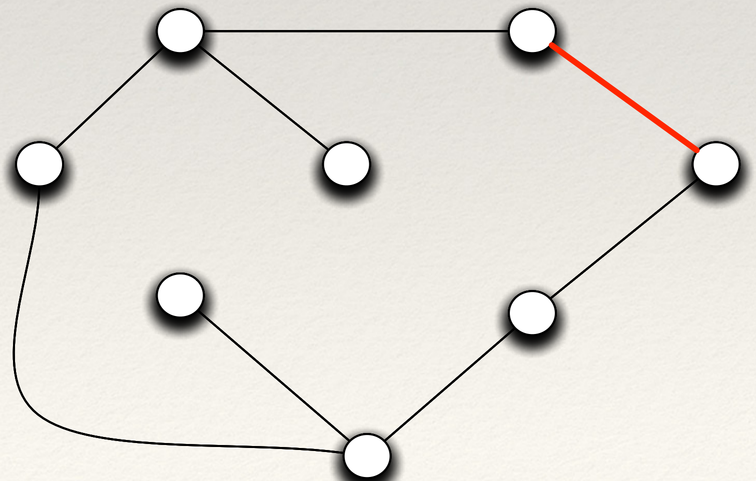
Lélia Blin

Algorithme distribué d'arbres couvrants de poids minimum

Algorithmique répartie
M1 Université d'Evry

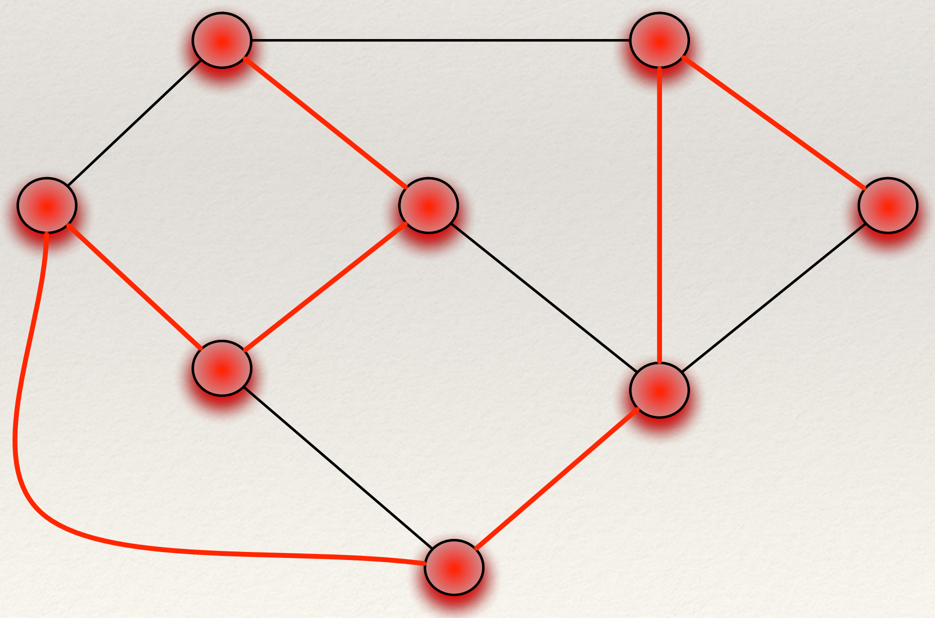
Définitions d'un arbre

- Pour un arbre T a n sommets il y a équivalence entre les définitions suivantes :
  - T est un arbre
  - T est un graphe connexe à n-1 arêtes
  - T est un graphe acyclique à n-1 arêtes
  - T est un graphe connexe et la suppression de toute arête le déconnecte
  - T est un graphe acyclique et l'ajout de toute arête le rend cyclique.



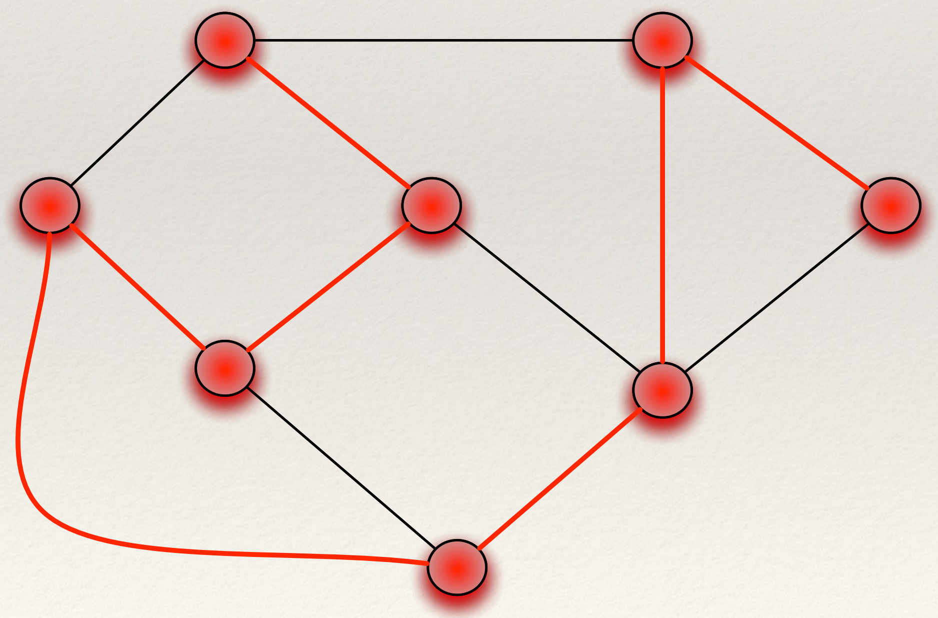
Graphe partiel

- Un graphe partiel G'(V,E') d'un graphe G(V,E) est:
  - Un graphe qui a les mêmes sommets que G.
  - Un graphe dont l'ensemble des arêtes E' est inclus dans E.



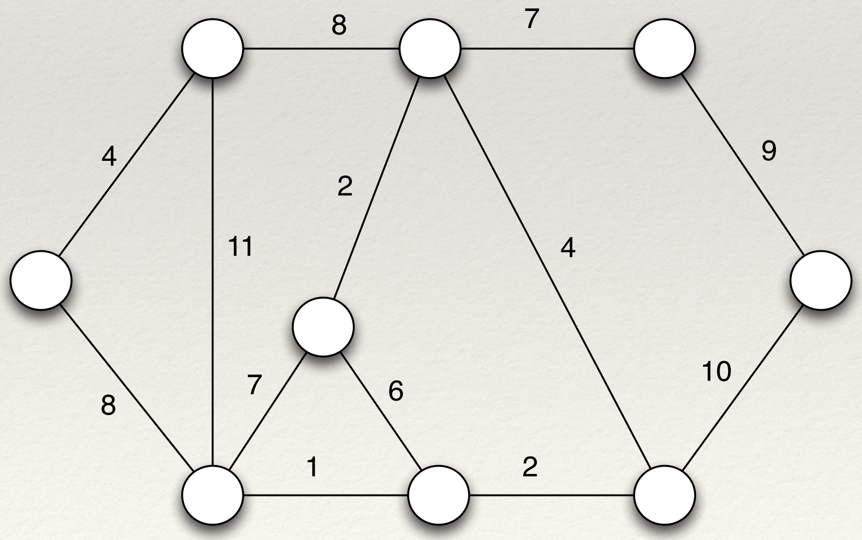
Arbres couvrants

- Un arbre couvrant T d'un graphe G(V,E) est:
  - Un graphe partiel, sans cycle.



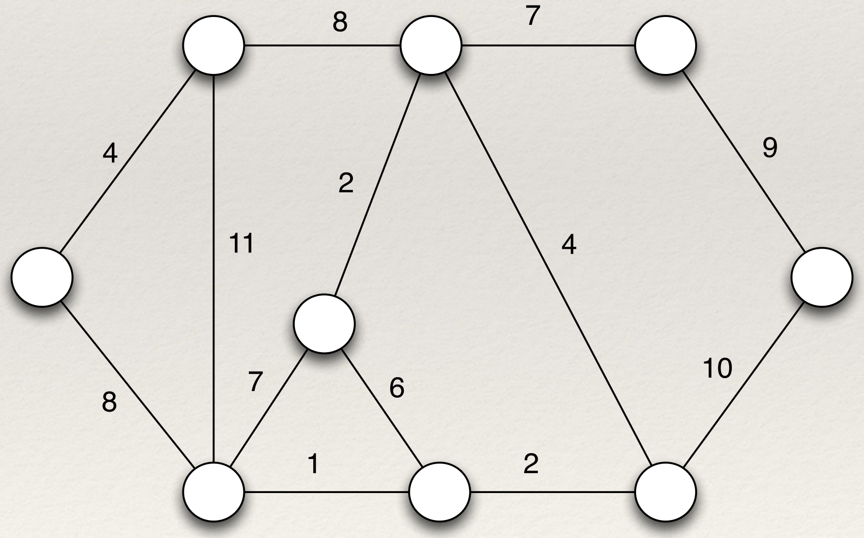
Graphe pondéré

- Un graphe pondéré G(V,E,ω) est un graphe où un entier positif est affecté à chaque arête.
- On appelle cet entier poids de l'arête.



Poids d'un graphe

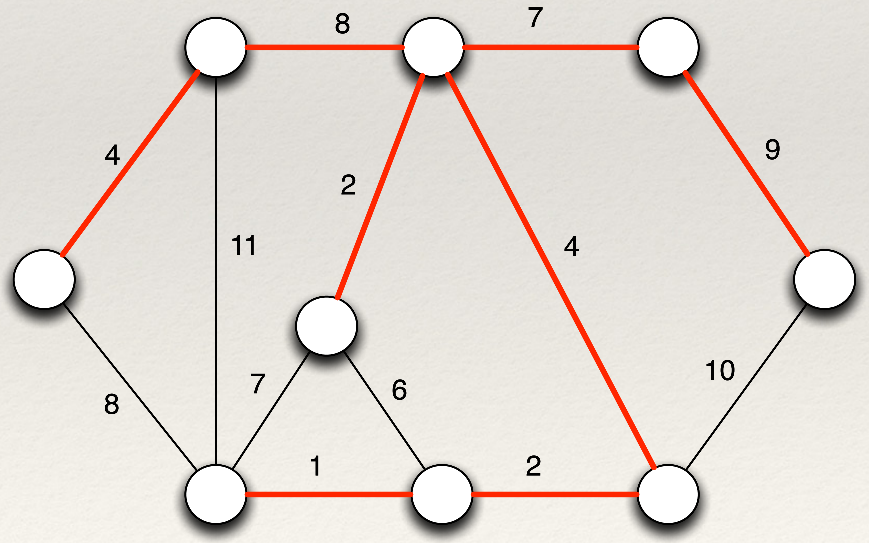
- Le poids (où coût) d'un graphe est la somme des poids des arêtes du graphe.
- On le note ω(G)



ω(G)=103

Arbre couvrant de poids minimum

- Soit un graphe $G=(V,E,\omega)$ un graphe non orienté pondéré.
- On appelle arbre couvrant de poids minimum (ou maximum) de G
    - noté ACPM ou MST (minimum Spanning Tree)
    - Tout arbre couvrant dont la somme des poids des arêtes le constituant est minimal (maximal).



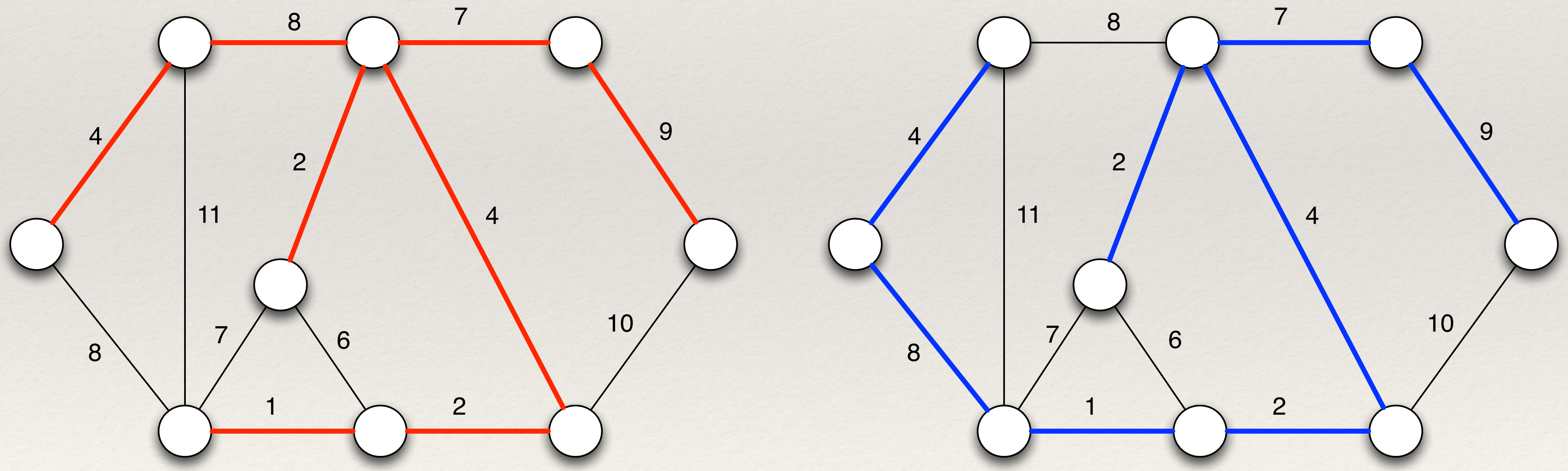
$\omega(G')=37$

Proposition

- Un graphe admet un arbre couvrant si, et seulement si, il est connexe.

Remarque 1

- L'arbre couvrant de poids minimal n'est pas forcément unique.



Remarque 2

- Un arbre couvrant de poids minimum est unique si et seulement si les poids de ces arêtes sont deux à deux distincts.

Propriété Cycle-Max

- L'arête de poids maximum d'un cycle ne fait partie d'aucun arbre couvrant de poids minimum.

