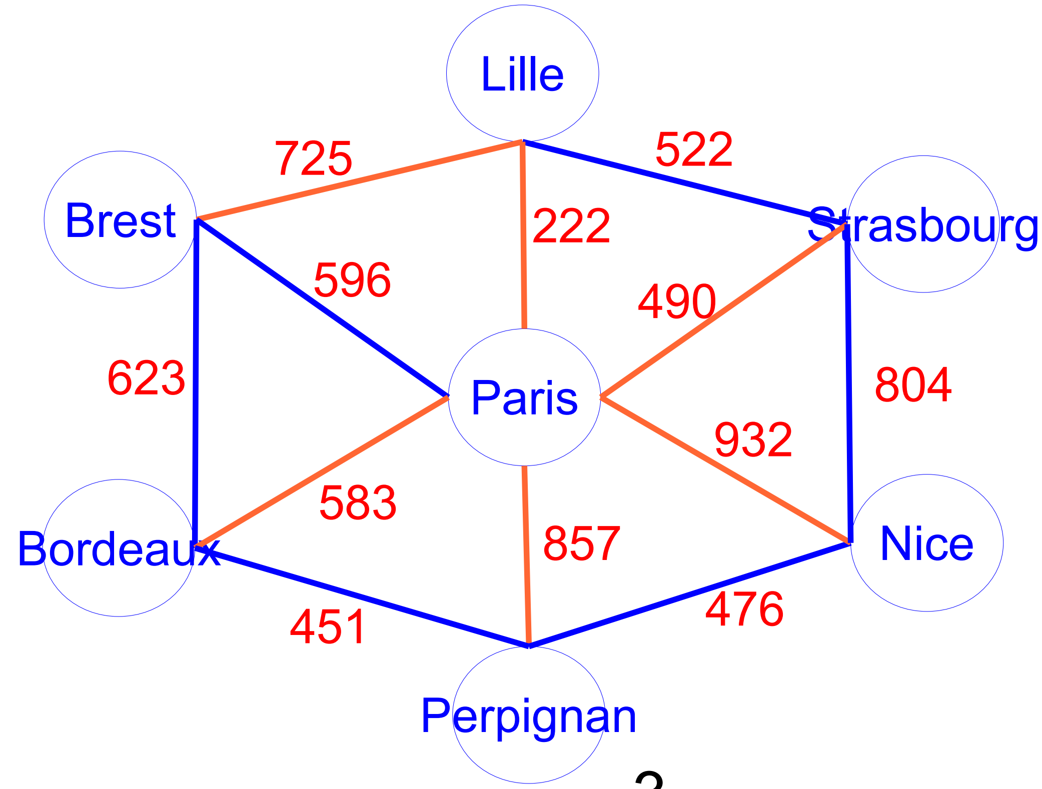
Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

v : E -> R*+

Exemple. Des villes et leurs distances



Problème 1. Construire un *réseau qui permet de connecter deux villes qcq*

Solution 1

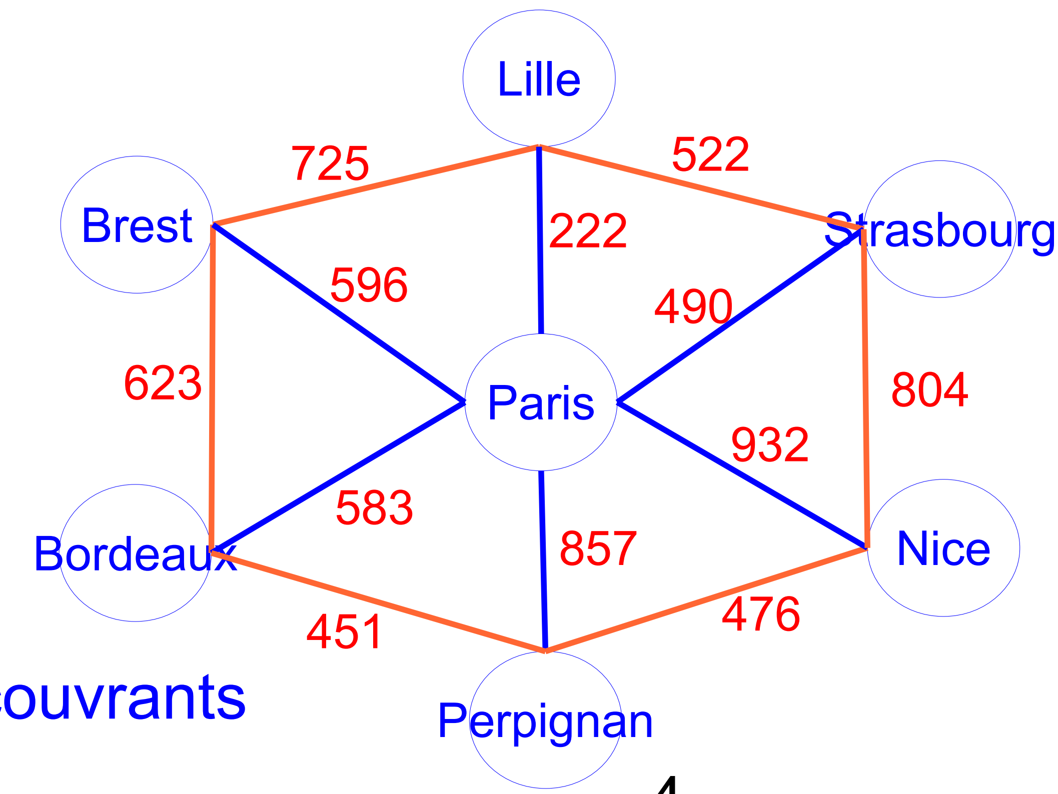
Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

v : E -> R*+

Exemple. Des villes et leurs distances



Problème 1. Construire un *réseau qui permet de connecter deux villes qcq*

Autres solutions = arbres couvrants

Théorie des Graphes

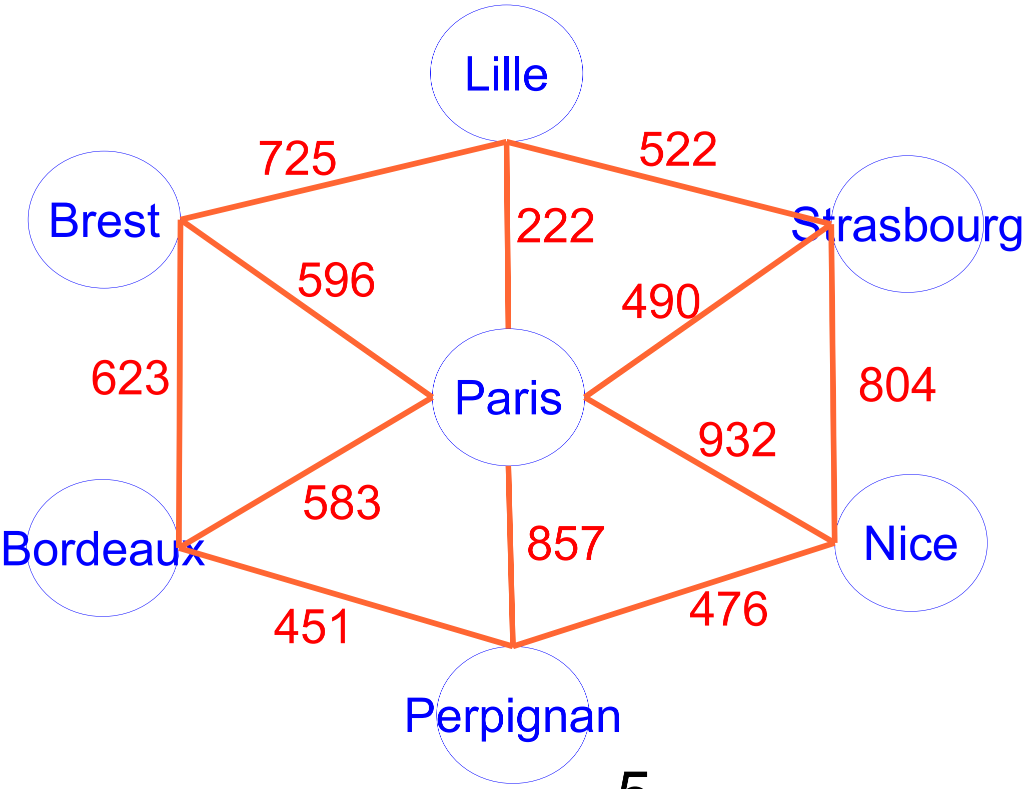
10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