Propriété Coupe-Min

- L'arête de poids minimum d'une coupe fait partie de l'arbre couvrant de poids minimum.

Algorithmes séquentiels

- Algorithme de Borůkva (1926) (Coupe-Min)
- Algorithme de Prim (1957) (Coupe-Min)
- Algorithme de Kruskal (1956) (Cycle-Max)
- Algorithme de Solin (1961) (Coupe-Min)
- …

Question

- Comment construire de façon distribué un arbre couvrant de poids minimum?

Kruskal distribué?

- L'approche de Kruskal est-elle distribuable?

Prim distribué?

- L'approche de Prim est-elle distribuable?

Blin Lélia

Algorithme de Borůvka

Arbre couvrant de poids minimum

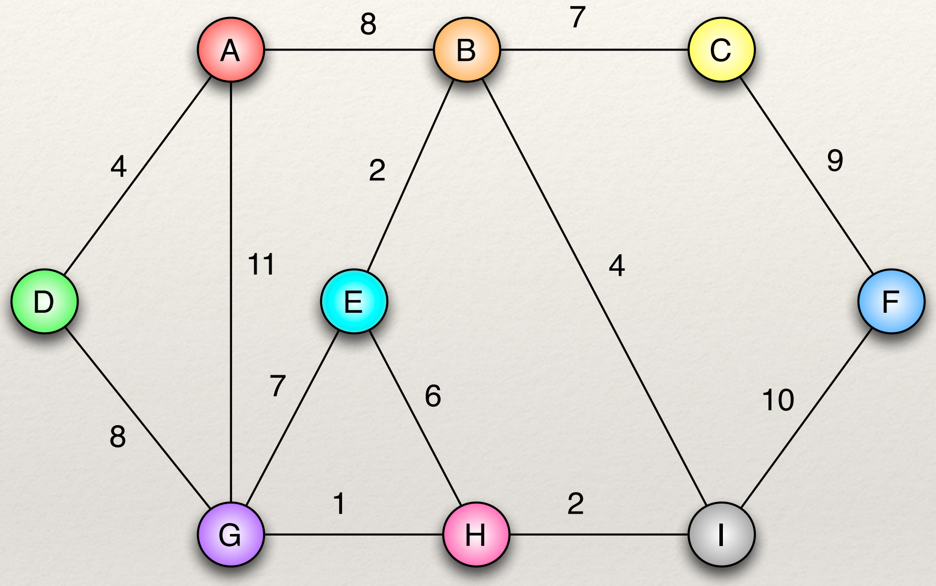
Algorithmique répartie

M1 Université d'Evry

Algorithme de Borůkva

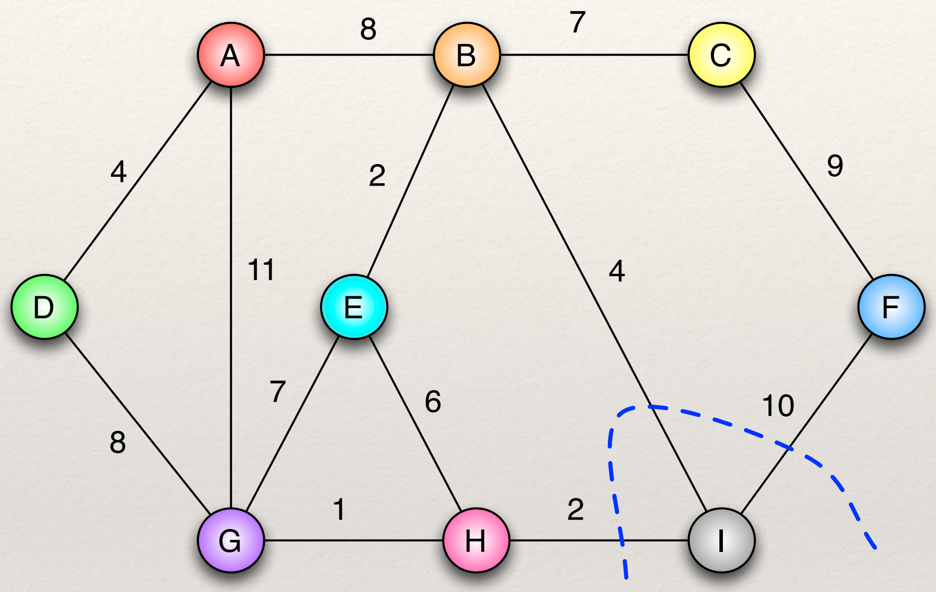
- Initialement chaque sommet est une composante connexe
- Tant qu'il existe plusieurs composantes connexe:
  - Choisir une composante connexe C
  - Trouver l'arête sortante de C de poids minimum: notée e
  - Soit S la composante connexe de l'autre extrémité de e
  - Créer une nouvelle composante connexe:
    - $C \cup \{e\} \cup S$

Exemple de Borůvka



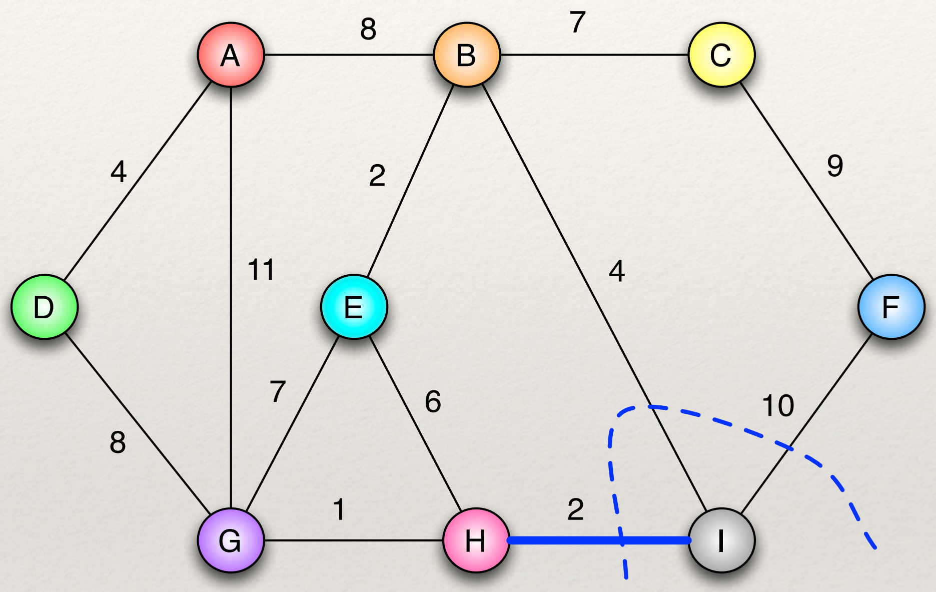
- Composantes={A}{B}{C}{D}{E}{F}{G}{H}{I}
- MST={}

Exemple de Borůkva



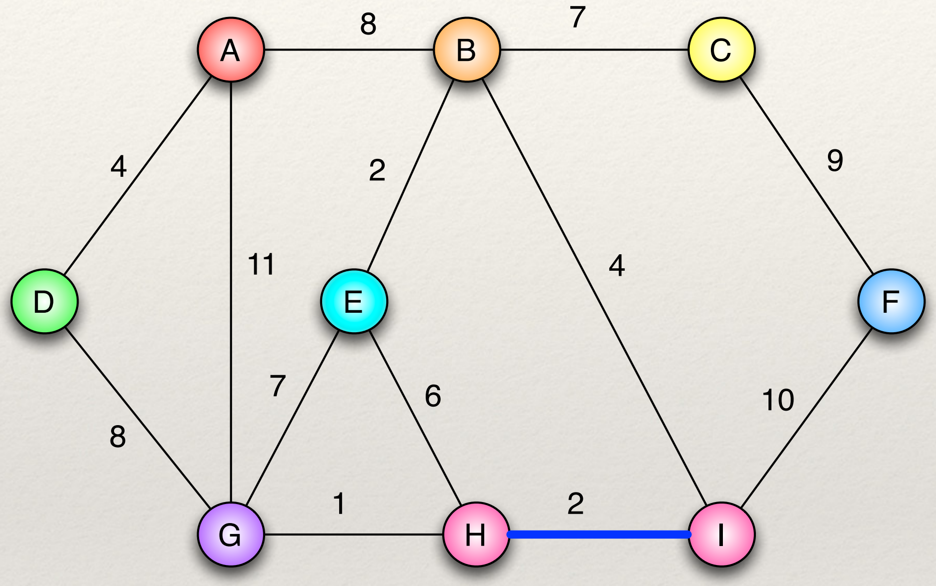
- Composantes={A}{B}{C}{D}{E}{F}{G}{H}{I}
- MST={}

Exemple de Borůvka



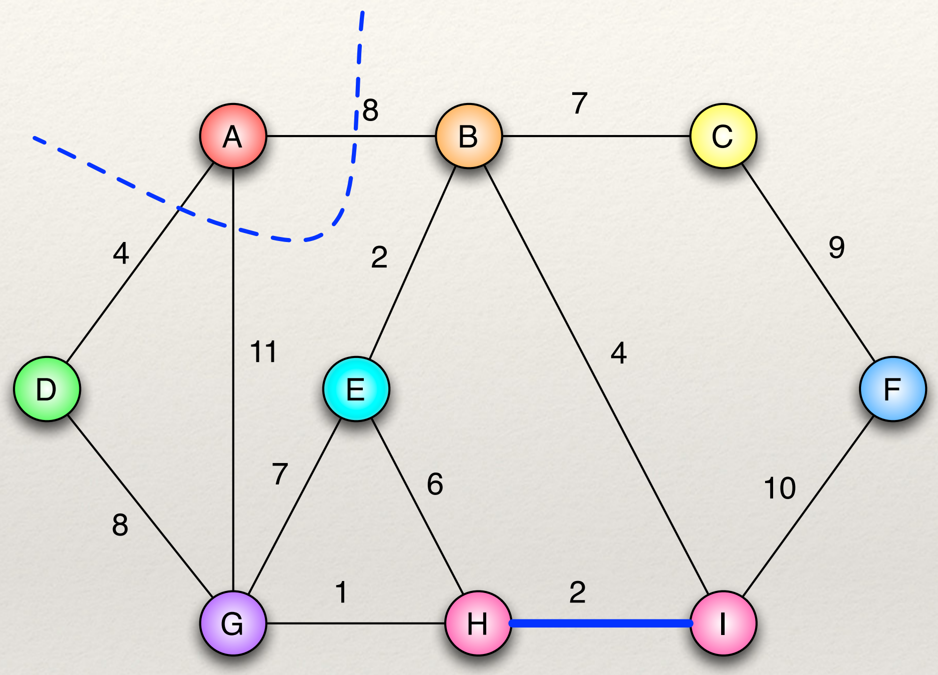
- {A}{B}{C}{D}{E}{F}{G}{H}{I}
- MST={}

Exemple de Borůkva



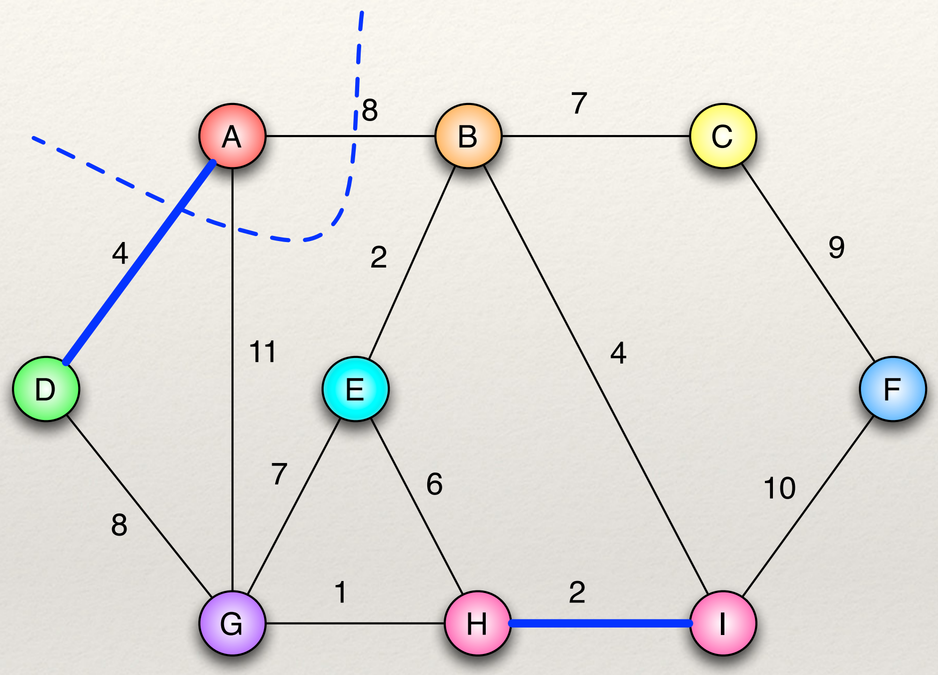
- {A}{B}{C}{D}{E}{F}{G}{H,I}
- MST={(H,I)}

Exemple de Borůvka



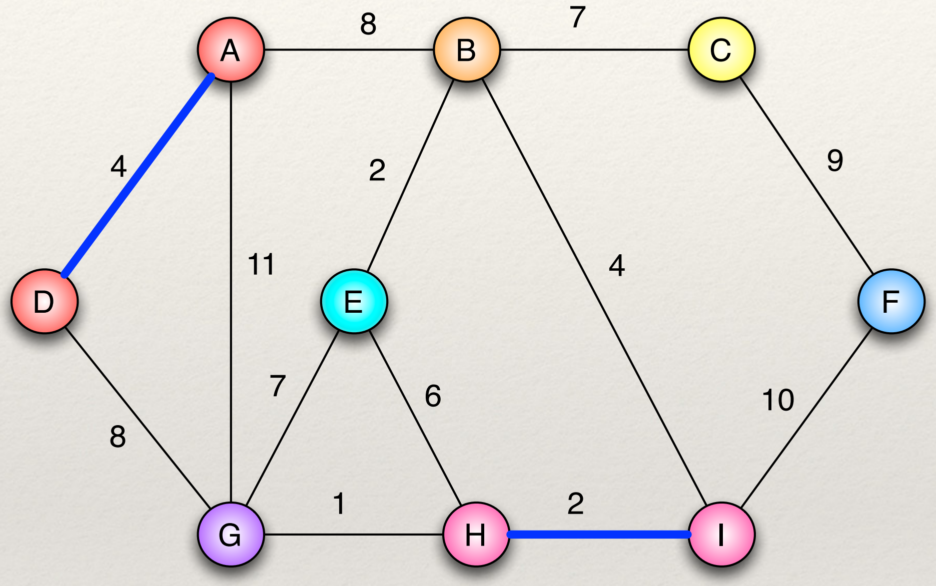
- {A}{B}{C}{D}{E}{F}{G}{H,I}
- MST={(H,I)}

Exemple de Borůkva



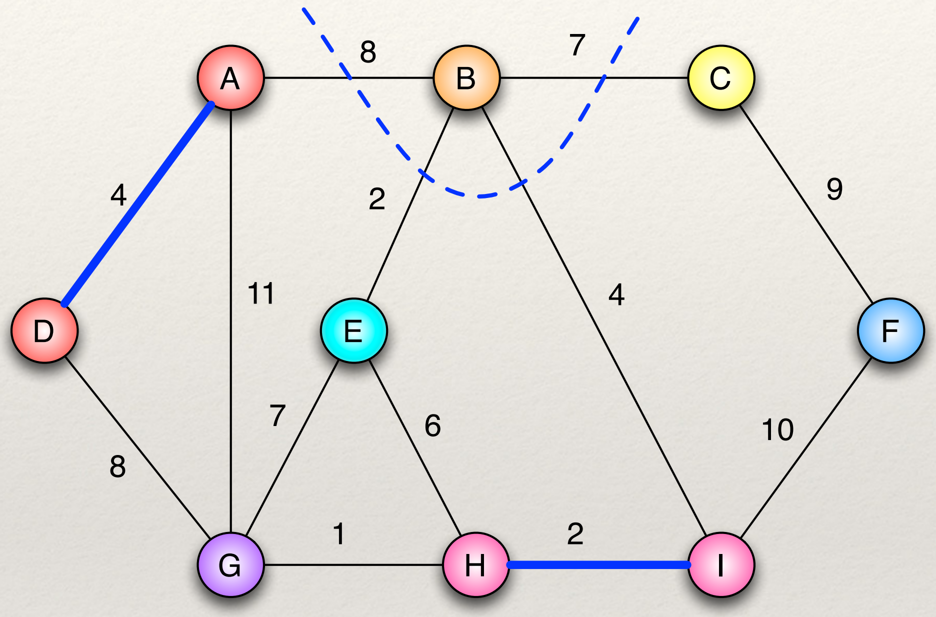
- {A}{B}{C}{D}{E}{F}{G}{H,I}
- MST={(H,I)}

Exemple de Borůvka



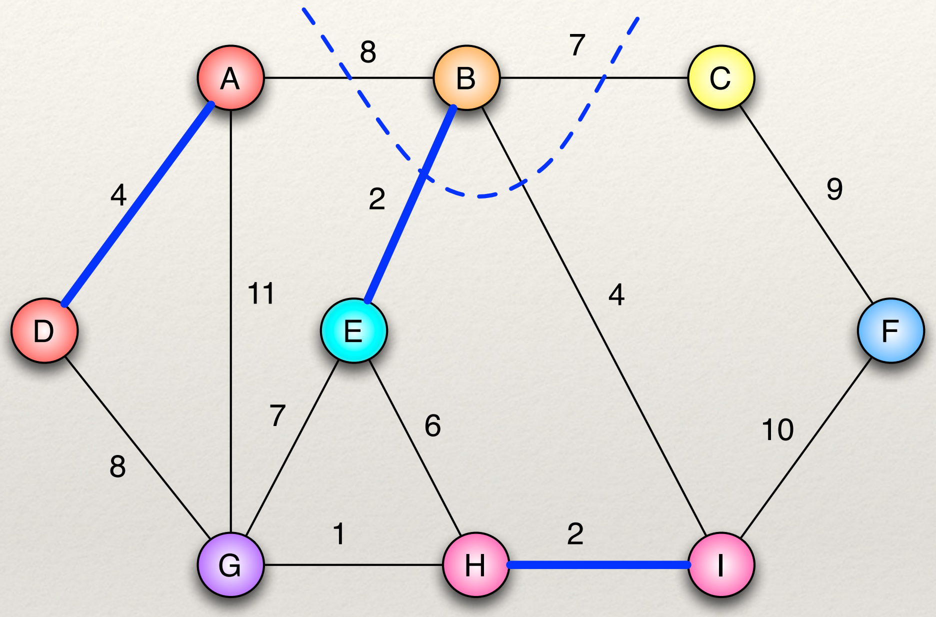
- {A,D}{B}{C}{E}{F}{G}{H,I}
- MST={(H,I),(A,D)}

Exemple de Borůkva



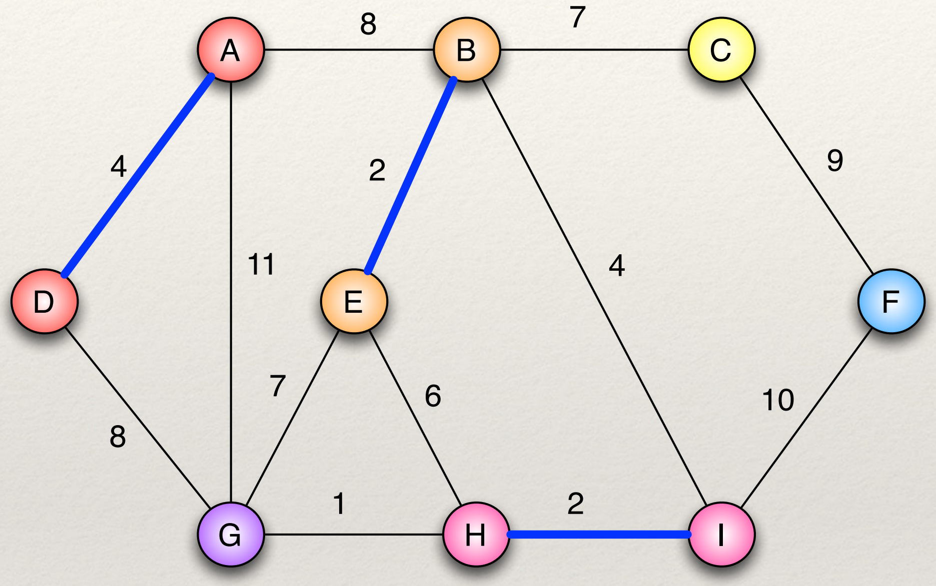
- {A,D}{B}{C}{E}{F}{G}{H,I}
- MST={(H,I),(A,D)}

Exemple de Borůkva



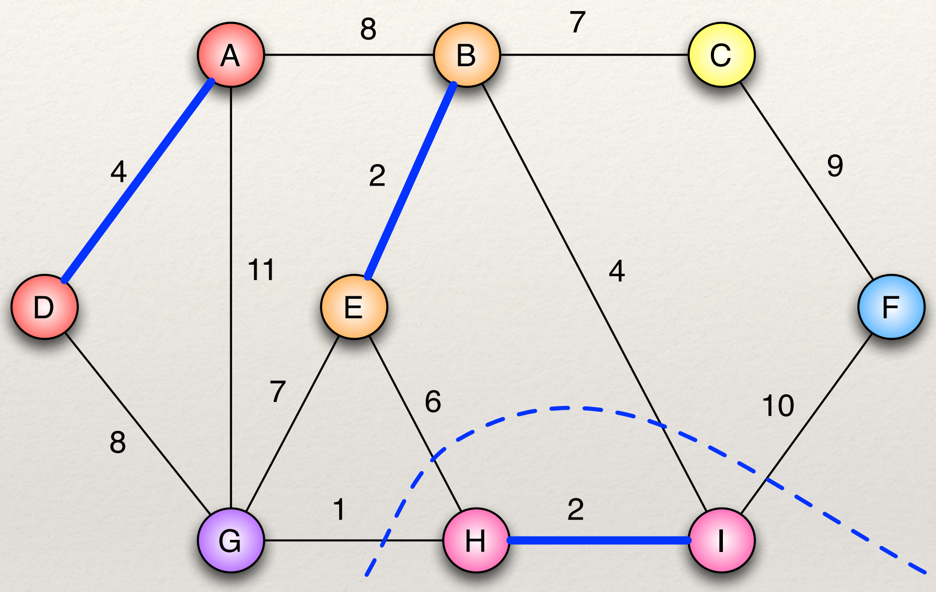
- {A,D}{B}{C}{E}{F}{G}{H,I}
- MST={(H,I),(A,D)}

Exemple de Borůkva



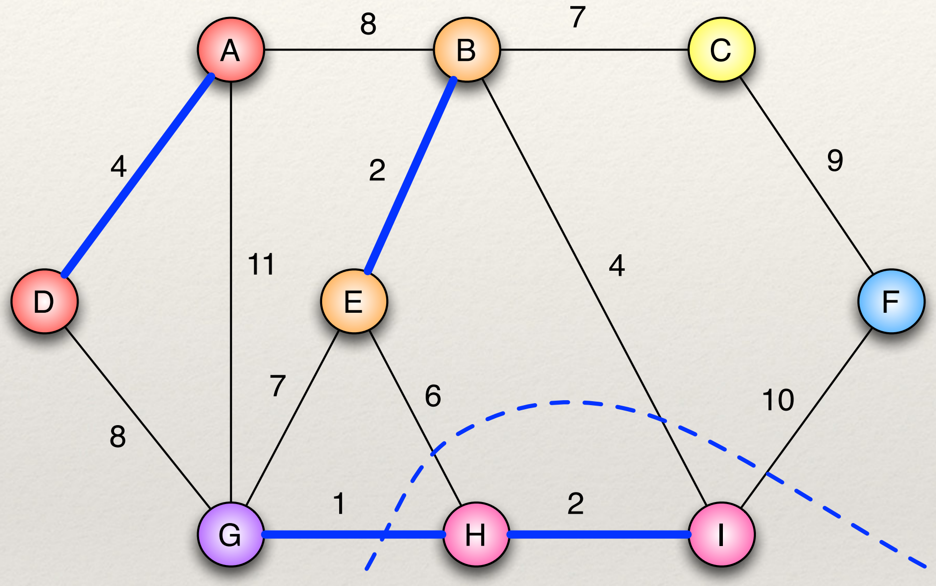
- {A,D}{B,E}{C}{F}{G}{H,I}
- MST={(H,I),(A,D),(B,E)}

Exemple de Borůkva



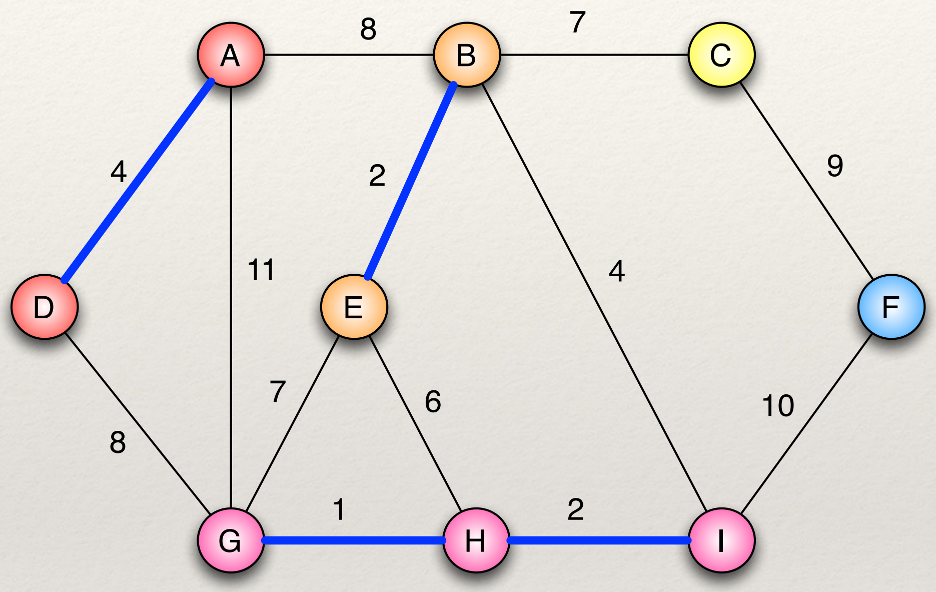
- {A,D}{B,E}{C}{F}{G}{H,I}
- MST={(H,I),(A,D),(B,E)}

Exemple de Borůkva



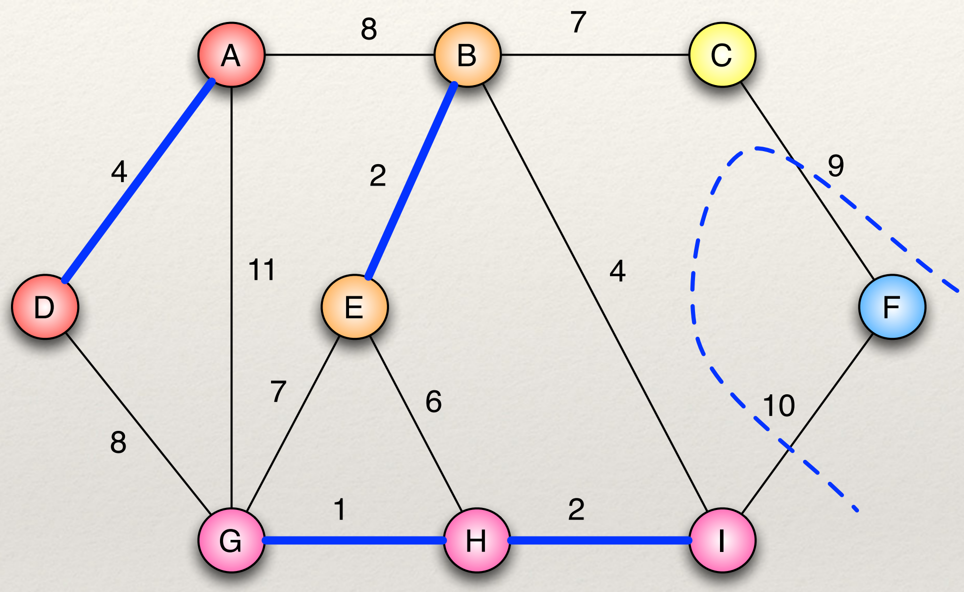
- {A,D}{B,E}{C}{F}{G}{H,I}
- MST={(H,I),(A,D),(B,E)}