$$v : E \rightarrow R^{*}+$$

Exemple. Des villes et leurs distances

Problème 2. Construire un tel réseau *de longueur minimum.*



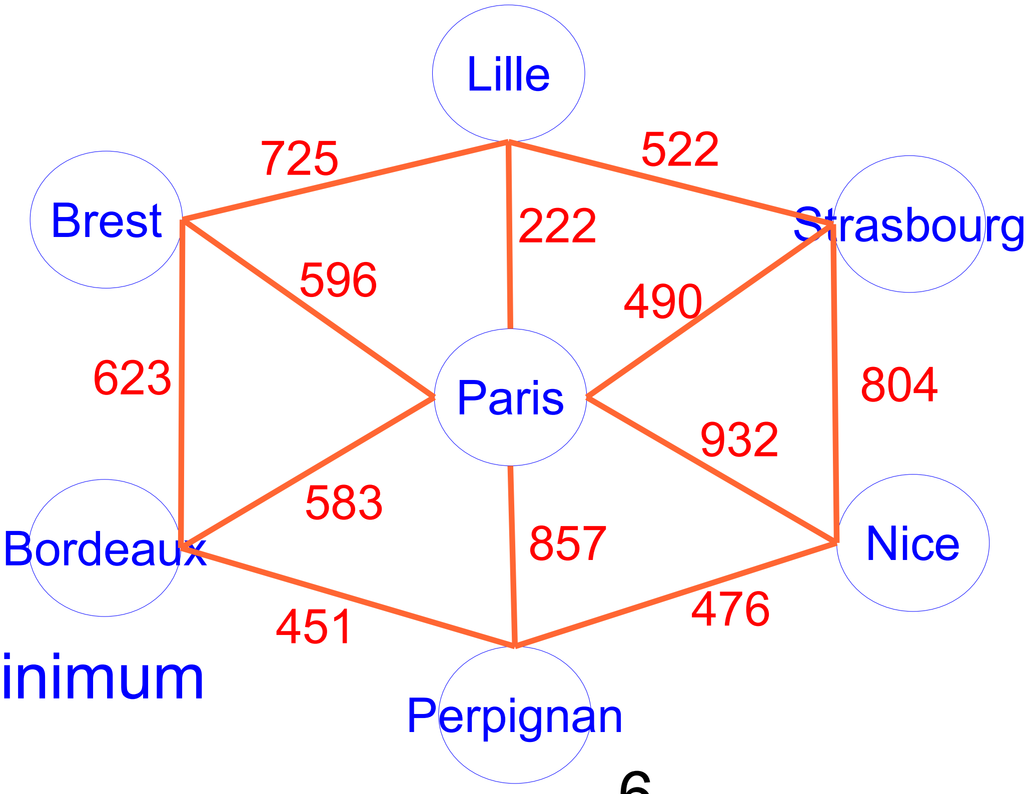
Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

$v : E \to R^{*+}$

Exemple. Des villes et leurs distances



Problème 2. Construire un tel réseau *de longueur minimum.*

Solution = arbre couvrant minimum

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

v : E -> R*+

Désignons par T(G) : l'ensemble des arbres couvrants

Pour chaque arbre couvrant T, notons v(T) sa valeur :

$$v(T) = \sum_{e \in T} v(e)$$

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

v : E -> R*+

Désignons par T(G) : l'ensemble des arbres couvrants

Pour chaque arbre couvrant T, notons v(T) sa valeur :

$$v(T) = \sum_{e \in T} v(e)$$

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Soit G=(X,E) un graphe simple valué par des valeurs strictement positives :

v : E -> R*+

Désignons par T(G) : l'ensemble des arbres couvrants

Pour chaque arbre couvrant T, notons v(T) sa valeur :

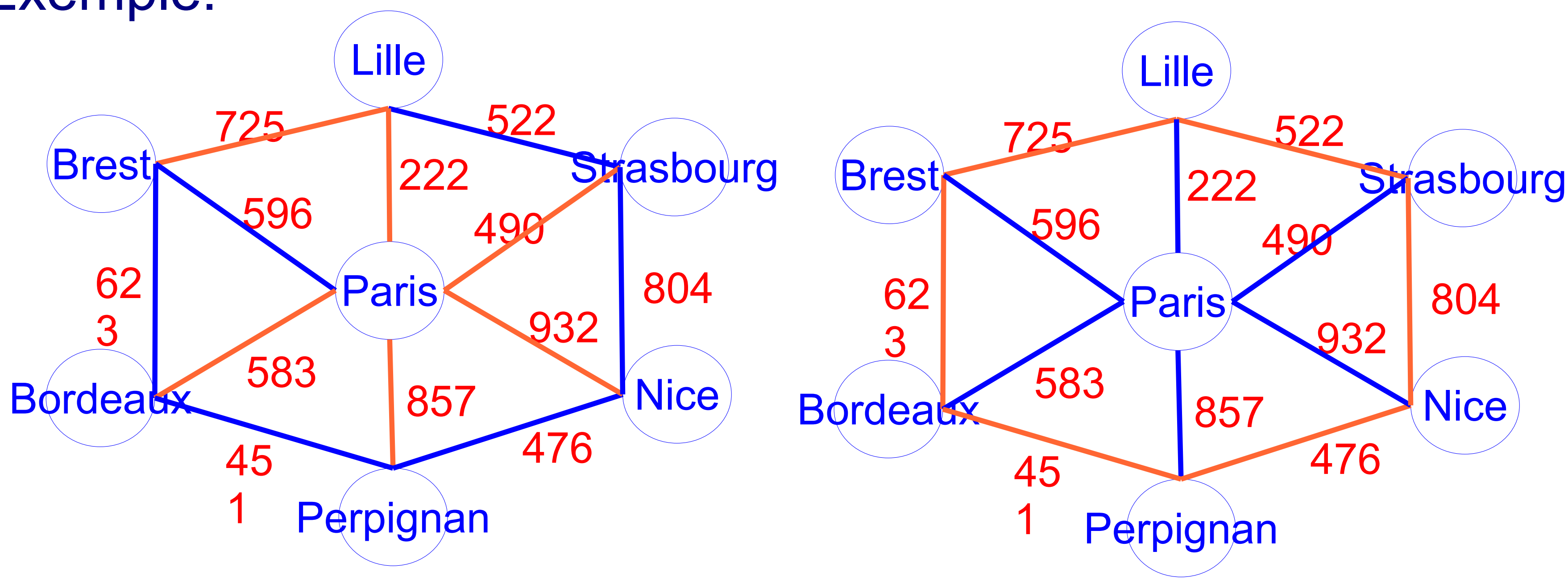
$$v(T) = \sum_{e \in T} v(e)$$

Les éléments de T(G) sont *ordonnés* par leurs valeurs.

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Exemple.

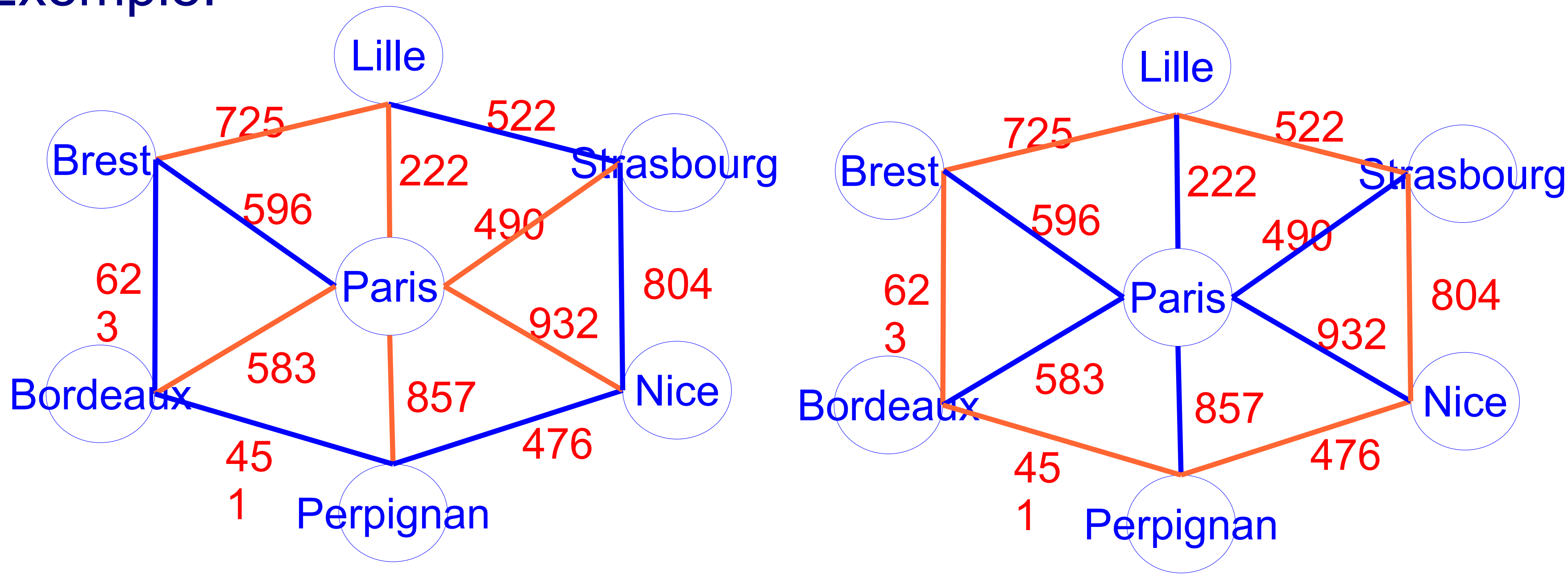


v (T1)=522+804+476+451+623+596
=3472

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Exemple.



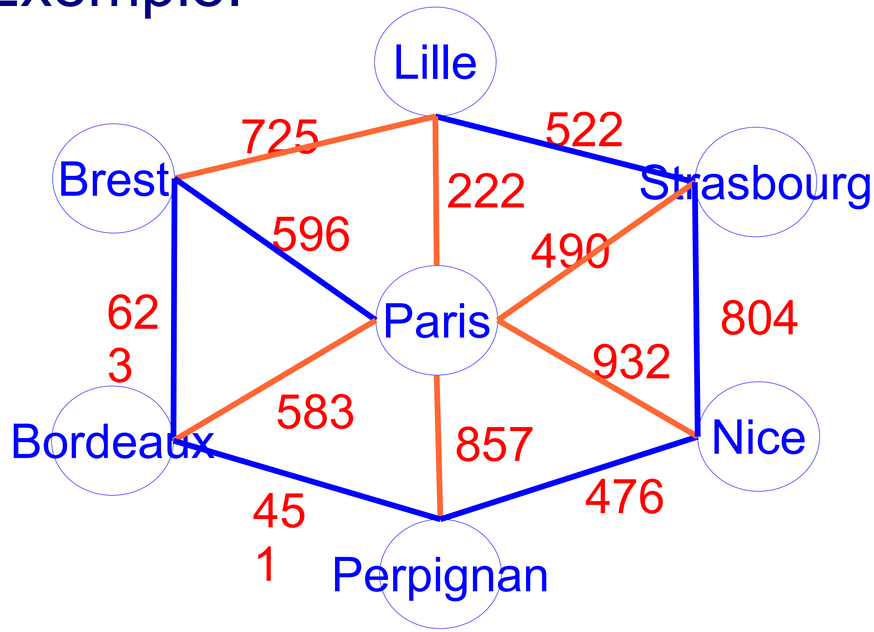
v (T1)=522+804+476+451+623+596
=3472

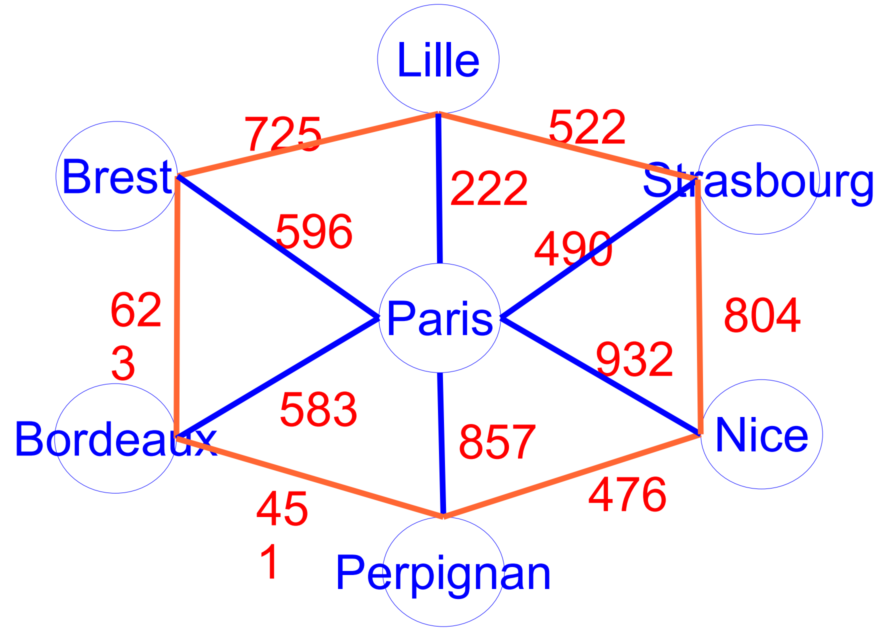
v (T2)=3680

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Exemple.





v (T1)=3472 v (T2)=3680

T1 est plus court: v (T1)<v (T2).

Quel est l'arbre couvrant le plus court?

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Deux arbres couvrants T et T' sont *voisins* s'il existe deux arêtes e et f t.q.

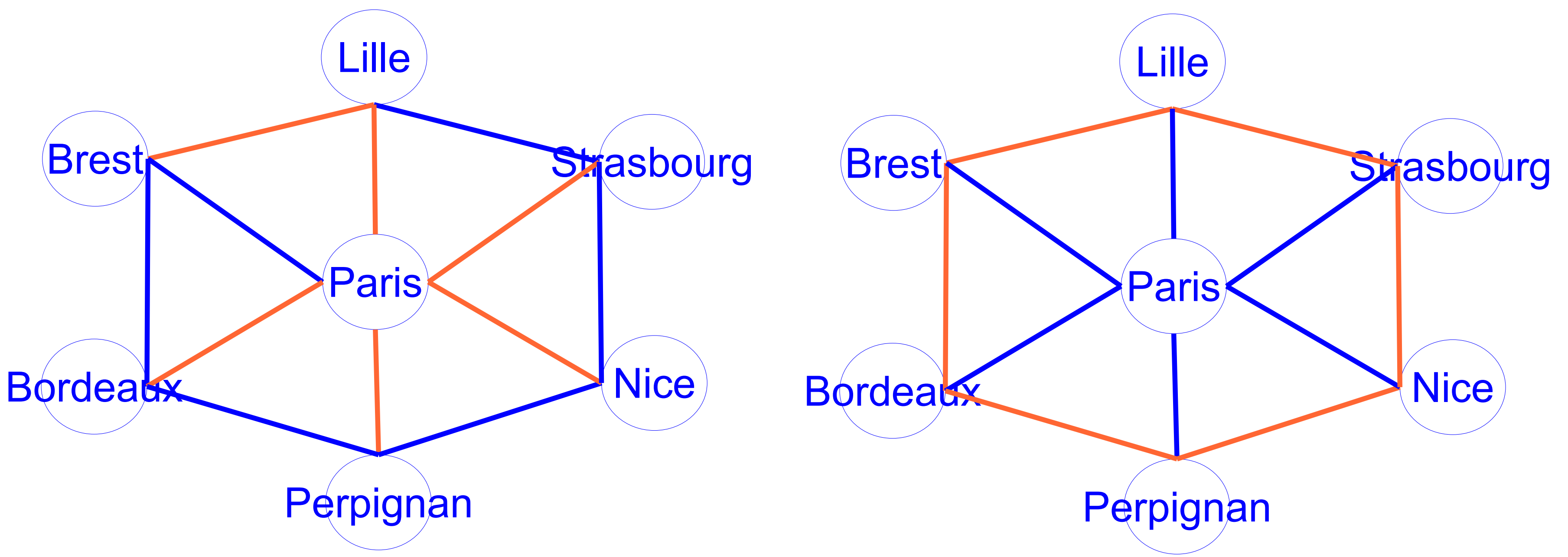
T'=T+e-f   et   T=T'+f-e

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Deux arbres couvrants T et T' sont *voisins* s'il existe deux arêtes e et f t.q.

$$T'=T+e-f \quad \text{et} \quad T=T'+f-e$$

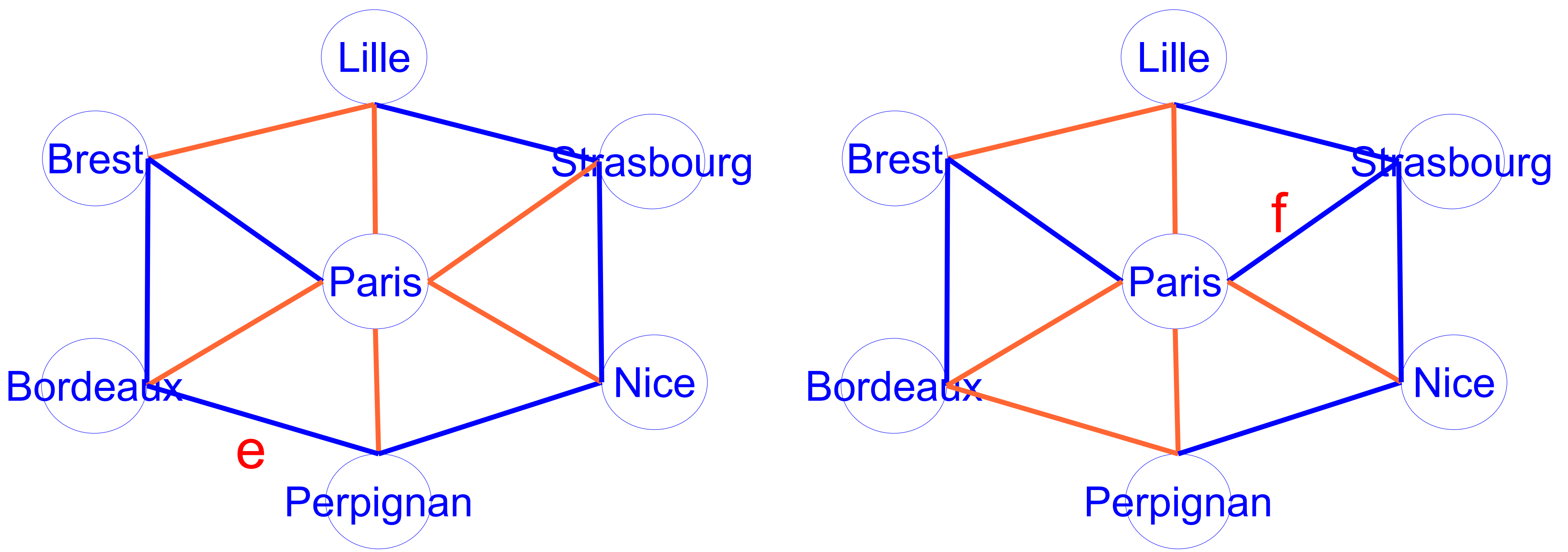


Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Deux arbres couvrants T et T' sont *voisins* s'il existe deux arêtes e et f t.q.