Exemple de Borůkva



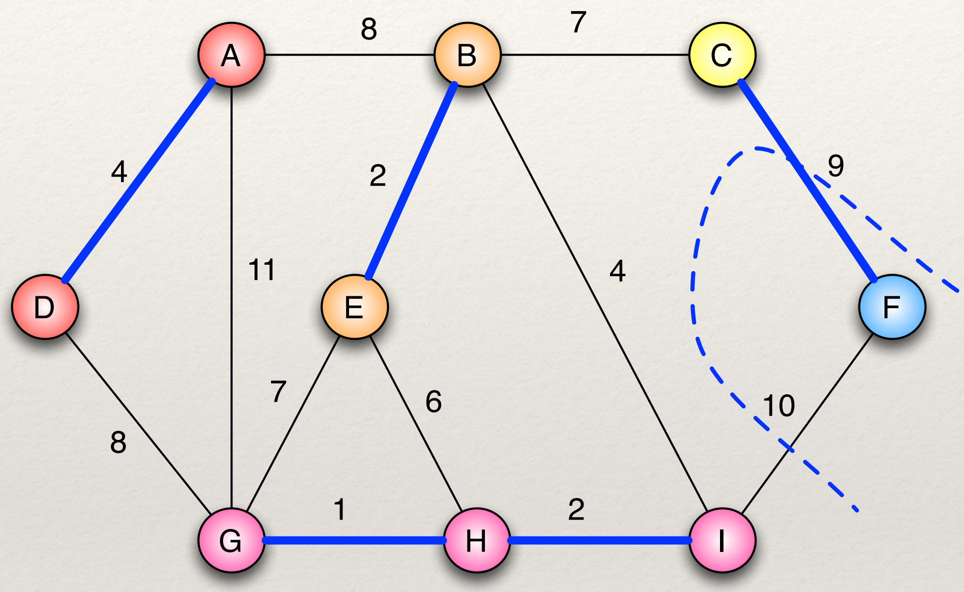
- {A,D}{B,E}{C}{F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H)}

Exemple de Borůvka



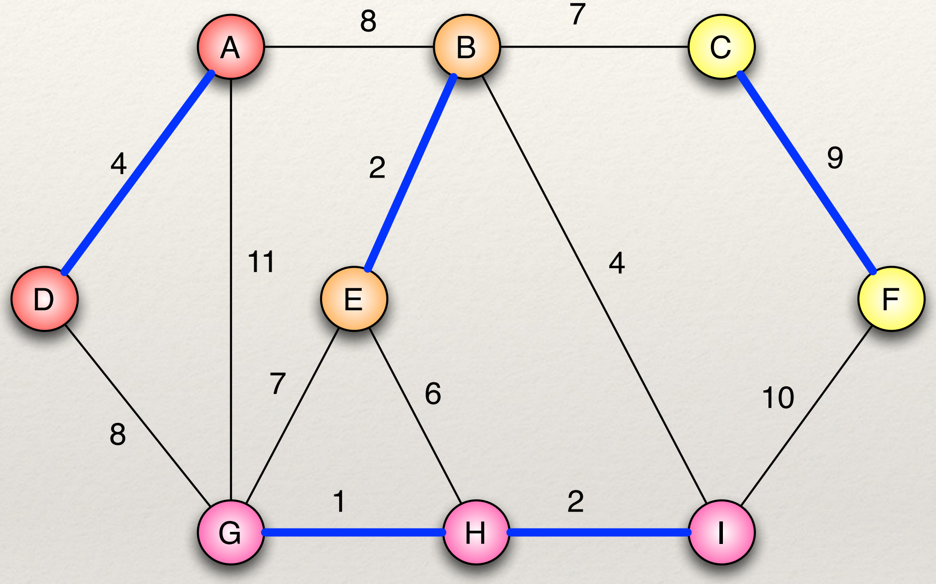
- {A,D}{B,E}{C}{F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H)}

Exemple de Borůkva



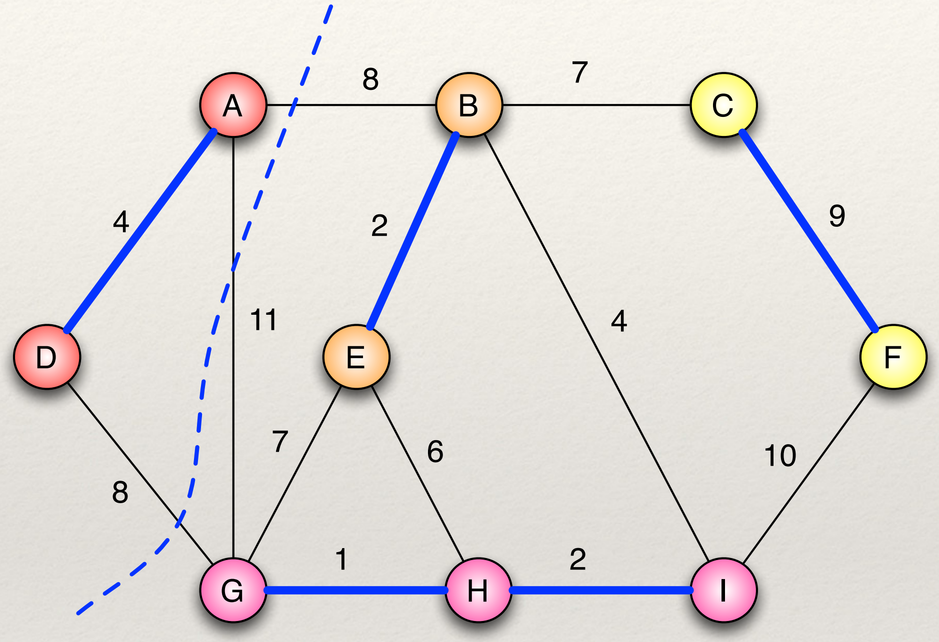
- {A,D}{B,E}{C}{F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H)}

Exemple de Borůkva



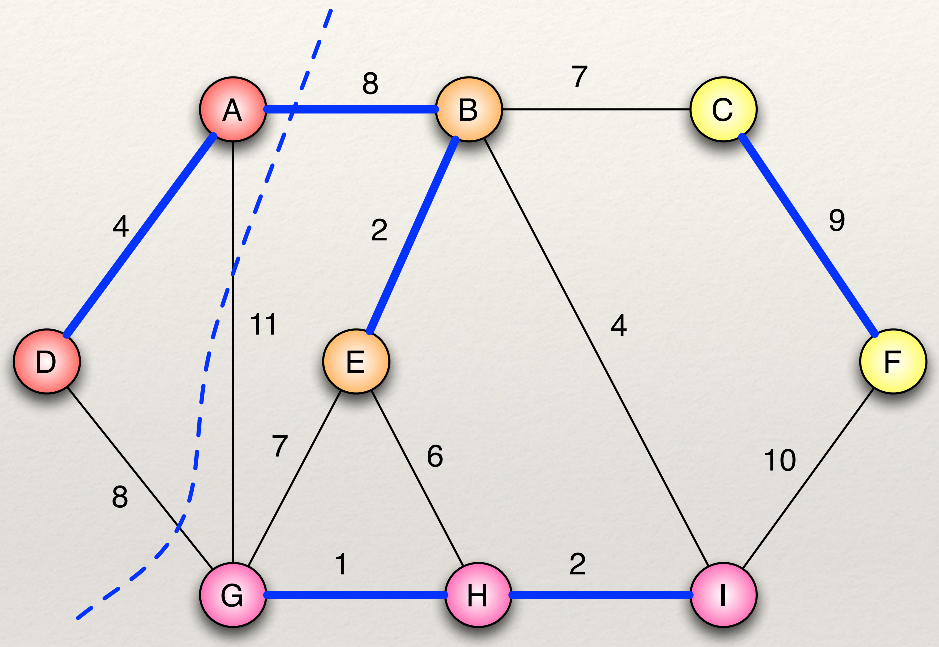
- {A,D}{B,E}{C,F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F)}

Exemple de Borůkva



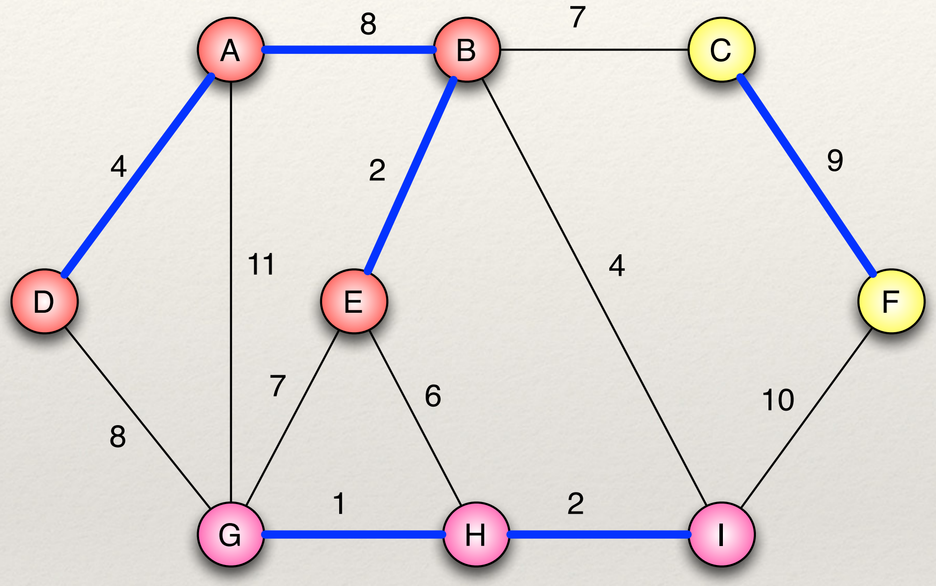
- {A,D}{B,E}{C,F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F)}

Exemple de Borůkva



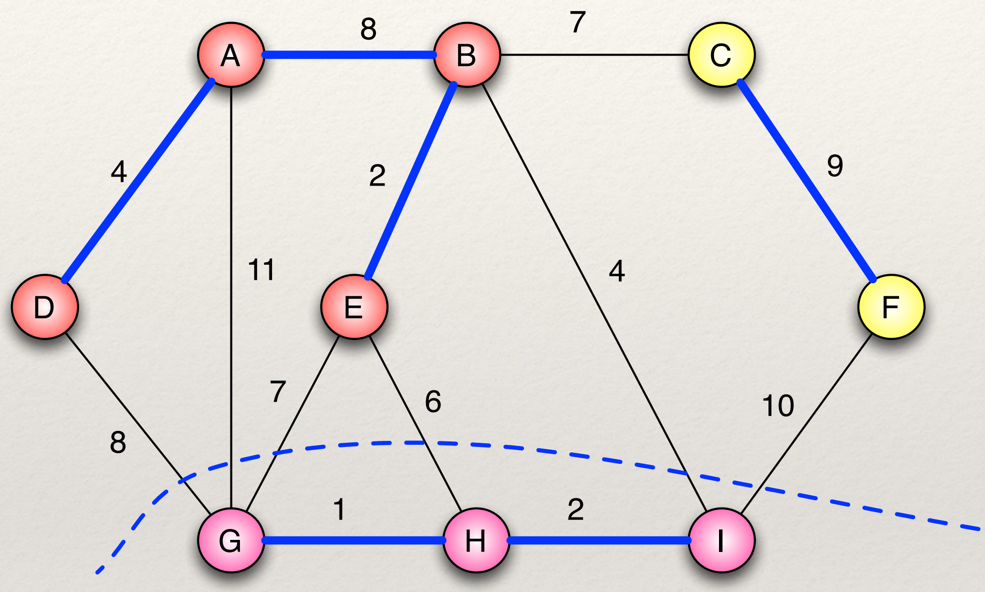
- {A,D}{B,E}{C,F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F)}

Exemple de Borůkva



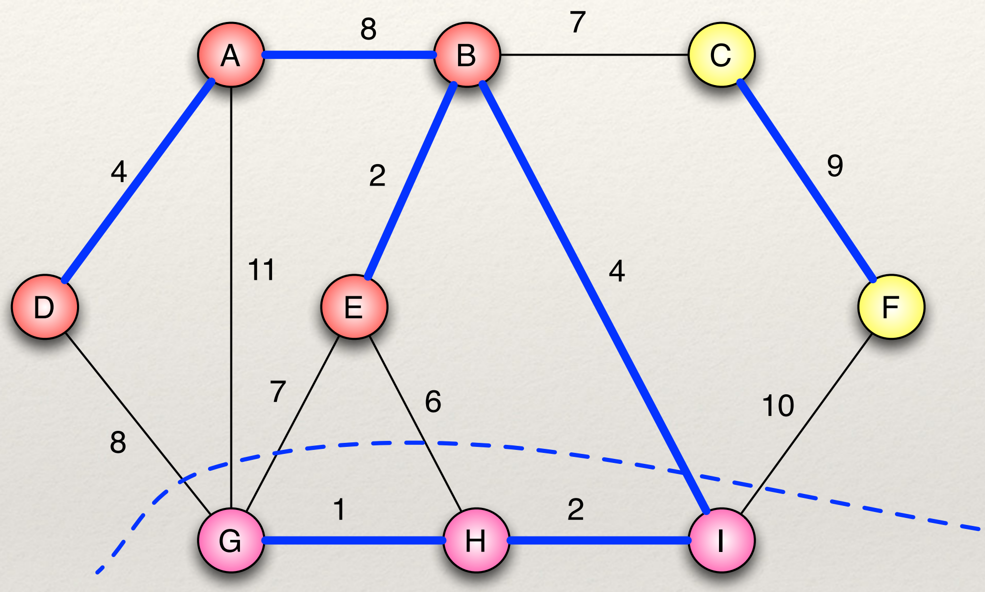
- {A,B,D,E}{C,F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B)}