T'=T+e-f   et   T=T'+f-e



Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Deux arbres couvrants T et T' sont *voisins* s'il existe deux arêtes e et f t.q.

$$T' = T + e - f \quad \text{et} \quad T = T' + f - e$$

Dans T(G), T est un *minimum local* si pour tout T' voisin de T, on a

$$v(T) \leq v(T')$$

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Deux arbres couvrants T et T' sont *voisins* s'il existe deux arêtes e et f t.q.

$$T' = T + e - f \quad \text{et} \quad T = T' + f - e$$

Dans T(G), T est un *minimum local* si pour tout T' voisin de T, on a

$$v(T) \leq v(T')$$

Dans T(G), T est un *minimum absolu* si pour tout T', on a

$$v(T) \leq v(T')$$

Théorie des Graphes

10.4. Arbres couvrants dans un graphe valué

Deux arbres couvrants T et T' sont *voisins* s'il existe deux arêtes e et f t.q.

T'=T+e-f et T=T'+f-e

Dans T(G), T est un *minimum local* si pour tout T' voisin de T, on a

$$v(T) \leq v(T')$$

Dans T(G), T est un *minimum absolu* si pour tout T', on a

$$v(T) \leq v(T')$$

Théorème. Dans T(G), un *minimum local* est un *minimum absolu*

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe $G=(X,E)$ valué par l'applicaton v (dans R*+).

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton v (dans R*+).

Solution 1. (Recherche exhaustive)
Examiner tous les arbres couvrants.

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe $G=(X,E)$ valué par l'applicaton v (dans R^{*+}).

Solution 1. (Recherche exhaustive)
Examiner tous les arbres couvrants.

Quel est le nombre d'arbres couvrants à considérer?

Théorie des Graphes

10.5. Arbre couvrant minimum

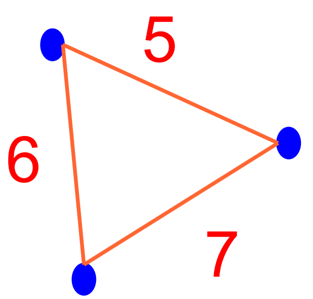
Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 1. (Recherche exhaustive)
Examiner tous les arbres couvrants.

Quel est le nombre d'arbres couvrants à considérer?

S'il s'agit d'un petit graphe:

K3 → 3



Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 1. (Recherche exhautive)
Examiner tous les arbres couvrants.

Quel est le nombre d'arbres couvrants à considérer?

S'il s'agit d'un petit graphe:

$K3 \rightarrow 3$
$K4 \rightarrow 16$

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 1. (Recherche exhautive)
Examiner tous les arbres couvrants.

Quel est le nombre d'arbres couvrants à considérer?

Un graphe un peu plus grand:

K20 $\rightarrow 20^{18}$

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 1. (Recherche exhaustive)
Examiner tous les arbres couvrants.

Quel est le nombre d'arbres couvrants à considérer?

Un graphe un peu plus grand:
K20 $\rightarrow$ 20^{18}

Si l'on peut examiner 10^9 arbres / s

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 1. (Recherche exhaustive)
Examiner tous les arbres couvrants.

Quel est le nombre d'arbres couvrants à considérer?

Un graphe un peu plus grand:
K20 $\rightarrow 20^{18}$

Il faut trouver une autre solution!

Si l'on peut examiner 10^9 arbres / s
Il faut $2^{18} \times 10^9$ secondes – plus de 8000 siècles !!!

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 2. (*Algorithme glouton*)

- On part d'un ensemble vide A.
- A chaque étape, on ajoute dans A l'arête de valeur la plus petite possible t.q. G(A) soit acyclique.
- L'algo se termine si l'on ne peut plus ajouter d'arête dans A.

Théorie des Graphes

10.5. Arbre couvrant minimum

Problème. Trouver un arbre couvrant minimum dans un graphe simple connexe G=(X,E) valué par l'applicaton *v* (dans R*+).

Solution 2. (*Algorithme glouton*)

- On part d'un ensemble vide A.
- A chaque étape, on ajoute dans A l'arête de valeur la plus petite possible t.q. G(A) soit acyclique.
- L'algo se termine si l'on ne peut plus ajouter d'arête dans A.

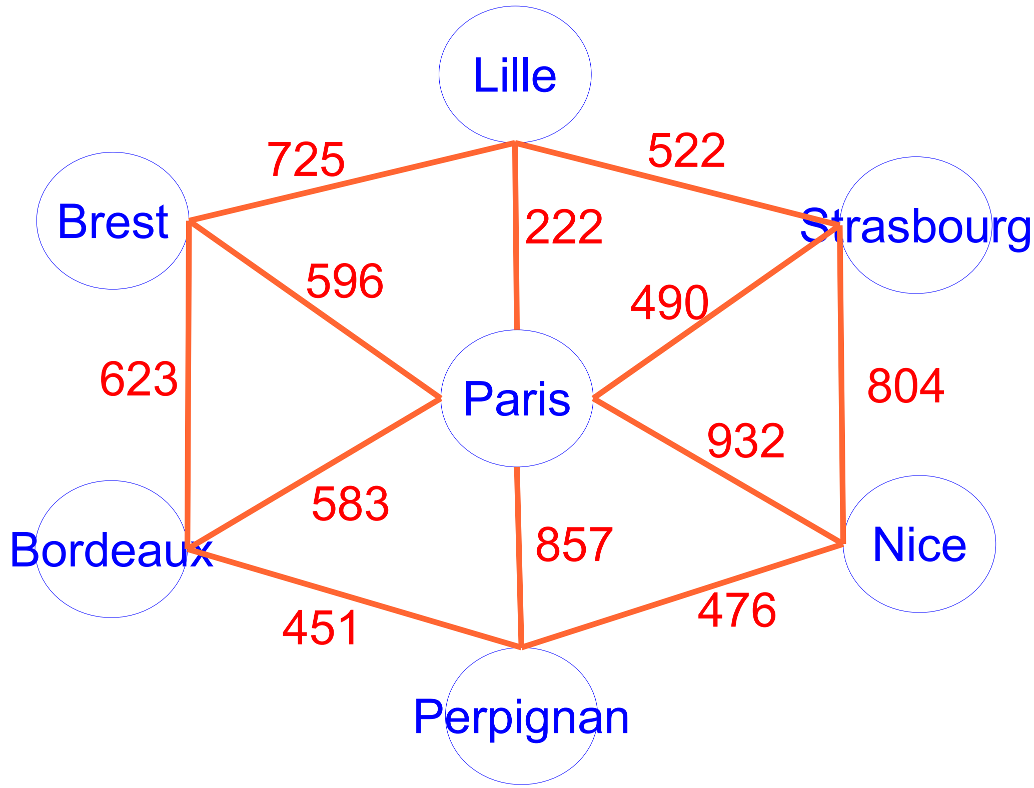
Questions.

- L'algo se termine-t-il?
- Le sous graphe ainsi obtenu est-il un arbre couvrant minimum?

Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

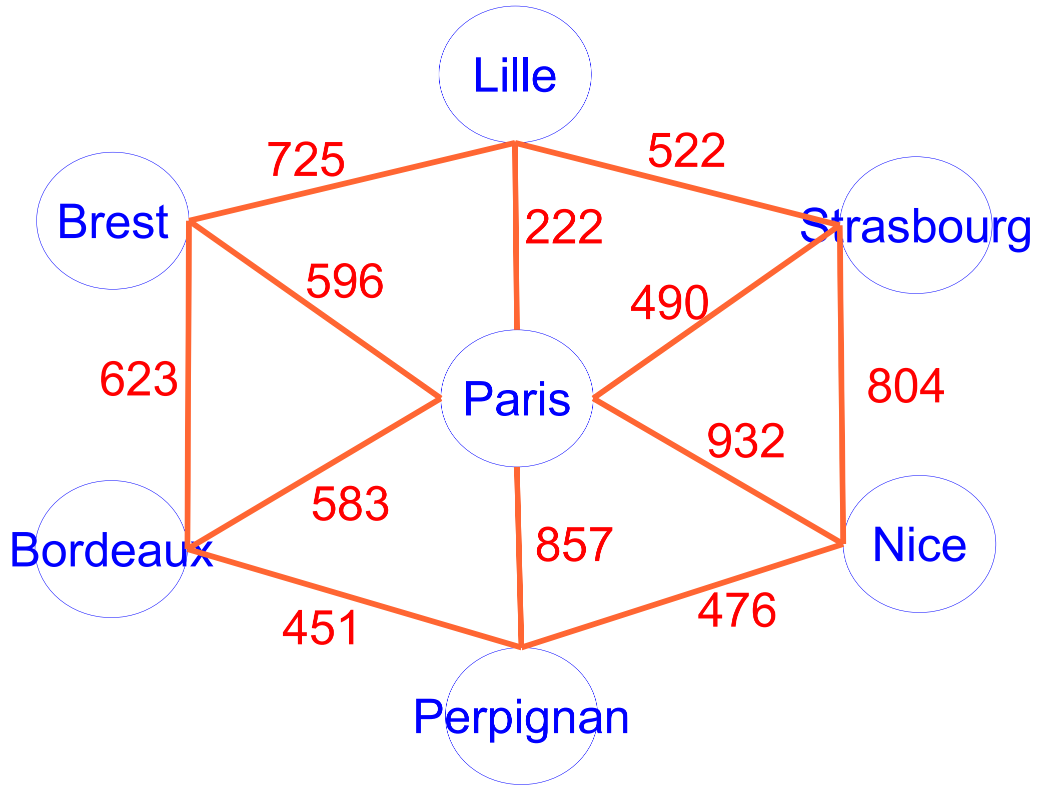


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide

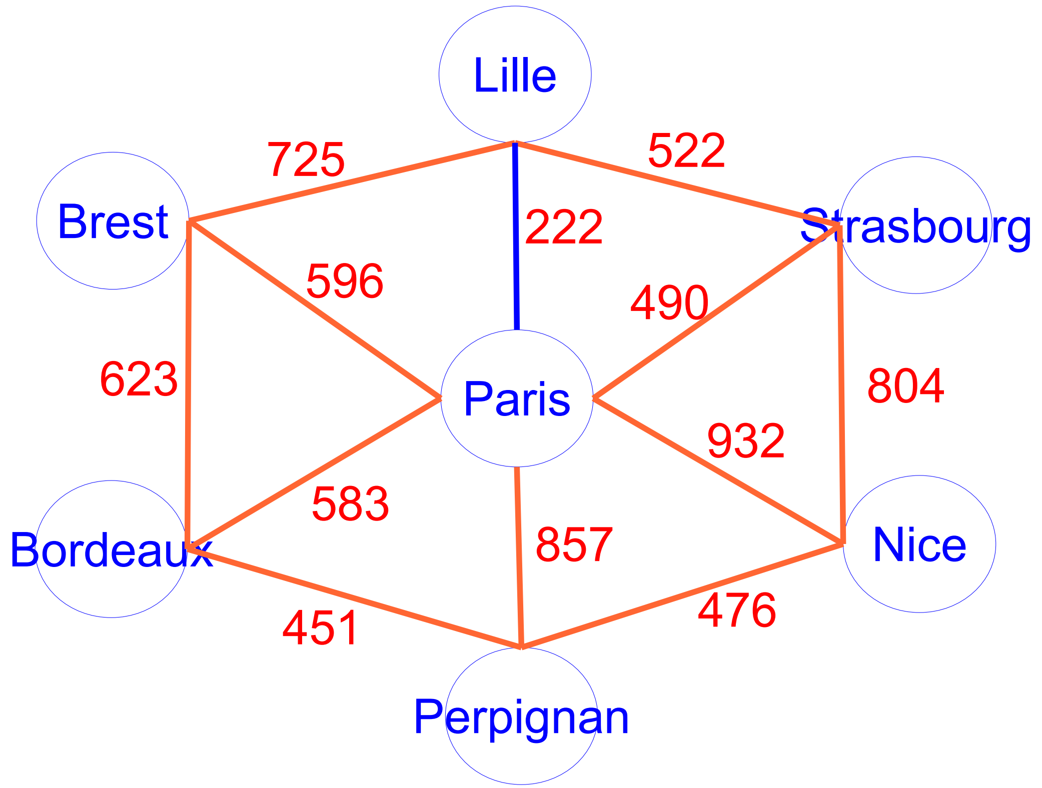


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide
2. ajouter 222

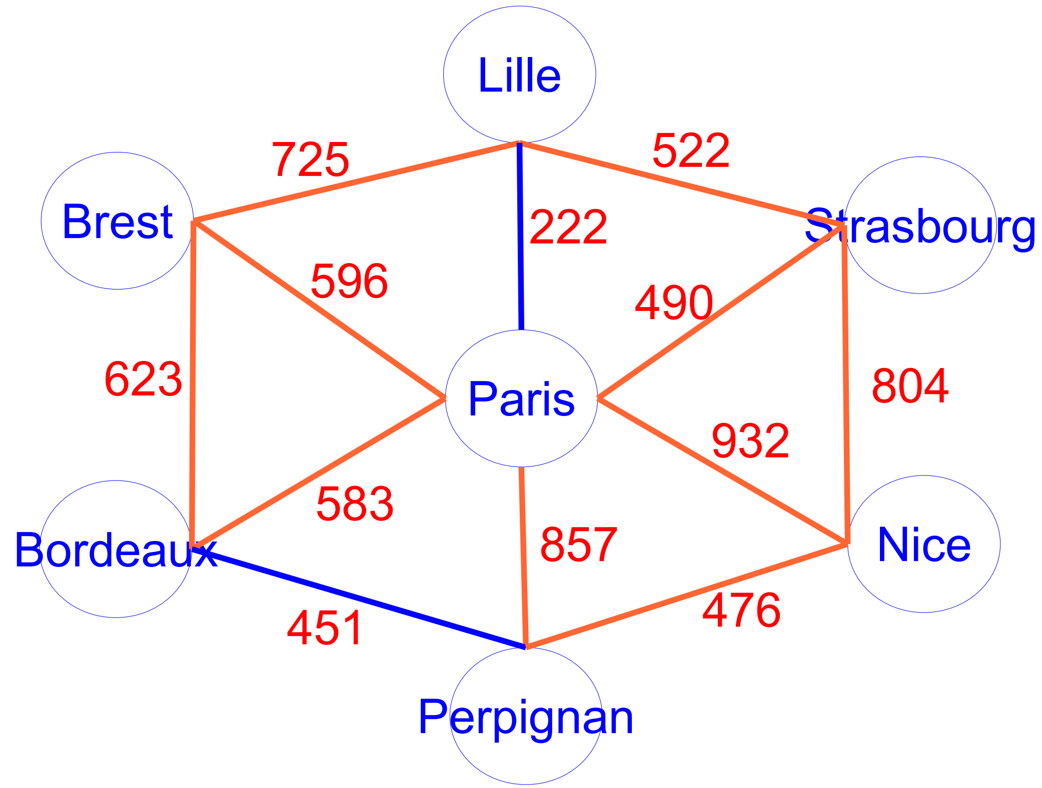


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide
2. ajouter 222
3. ajouter 451

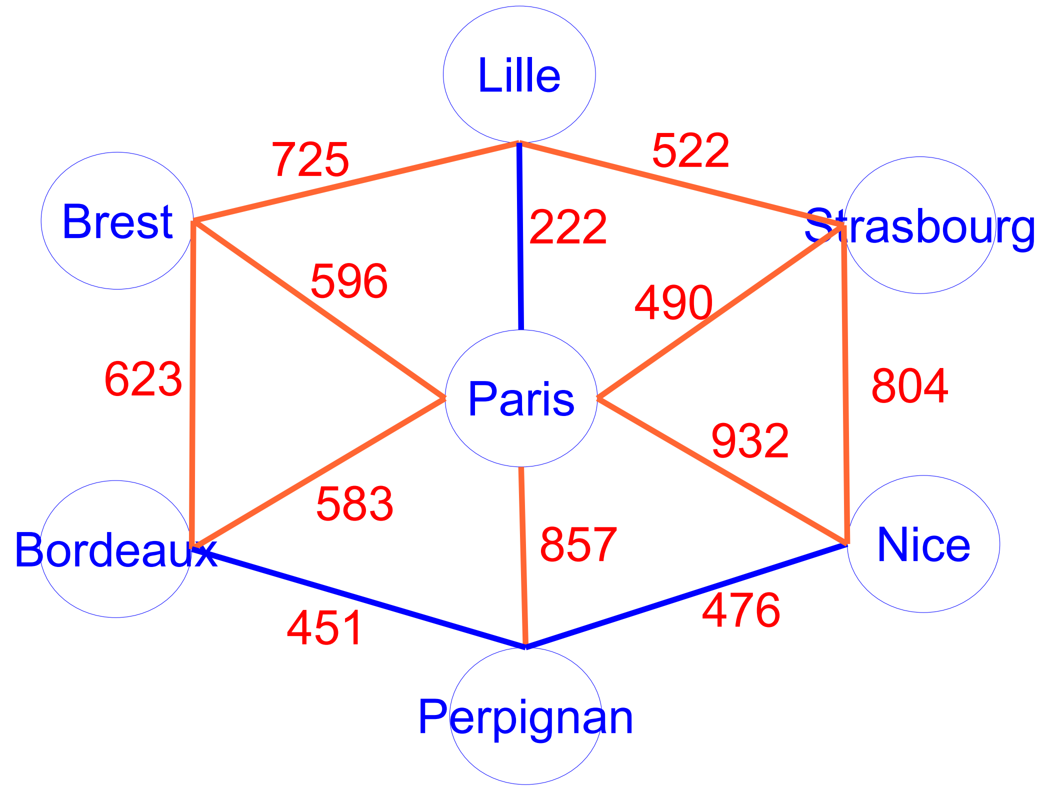


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide
2. ajouter 222
3. ajouter 451
4. ajouter 476