Exemple de Borůkva



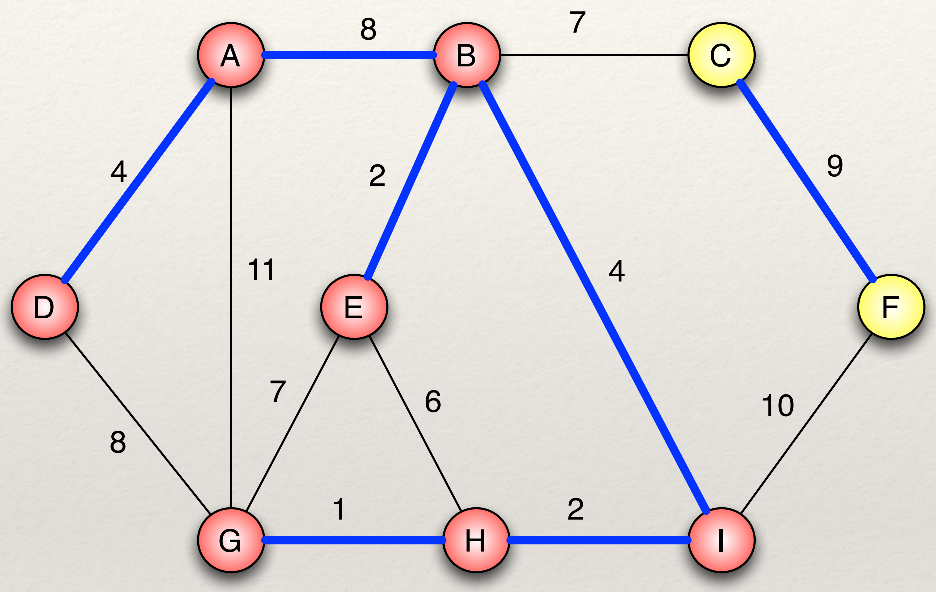
- {A,B,D,E}{C,F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B)}

Exemple de Borůkva



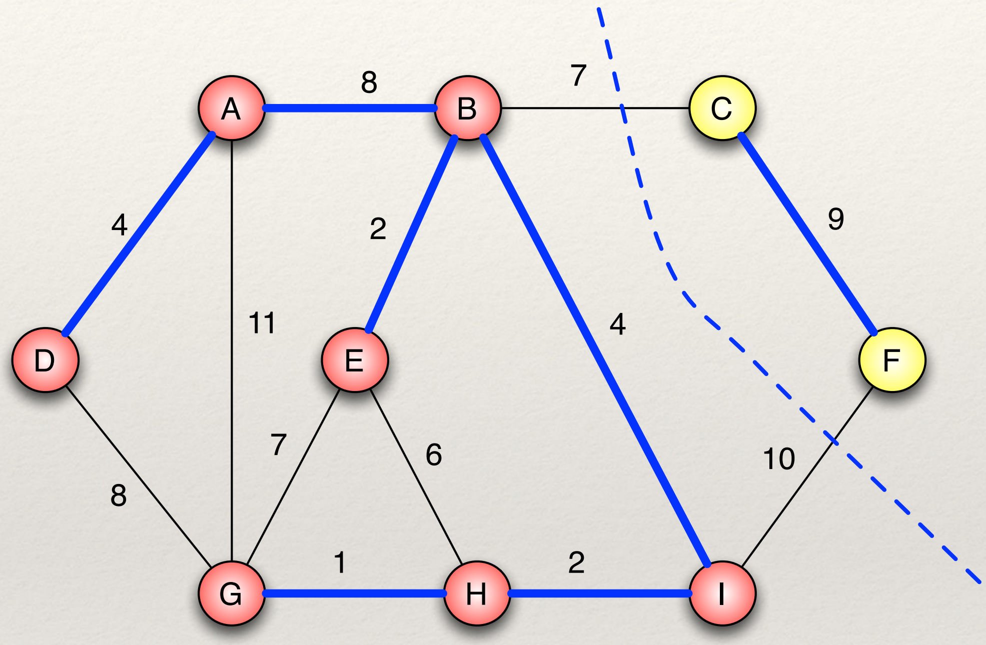
- {A,B,D,E}{C,F}{H,I,G}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B)}

Exemple de Borůkva



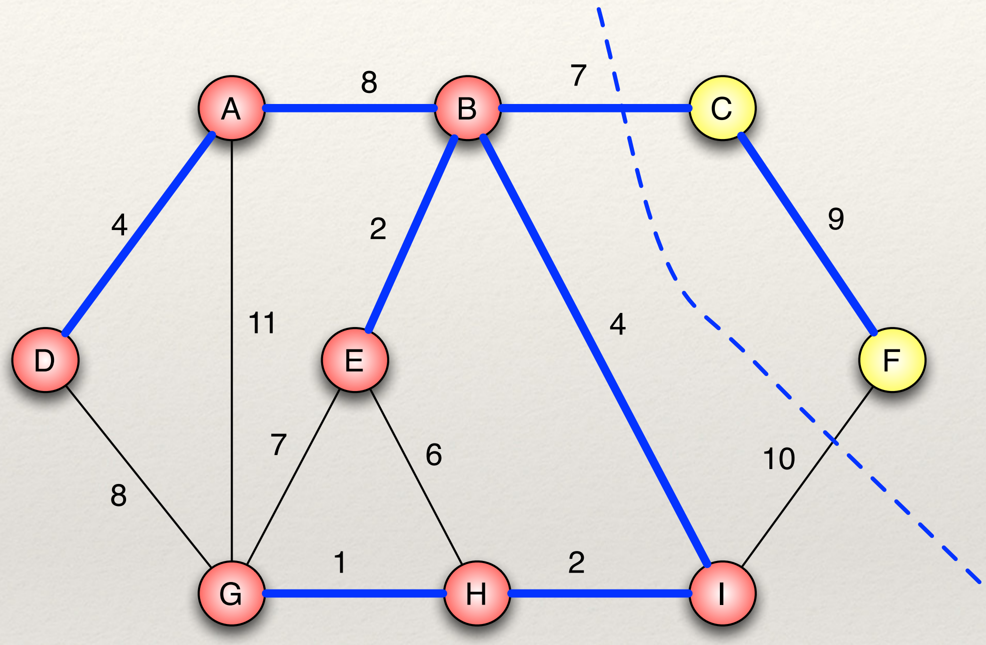
- {A,B,D,E,H,I,G}{C,F}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B),(B,I)}

Exemple de Borůkva



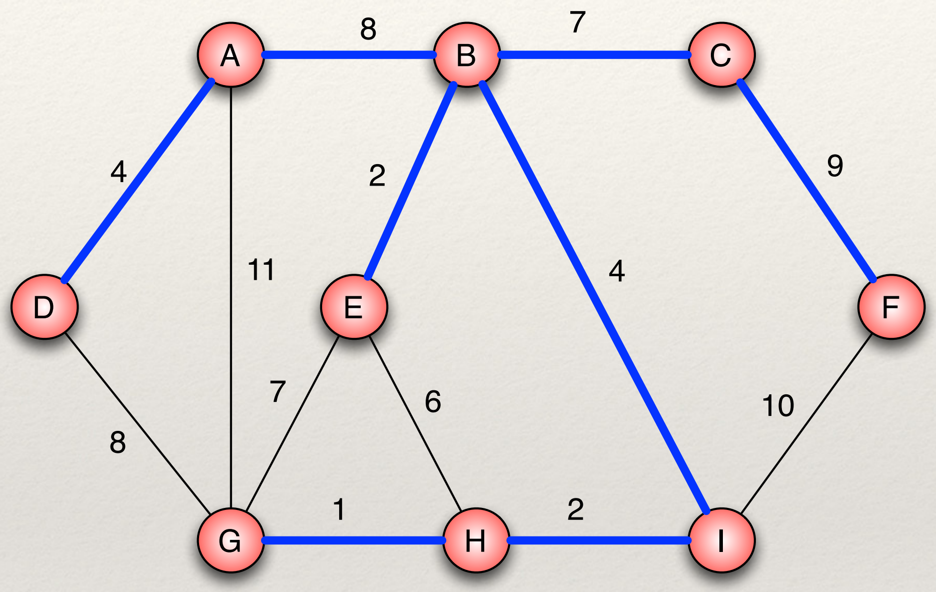
- {A,B,D,E,H,I,G}{C,F}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B),(B,I)}

Exemple de Borůkva



- {A,B,D,E,H,I,G}{C,F}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B),(B,I)}

Exemple de Borůkva



- {A,B,D,E,H,I,G,C,F}
- MST={(H,I),(A,D),(B,E),(G,H),(C,F),(A,B),(B,I),(B,C)}

Blin Lélia

Algorithme distribué

Arbre couvrant de poids minimum

Algorithmique répartie

M1 Université d'Evry

Passage au distribué

- Construire un algorithme pour le MST est « facile » en séquentiel beaucoup moins en distribué.

Gallager, Humblet et Spira (1983) *

- C'est le premier algorithme distribué
- Squelette de tous les autres travaux
  - Notion de fragment
    - Un fragment est l'ensemble des nœuds appartenant à un même sous-arbre de poids minimum.
- Nous allons voir dans ce cours une adaptation de l'algorithme de GHS83.

*Cet articles a valu à ces auteurs le prix Dijkstra en 2004

Algorithmes distribués pour le MST

- Initialement chaque nœud est un fragment
- Les nœuds d'un même fragment coopèrent
  - pour trouver l'arête de poids minimum sortante du fragment
- Les fragments sont fusionner entre eux
  - grâce à l'arête de poids minimum
- La démarche est répété jusqu'à
  - ce qu'il ne reste qu'un seul fragment
  - Le MST
- Algorithme de Borůkva

Difficultés

- Quelle sont les difficultés d'une telle approche?

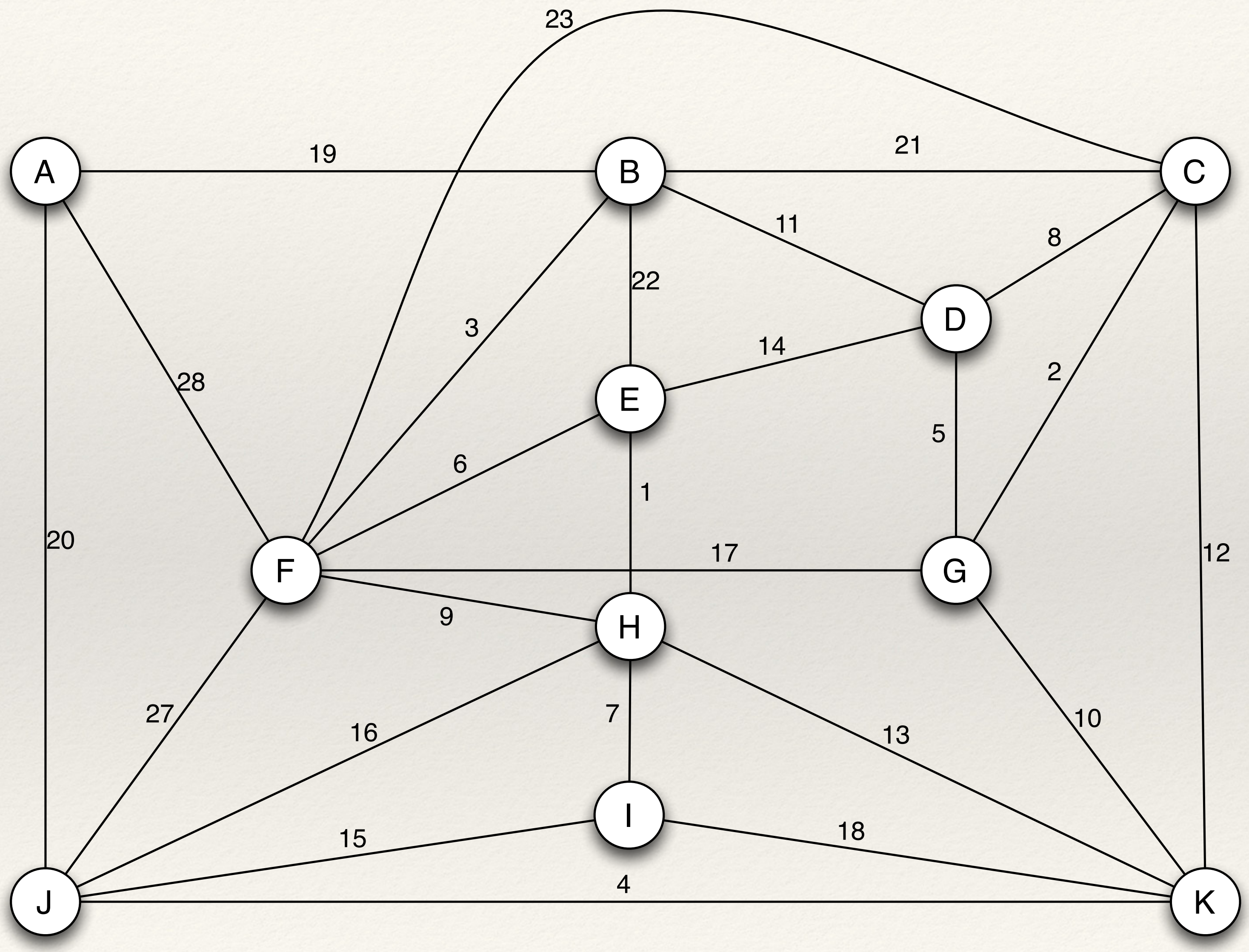
Version synchrone

- Initialement chaque nœud est un fragment
- Les nœuds d'un même fragment coopèrent
  - pour trouver l'arête de poids minimum sortante du fragment
- Les fragments sont fusionner entre eux
  - grâce à l'arête de poids minimum
- La démarche est répété jusqu'à ce qu'il ne reste
  - qu'un seul fragment
  - Le MST