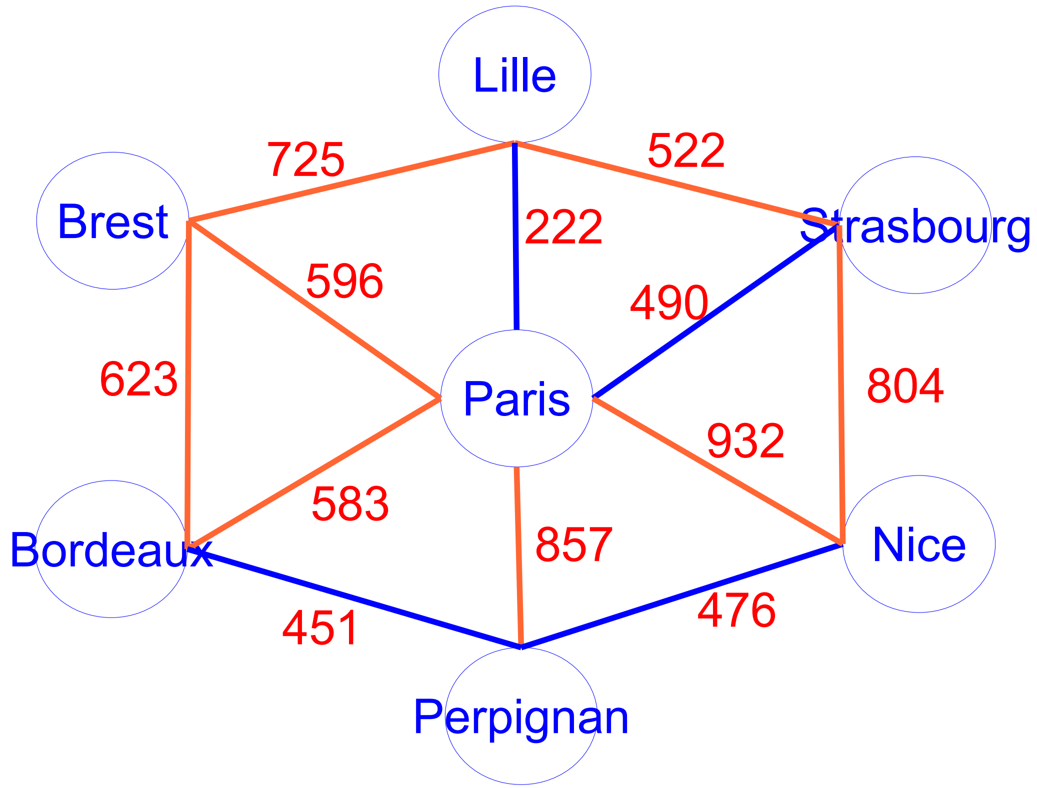


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide
2. ajouter 222
3. ajouter 451
4. ajouter 476
5. ajouter 490

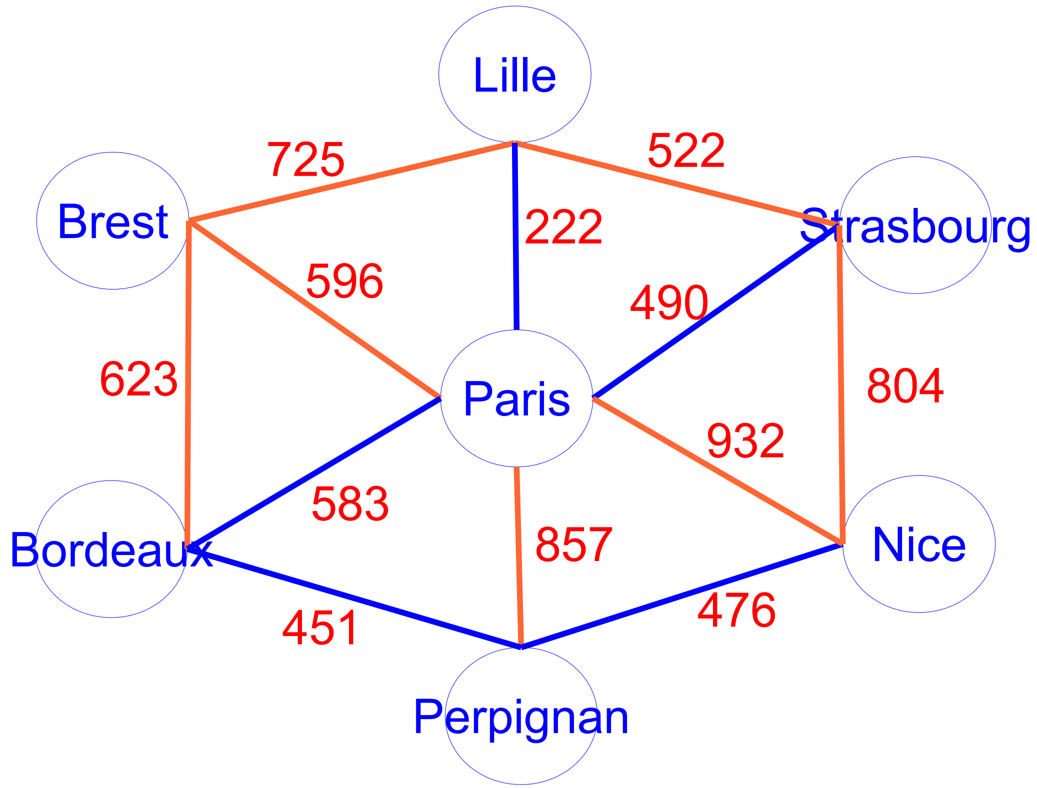


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide
2. ajouter 222
3. ajouter 451
4. ajouter 476
5. ajouter 490
6. ajouter 583

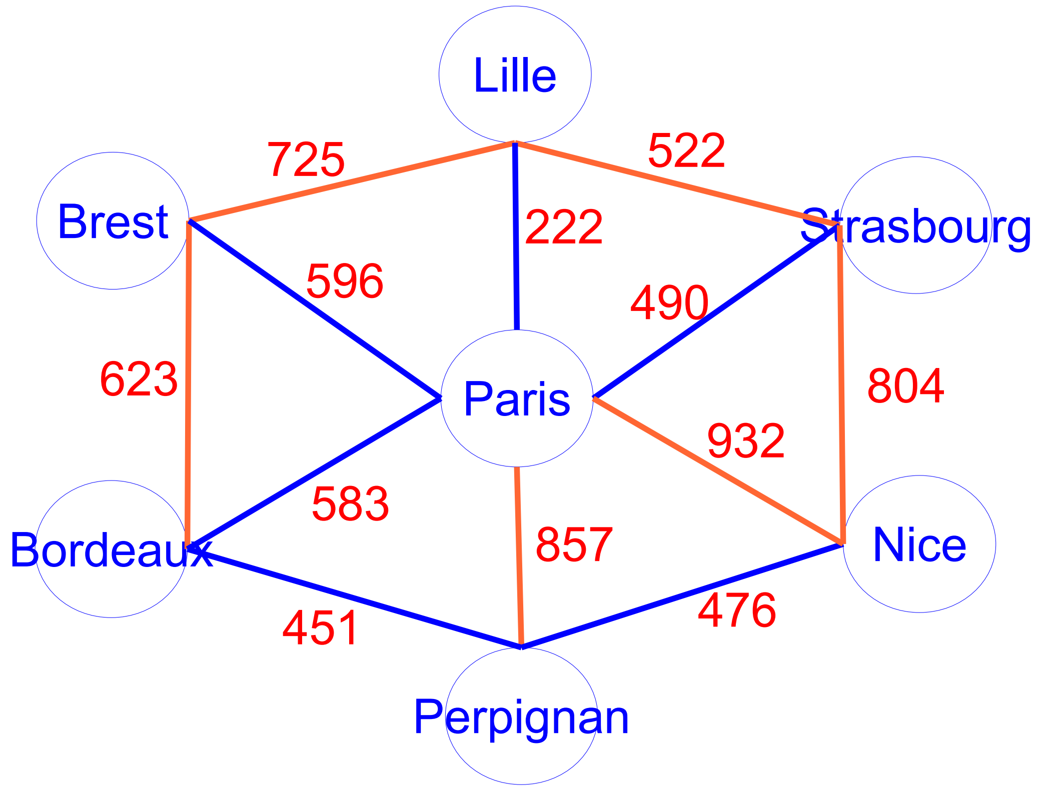


Théorie des Graphes

10.5. Arbre couvrant minimum

Exemple.

1. A vide
2. ajouter 222
3. ajouter 451
4. ajouter 476
5. ajouter 490
6. ajouter 583
7. ajouter 596



Théorie des Graphes

10.5. Arbre couvrant minimum

Théorème. Pour un graphe simple connexe valué, l'algorithme glouton se termine et donne l'arbre couvrant minimum.

Preuve. (indication) Montrons que le graphe obtenu est un arbre couvrant minimum local.

Théorie des Graphes

10.5. Arbre couvrant minimum

Algorithme de Kruskal (réalisant l'algorithme glouton).

procedure Kruskal (G, v);
begin
    F := E; A:=∅;
    while |A| < n-1 **loop**
        trouver e dans F tel que v(e) soit minimum
        F := F – {e};
        if G(A+{e}) acyclique **then**
            A := A +{e};
        end if;
    end loop;
end Kruskal;

Théorie des Graphes

10.5. Arbre couvrant minimum

Algorithme de Kruskal (réalisant l'algorithme glouton).

```
procedure Kruskal (G, v);
begin
    F := E; A:=Ø;
    while  |A| < n-1 loop
        trouver e dans F tel que v(e) soit minimum
        F := F – {e};
        if G(A+{e}) acyclique then
            A := A +{e};
        end if;
    end loop;
end Kruskal;
```

F: les arêtes restant à considérer

Ajouter l'arête de valeur la plus petite dans A t.q. G(A) soit acyclique

Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

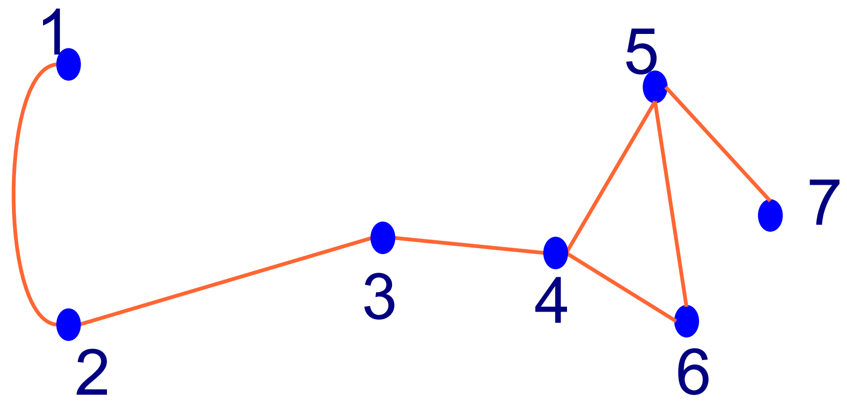
Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Exemple.



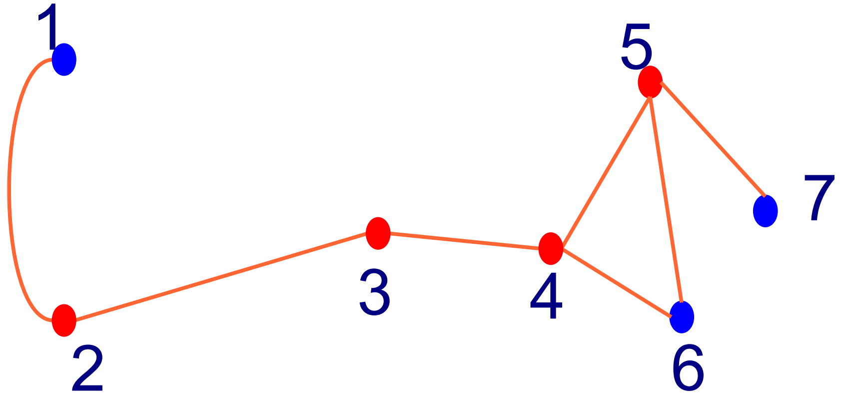
Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Exemple.



Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Un *bloc* est *un sous graphe engendré connexe et sans points d'articulation, maximal avec cette propriété*.

Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Un *bloc* est *un sous graphe engendré connexe et sans points d'articulation, maximal avec cette propriété*.

engendré par un ensemble d'arêtes

on ne peut pas ajouter d'arêtes
pour obtenir un sous graphe
strictement plus large qui est
connexe, sans points d'articulation

Théorie des Graphes

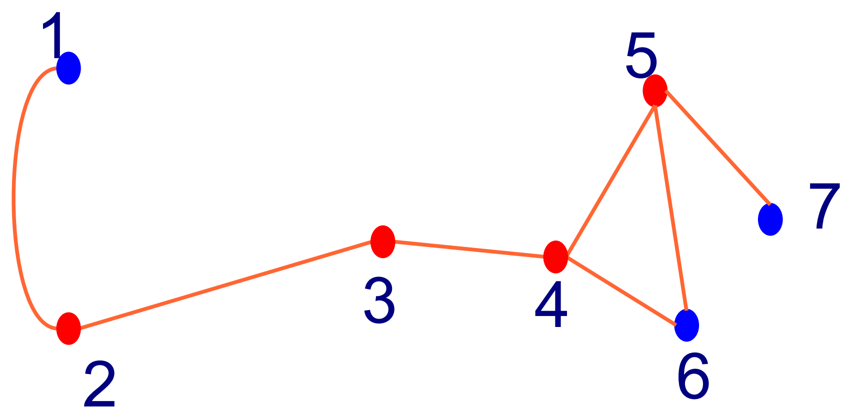
10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Un *bloc* est *un sous graphe engendré connexe et sans points d'articulation, maximal avec cette propriété*.

Exemple.



Théorie des Graphes

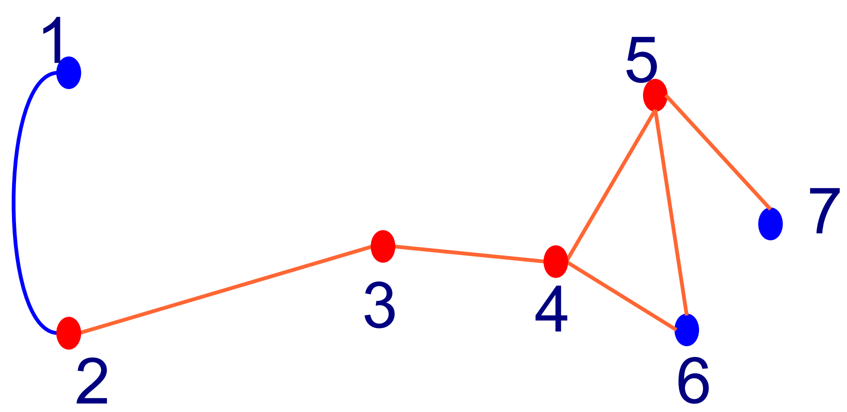
10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Un *bloc* est *un sous graphe engendré connexe et sans points d'articulation, maximal avec cette propriété*.

Exemple.



Théorie des Graphes

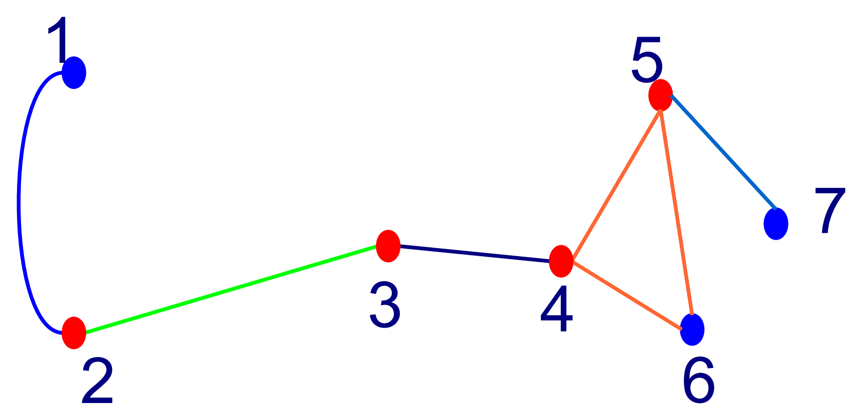
10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Un *bloc* est *un sous graphe engendré connexe et sans points d'articulation, maximal avec cette propriété*.

Exemple.



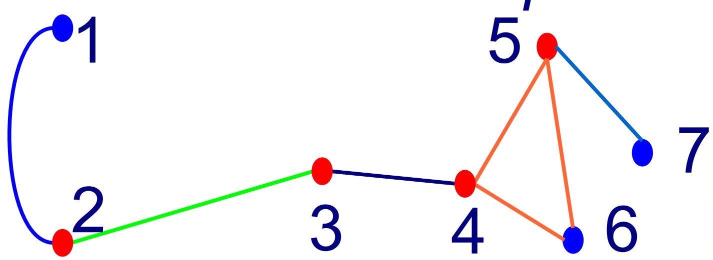
Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

Un sommet x est un *point d'articulation* si G-x a au moins une composante connexe de plus que G.

Un *bloc* est *un sous graphe engendré connexe et sans points d'articulation, maximal avec cette propriété*.



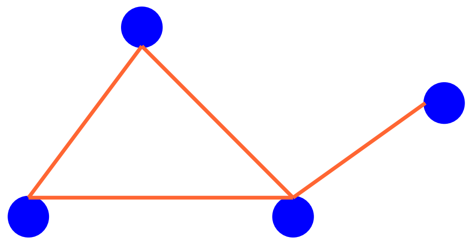
Propriétés.

- Les blocs = une partition de E(G)
- Deux blocs n'ont en commun qu'au plus un sommet, qui est un point d'articulation de G.
- Tout point d'articulation est un sommet commun à au moins deux blocs.

Théorie des Graphes

10.6. Connectivité

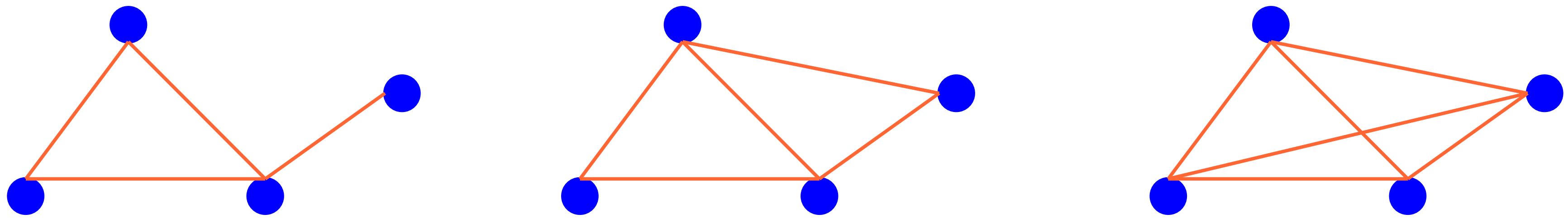
Question. Quel est le plus petit nombre de sommets qu'il faut retirer pour rendre un graphe non connexe?