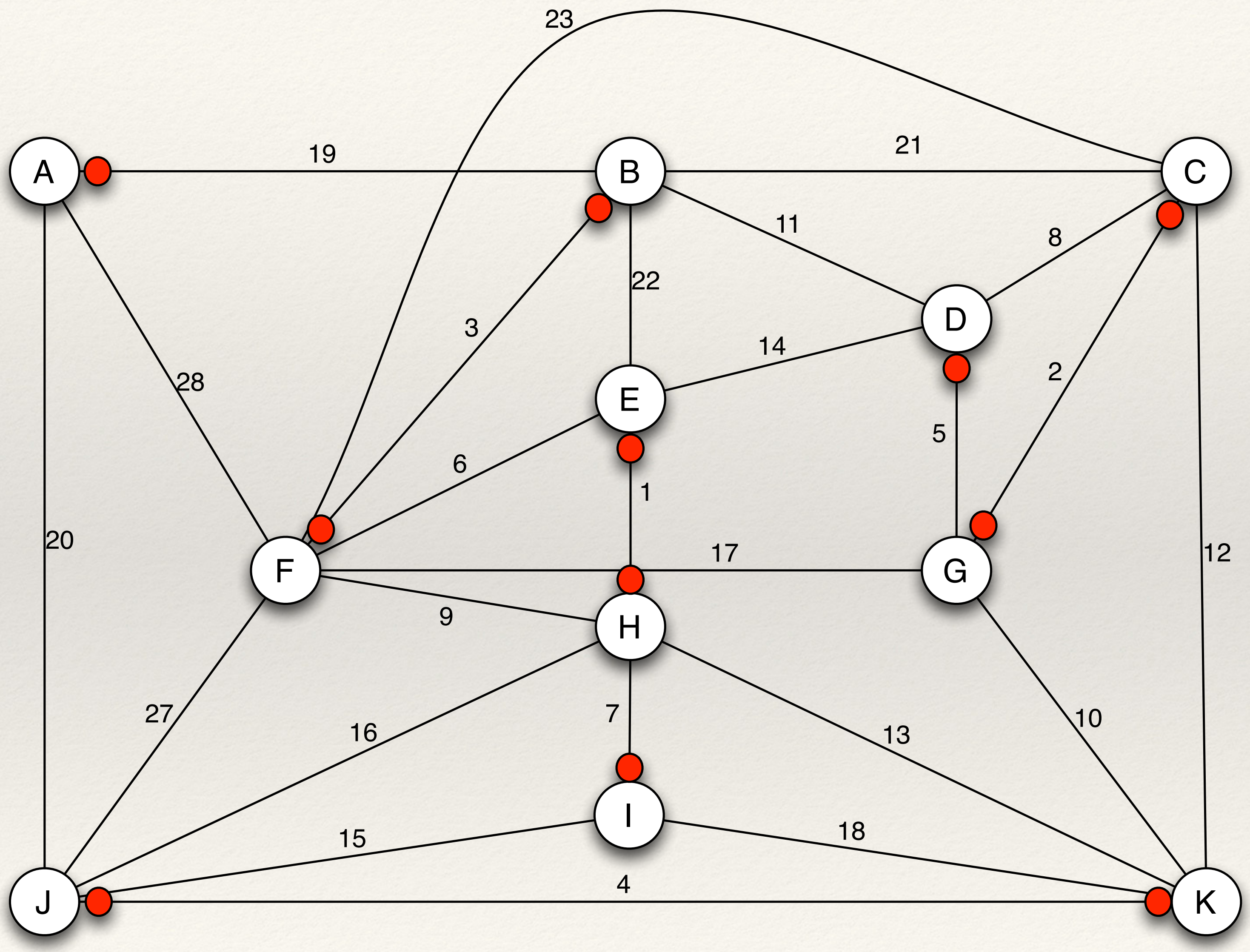
Version synchrone

- La première étape est «facile»
- Les fragments sont réduits à un seul noeud
- Donc chaque fragment peut décider localement de son arête sortante de poids minimum.

Exemple première étape



Exemple première étape



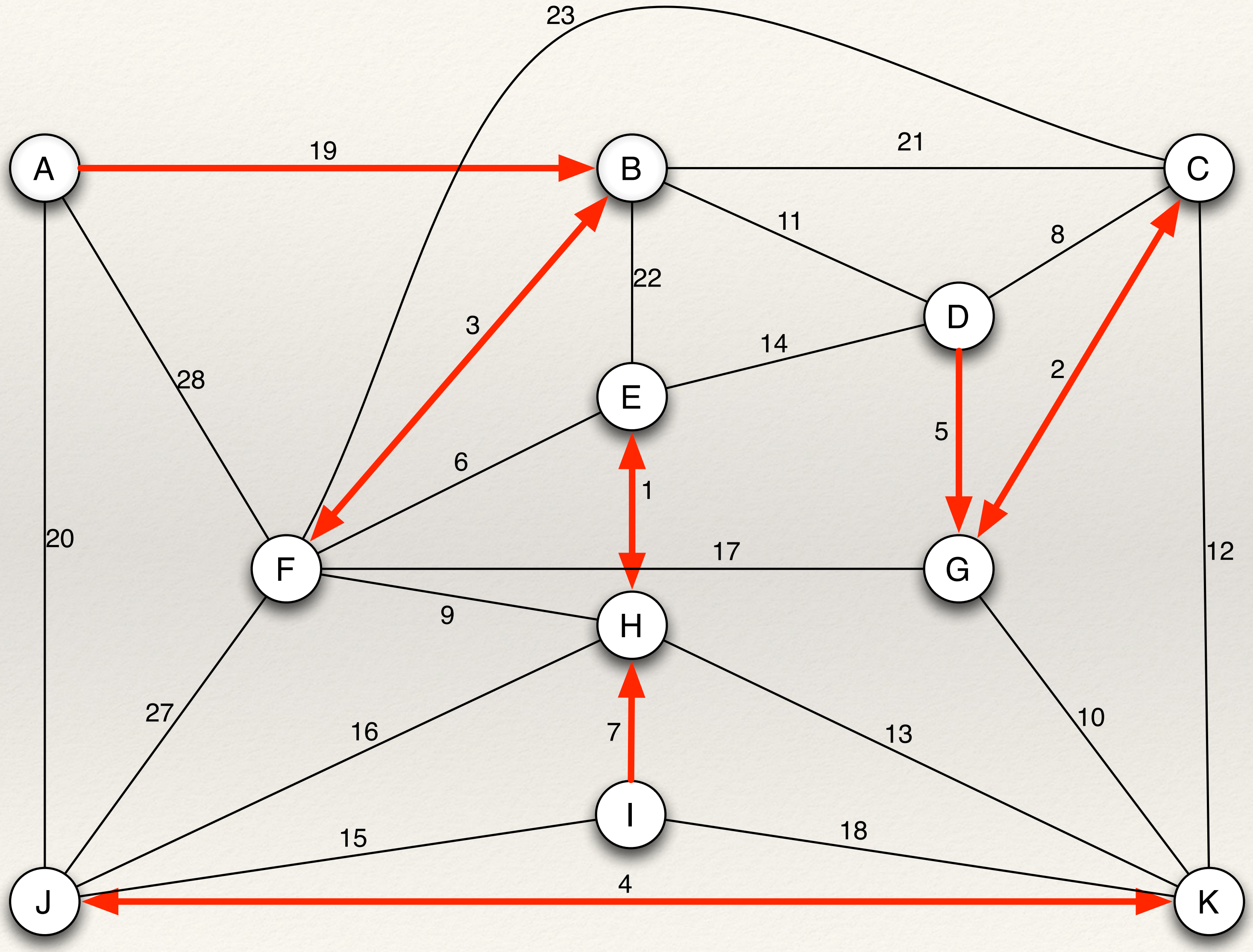
Et après la première étape

- Initialement chaque nœud est un fragment
- Les nœuds d'un même fragment coopèrent
  - pour trouver l'arête de poids minimum sortante du fragment
- Les fragments sont fusionner entre eux
  - grâce à l'arête de poids minimum
- La démarche est répété jusqu'à ce qu'il ne reste
  - qu'un seul fragment
  - Le MST

Fragments

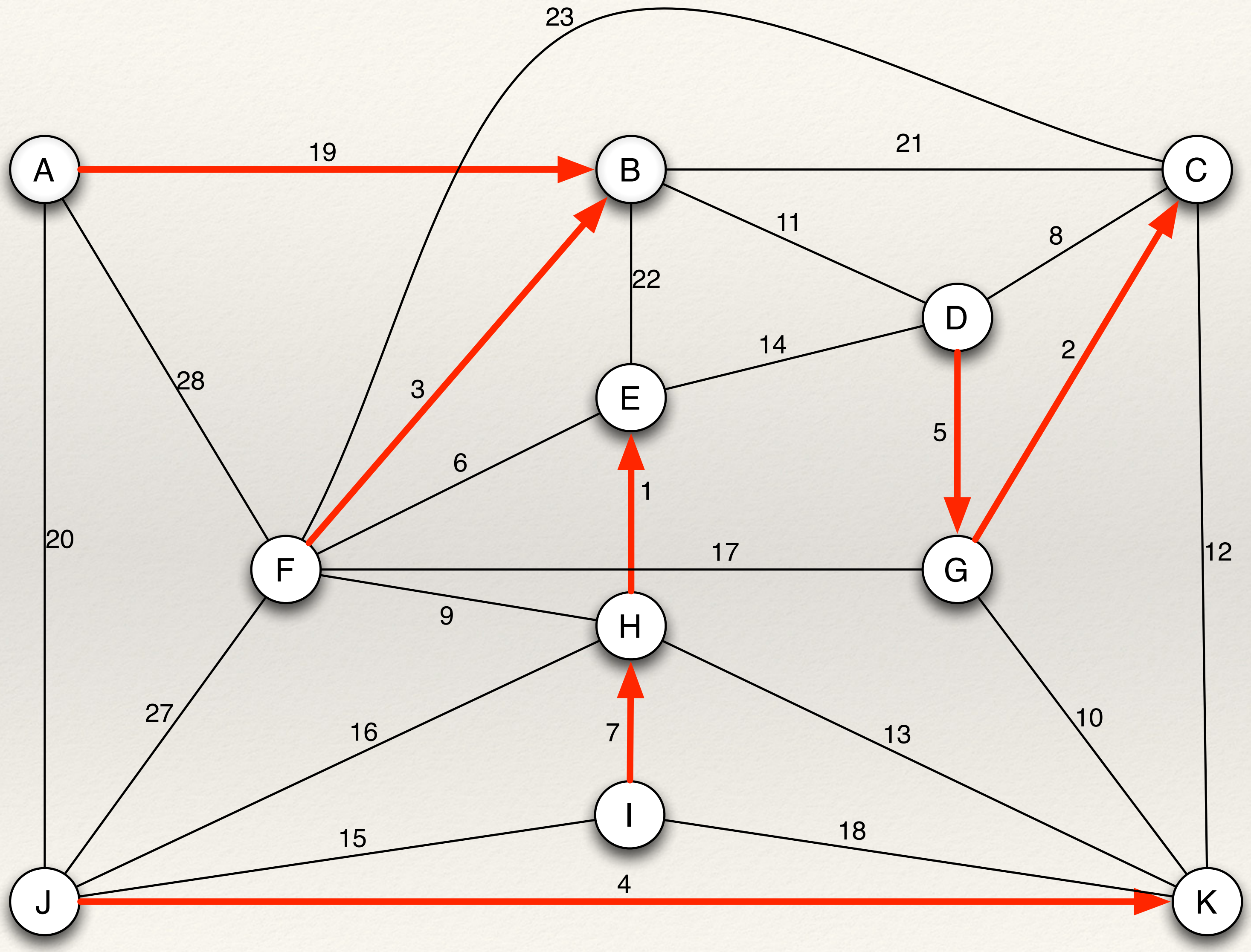
- Un fragment est un sous-arbre couvrant orienté.
- Le parent d'un noeud est l'autre extrémité de son arête de poids minimum
- Si deux noeuds ont:
  - la même arête de poids minimum
  - le parent est celui d'identifiant minimum