Théorie des Graphes

10.6. Connectivité

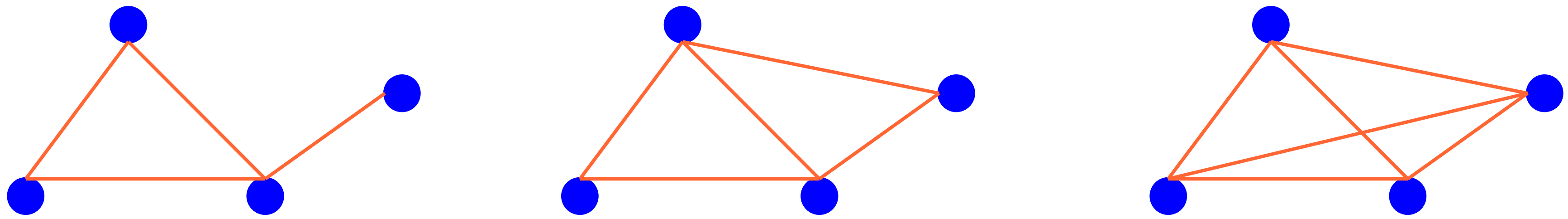
Question. Quel est le plus petit nombre de sommets qu'il faut retirer pour rendre un graphe non connexe?



Théorie des Graphes

10.6. Connectivité

Question. Quel est le plus petit nombre de sommets qu'il faut retirer pour rendre un graphe non connexe?

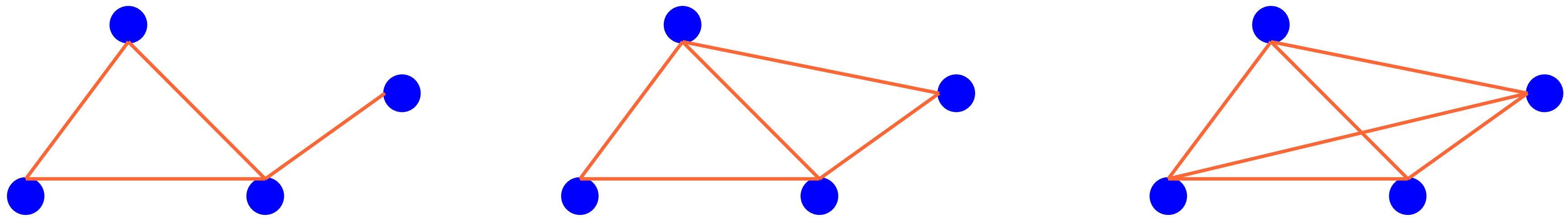


Le *nombre de connexité* $\kappa(G)$ = le plus petit nombre de sommets qu'il faut retirer pour obtenir un graphe non connexe ou réduit à un point isolé.

Théorie des Graphes

10.6. Connectivité

Question. Quel est le plus petit nombre de sommets qu'il faut retirer pour rendre un graphe non connexe?



Le *nombre de connexité* $\kappa(G)$ = le plus petit nombre de sommets qu'il faut retirer pour obtenir un graphe non connexe ou réduit à un point isolé.

On a $$0 \leq \kappa(G) \leq n-1$$

De plus, $$\kappa(G) \leq \delta(G)$$

Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

G est *k-connexe* si $\kappa(G) \geq k$

Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

G est *k-connexe* si $\kappa(G) \geq k$

Exemple.

- Tout graphe est *0-connexe*
- *1-connexe* = connexe et n>1
- *2-connexe* = connexe, sans points d'articulation et n>2

Théorie des Graphes

10.6. Connectivité

Soit un graphe simple G.

G est *k-connexe* si $\kappa(G) \geq k$

Exemple.

- Tout graphe est *0-connexe*
- *1-connexe* = connexe et n>1
- *2-connexe* = connexe, sans points d'articulation et n>2

Propriété.

Si *k'>k* alors *k'-connexe* implique *k-connexe*
(Par exemple, *2-connexe* implique *1-connexe*)

Théorie des Graphes

10.6. Connectivité

Théorème Menger (version sommets).

Soit G un graphe simple à n sommets, $n>k$.

G est *k-connexe* ssi deux sommets distincts qcq de G sont reliés par k chaînes élémentaires deux à deux *sommets-disjointes*.

pas de sommets communs
autre que les extrémités

Théorie des Graphes

10.6. Connectivité

Théorème Menger (version sommets).

Soit G un graphe simple à n sommets, n>k.

G est *k-connexe* ssi deux sommets distincts qcq de G sont reliés par k chaînes élémentaires deux à deux *sommets-disjointes*.

Théorème Menger (version arêtes).

Soit G un graphe simple à n sommets, n>k.

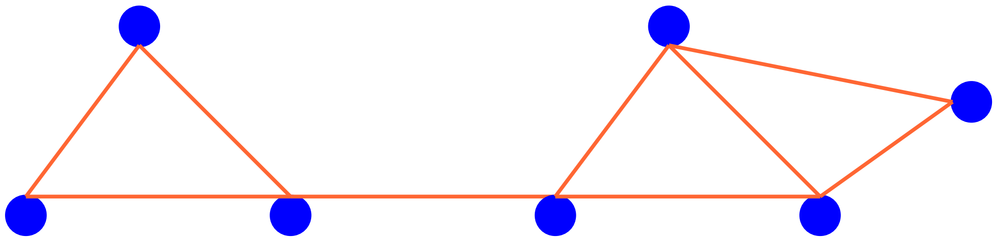
G est *k-arête-connexe* ssi deux sommets distincts qcq de G sont reliés par k chaînes élémentaires deux à deux *arêtes-disjointes*.

Théorie des Graphes

10.6. Connectivité

Arête-connexité. Soit un graphe G non réduit à un sommet.

Le nombre d'*arête-connexité* $\kappa'(G)$ de G = le plus petit nombre d'arêtes dont la suppression rend le graphe non connexe.

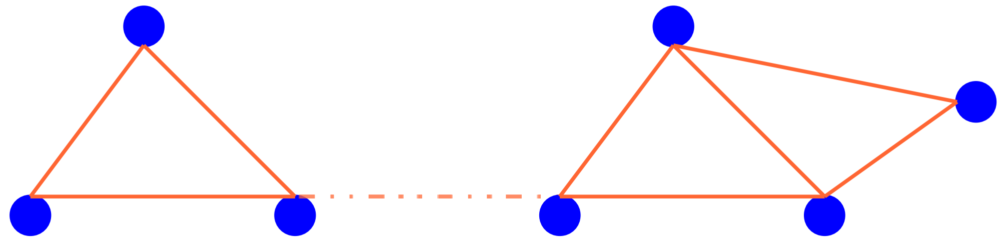


Théorie des Graphes

10.6. Connectivité

Arête-connexité. Soit un graphe G non réduit à un sommet.

Le nombre d'*arête-connexité* $\kappa'(G)$ de G = le plus petit nombre d'arêtes dont la suppression rend le graphe non connexe.

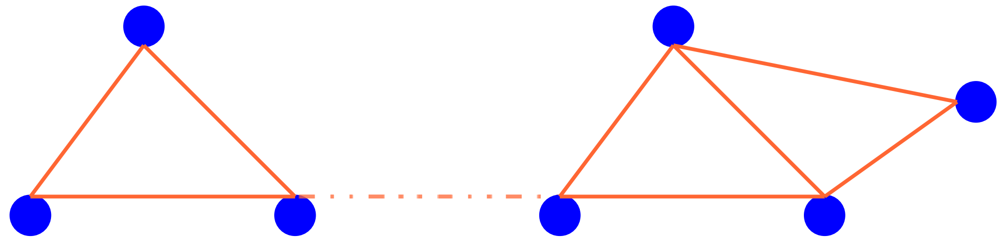


Théorie des Graphes

10.6. Connectivité

Arête-connexité. Soit un graphe G non réduit à un sommet.

Le nombre d'*arête-connexité* $\kappa'(G)$ de G = le plus petit nombre d'arêtes dont la suppression rend le graphe non connexe.



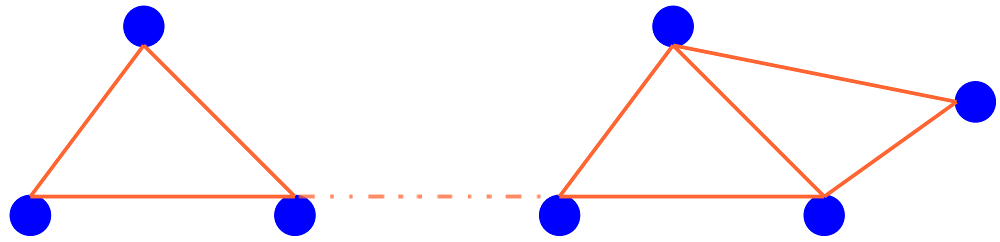
G est dit *k-arête-connexe* si $\kappa'(G) \geq k$

Théorie des Graphes

10.6. Connectivité

Arête-connexité. Soit un graphe G non réduit à un sommet.

Le nombre d'*arête-connexité* $\kappa'(G)$ de G = le plus petit nombre d'arêtes dont la suppression rend le graphe non connexe.



G est dit *k-arête-connexe* si $\kappa'(G) \geq k$

Proposition. Pour tout graphe simple G non réduit à un sommet,

$$\kappa(G) \leq \kappa'(G) \leq \delta(G)$$