Exemple



23
A
19
B
21
C
11
8
22
D
3
14
28
E
2
5
6
1
20
17
12
F
G
9
H
27
16
7
13
10
I
15
18
4
J
K

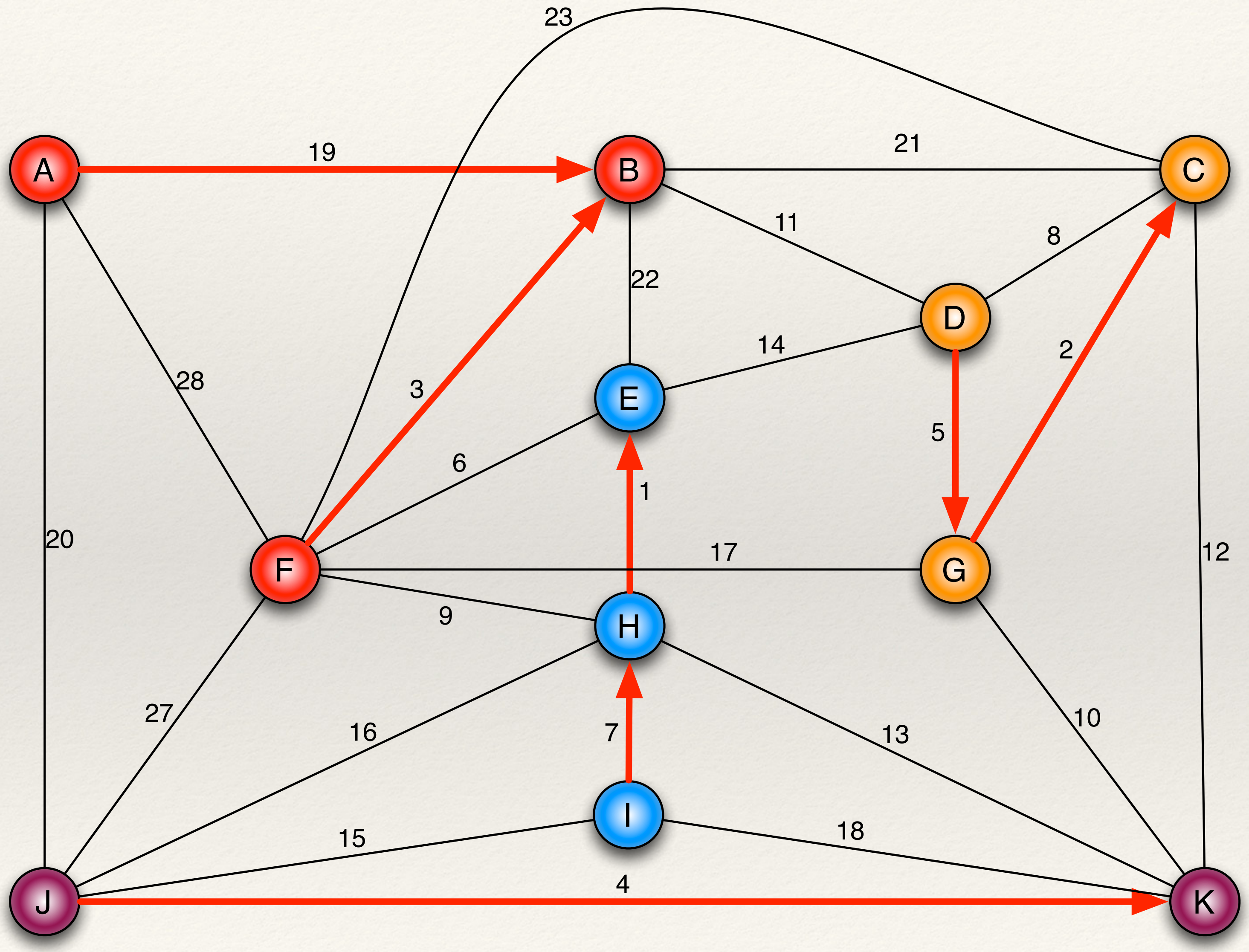
Exemple


23
A
19
B
21
C
11
8
22
D
14
2
28
3
E
5
6
1
20
17
12
F
G
9
H
27
16
7
13
10
15
I
18
4
J
K

Fragments

- La structure induite de la sélection des arêtes de poids minimum est un ensemble de sous-arbre.
- La racine de chaque sous-arbre diffuse son identifiant

Exemple


23
A
19
B
21
C
11
8
22
D
14
2
28
3
E
5
6
1
20
17
12
F
G
9
H
27
16
13
10
7
I
15
18
4
J
K

Arêtes sortantes

- Maintenant chaque noeud connait l'identifiant de son fragment
- Chaque noeud peut donc identifier
  - l'ensemble des arêtes internes au fragment
  - l'ensemble des arêtes sortantes du fragment

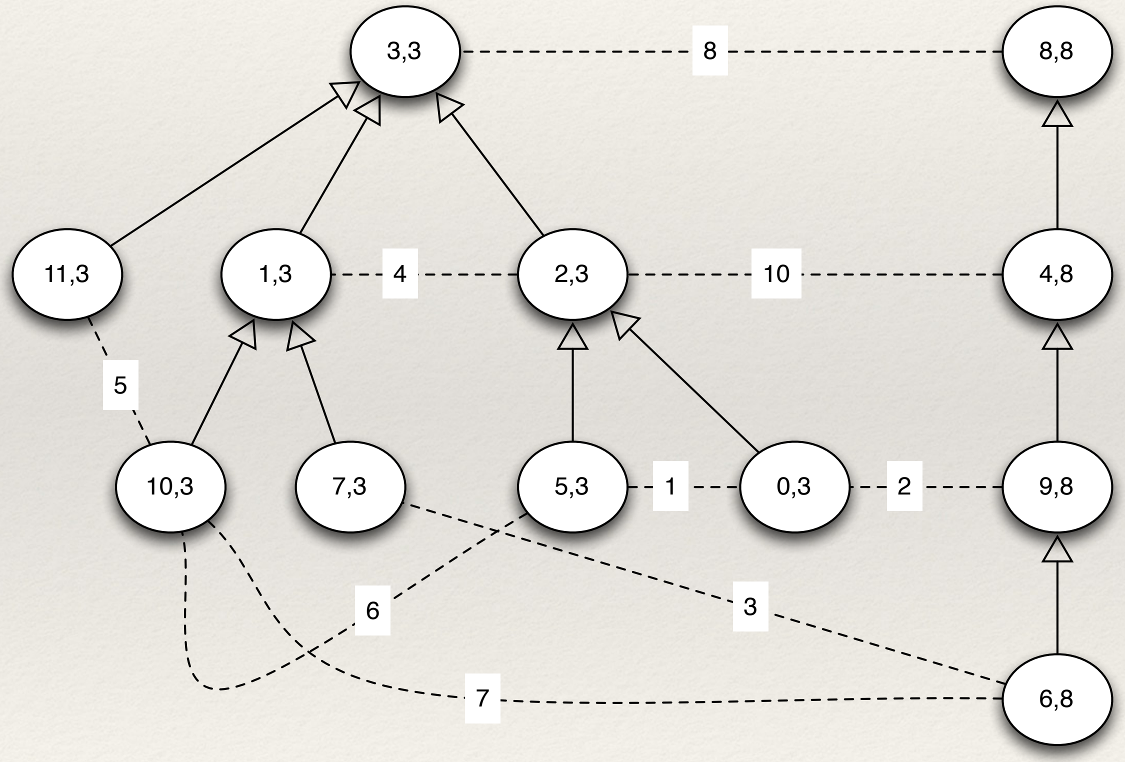
Arête sortante de poids minimum

- Comment choisir l'unique arête de poids minimum?

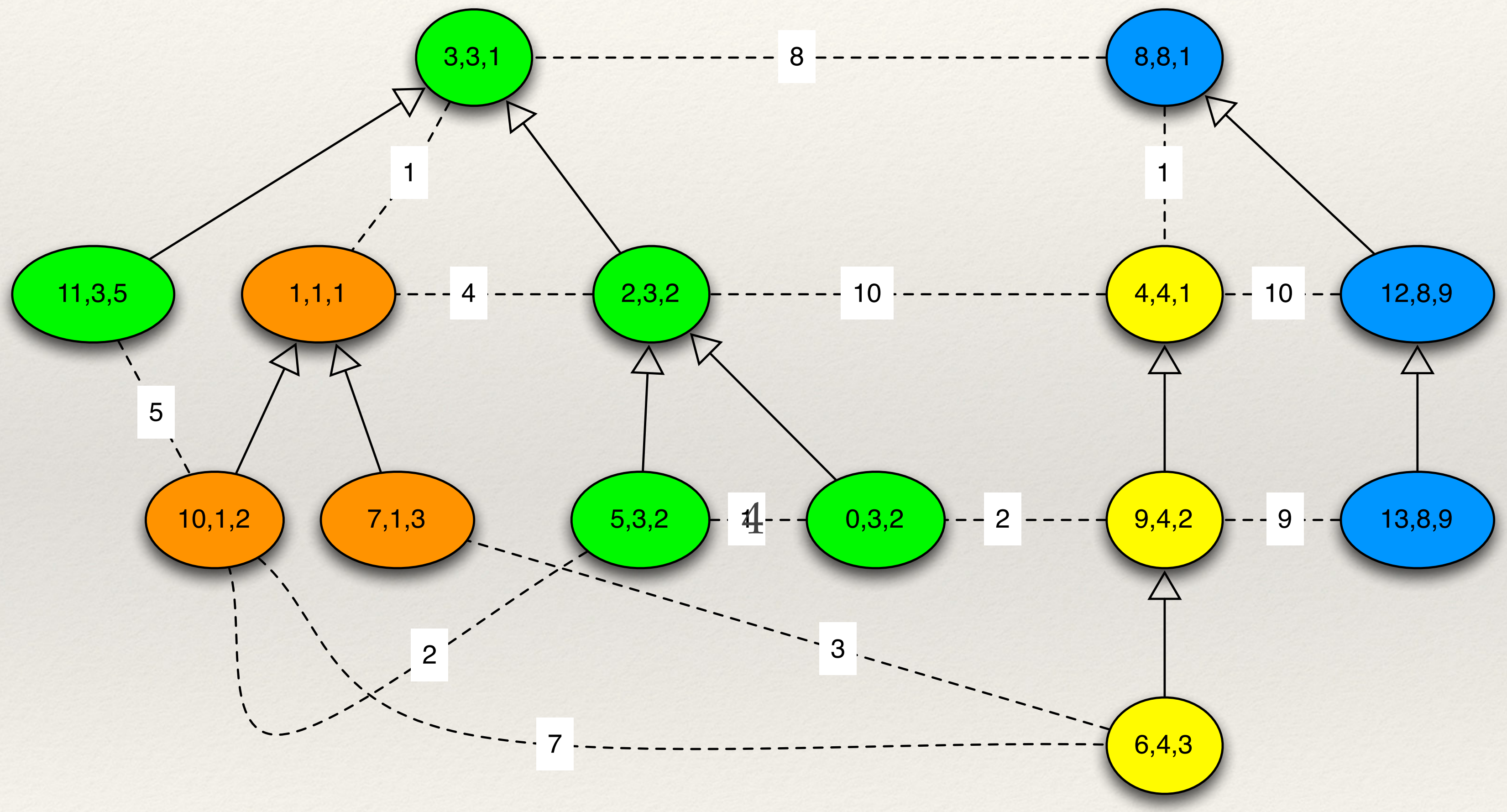
Arête sortante de poids minimum

- En partant des feuilles
  - Les noeuds sélectionnent l'arête de poids minimum sortante de leur sous-arbre.
- La racine concentre donc l'arête sortante de poids minimum du fragment

Arête sortante de poids minimum



Arête sortante de poids minimum



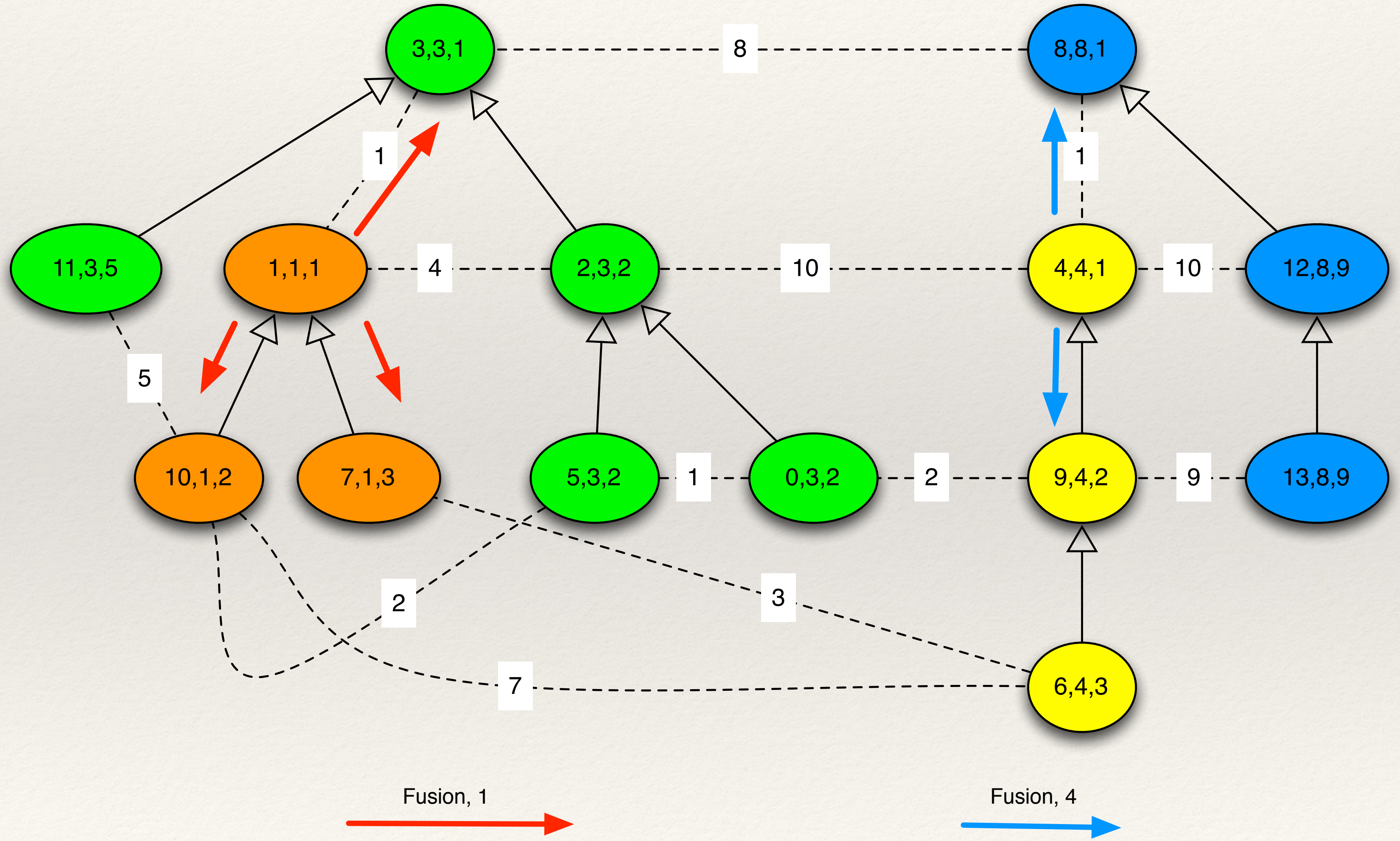
Comment fusionner deux fragments

- Les fragments f_1 et f_2 sont fusionnés grâce à l'arête de poids minimum e=(u,v)
  - avec $u \in f_1$ et $v \in f_2$
- La racine du fragment f_1 est déplacer au noeud u en réorientant le chemin de f_1 vers u
- Idem pour le fragment f_2

Comment fusionner deux fragments

- Un message de fusion est lancer depuis la nouvelle racine
  - message <Fusion, id nouveau fragment>

Arête sortante de poids minimum



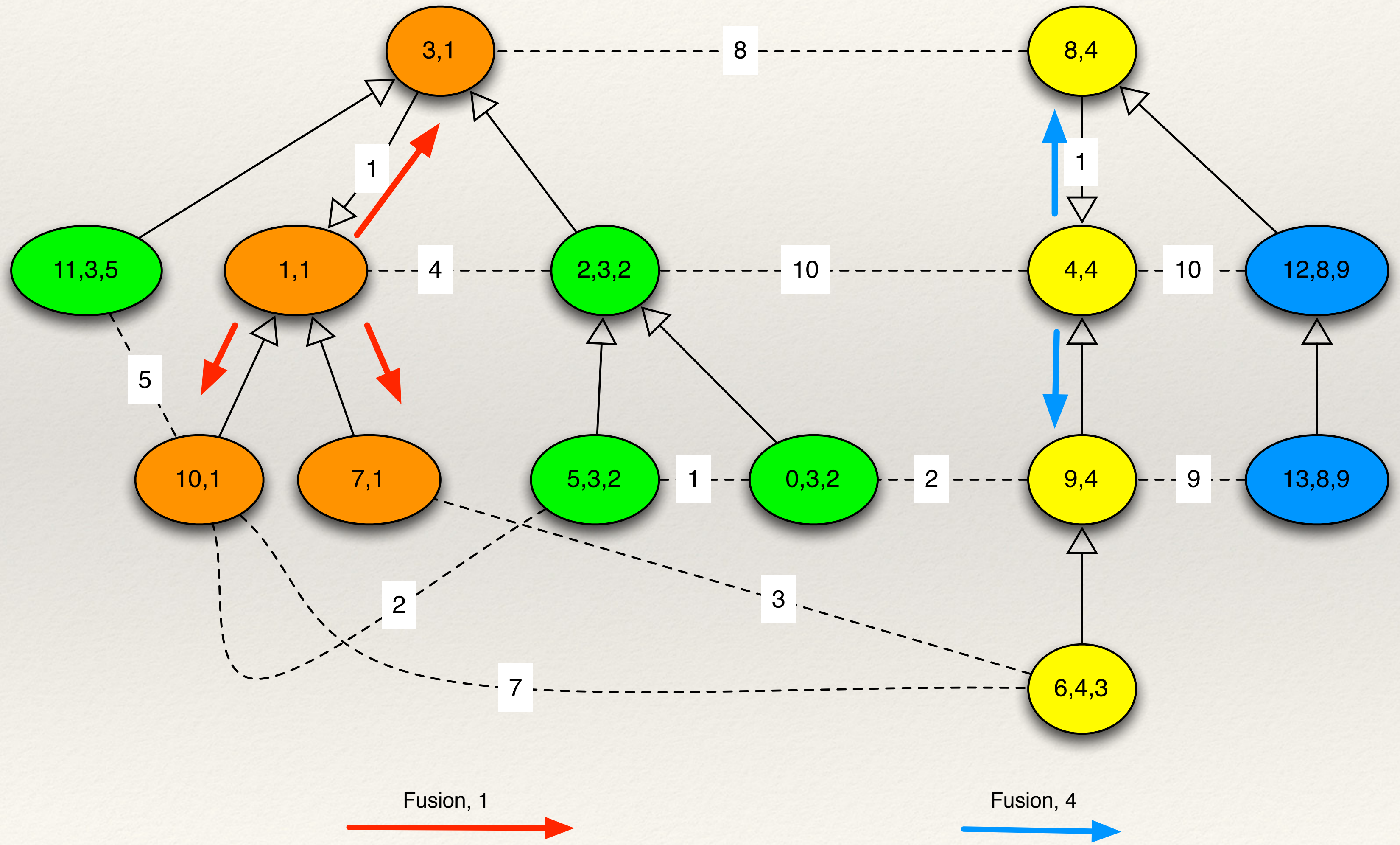
Comment fusionner deux fragments

- Quand un noeud u reçoit un message <Fusion,id_v>
- u est dans un autre fragment:
  - u est racine, il prend comme parent le noeud qui lui envoie le message
  - u change l'identifiant de son fragment

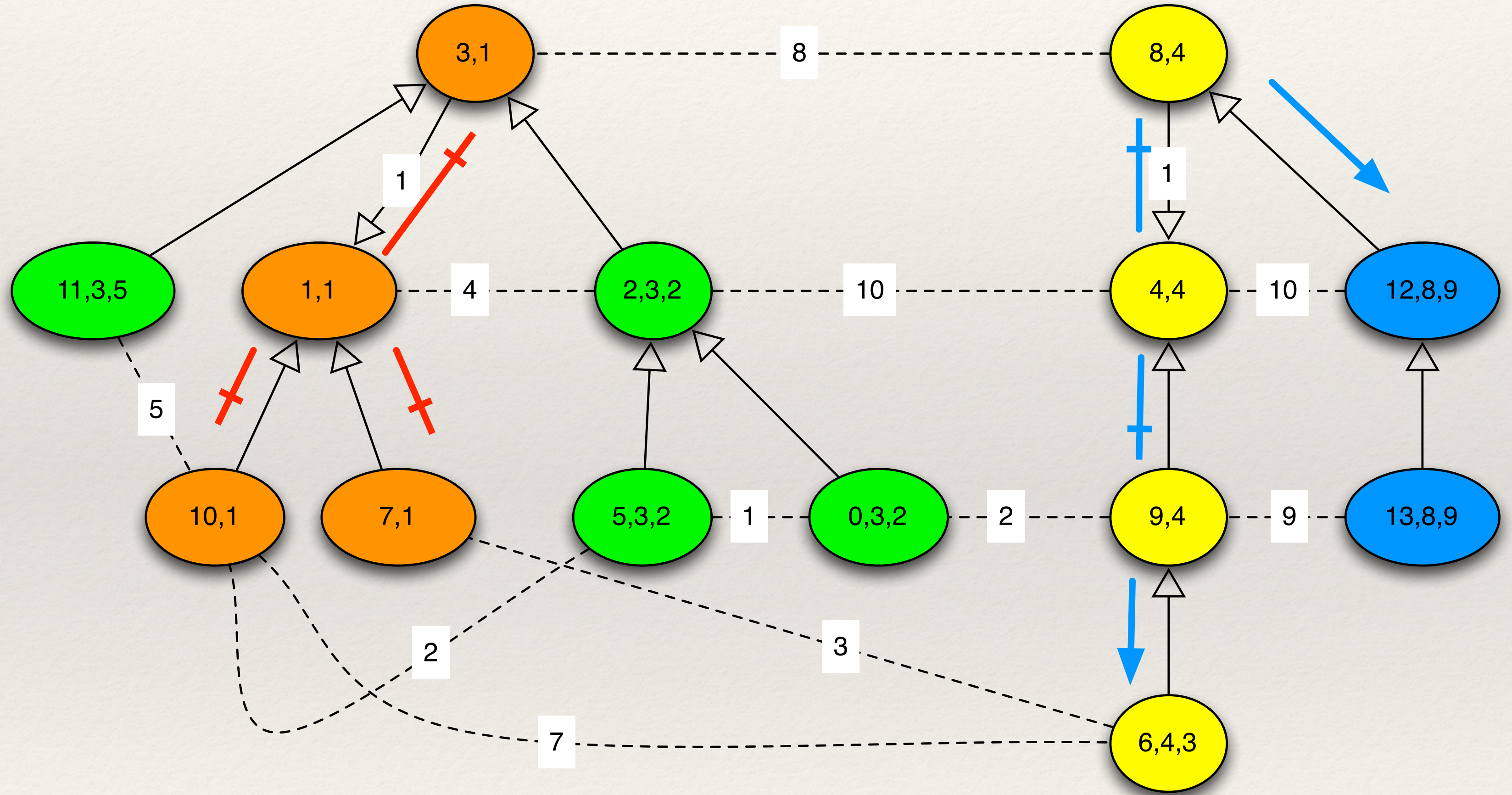
Comment fusionner deux fragments

- Quand un noeud v reçoit un message <Fusion,id_u>
- u est dans un autre fragment ou non:
    - u efface son arête de poids minimum
    - u envoie un message <Fusion,id_u> à ses enfants

Fusion


3,1
8
8,4
1
1
11,3,5
1,1
4
2,3,2
10
4,4
10
12,8,9
5
10,1
7,1
5,3,2
1
0,3,2
2
9,4
9
13,8,9
2
3
7
6,4,3
Fusion, 1
Fusion, 4

Fusion



3,1
8
8,4
1
1
11,3,5
1,1
4
2,3,2
10
4,4
10
12,8,9
5
10,1
7,1
5,3,2
1
0,3,2
2
9,4
9
13,8,9
2
3
7
6,4,3

Mise à jour des fragments

- Comment être sur que la mise à jour des fragments est terminé?
  - Un noeud sait qu'il à mis à jour son fragment car il a effacé son arête de poids minimum.
- Quand démarrer la collecte de l'arête de poids minimum?
  - Comment être sur que les voisins ont fini leurs mise à jour?

Collecte de l'arête de poids minimum

- Quand un noeud efface son arête de poids minimum il envoie fini à ses voisins (non enfants)
- Un noeud stocke fini pour chacun de ses voisins
- Une feuille u qui a effacée son arête de poids minimum et qui a stocké fini pour l'ensemble de ses voisins(en dehors du parent)
  - u envoie le message <Fragment>
- Chaque noeud qui reçoit le message <Fragment>
  - envoie l'identifiant de son fragment dans le message
    - <NumFragment, $Id_{fragment}$>

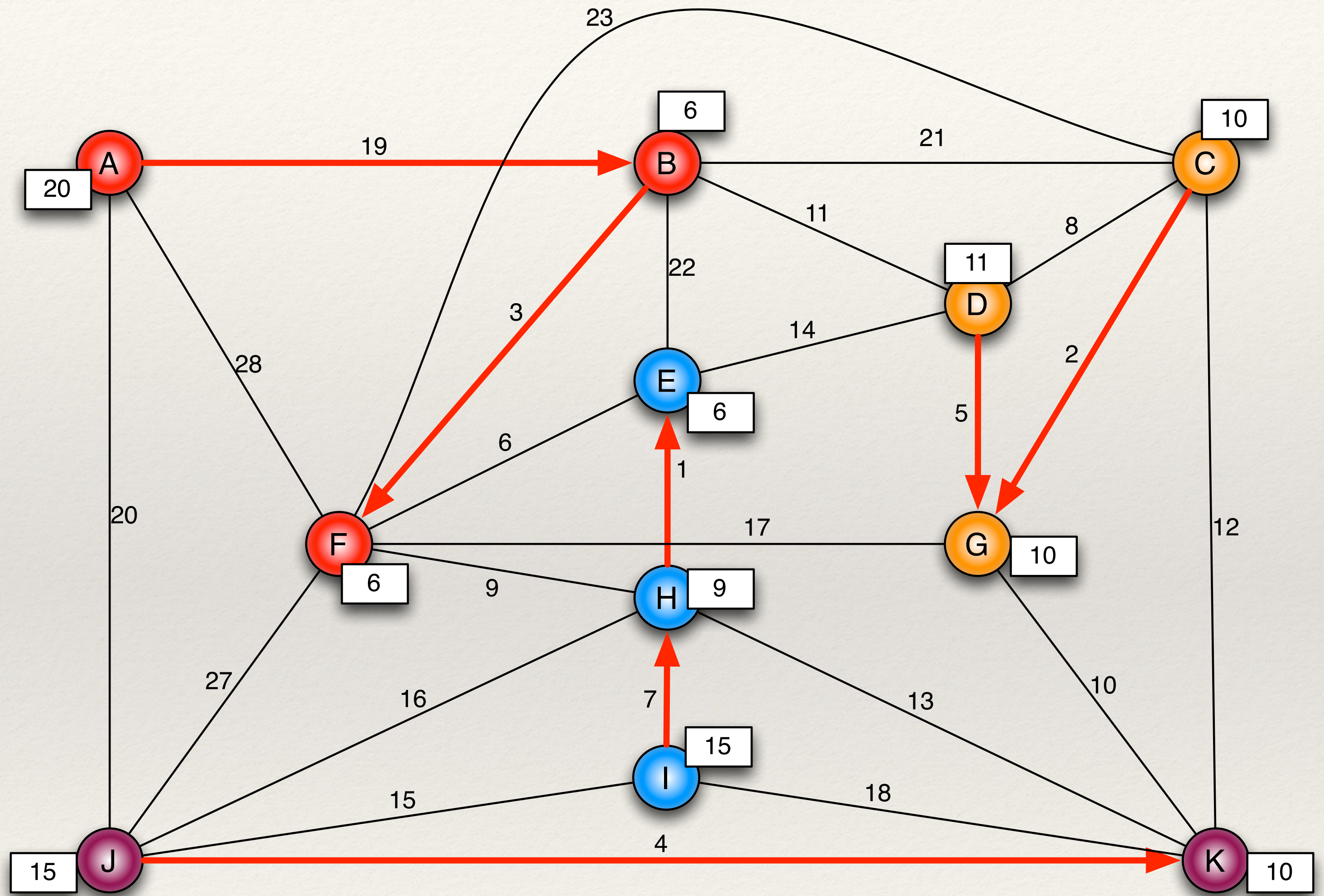
Collecte de l'arête de poids minimum

- Maintenant la collecte peut commencer
- Chaque feuille peut déterminer son arête de poids sortant et l'envoyer à son parent.

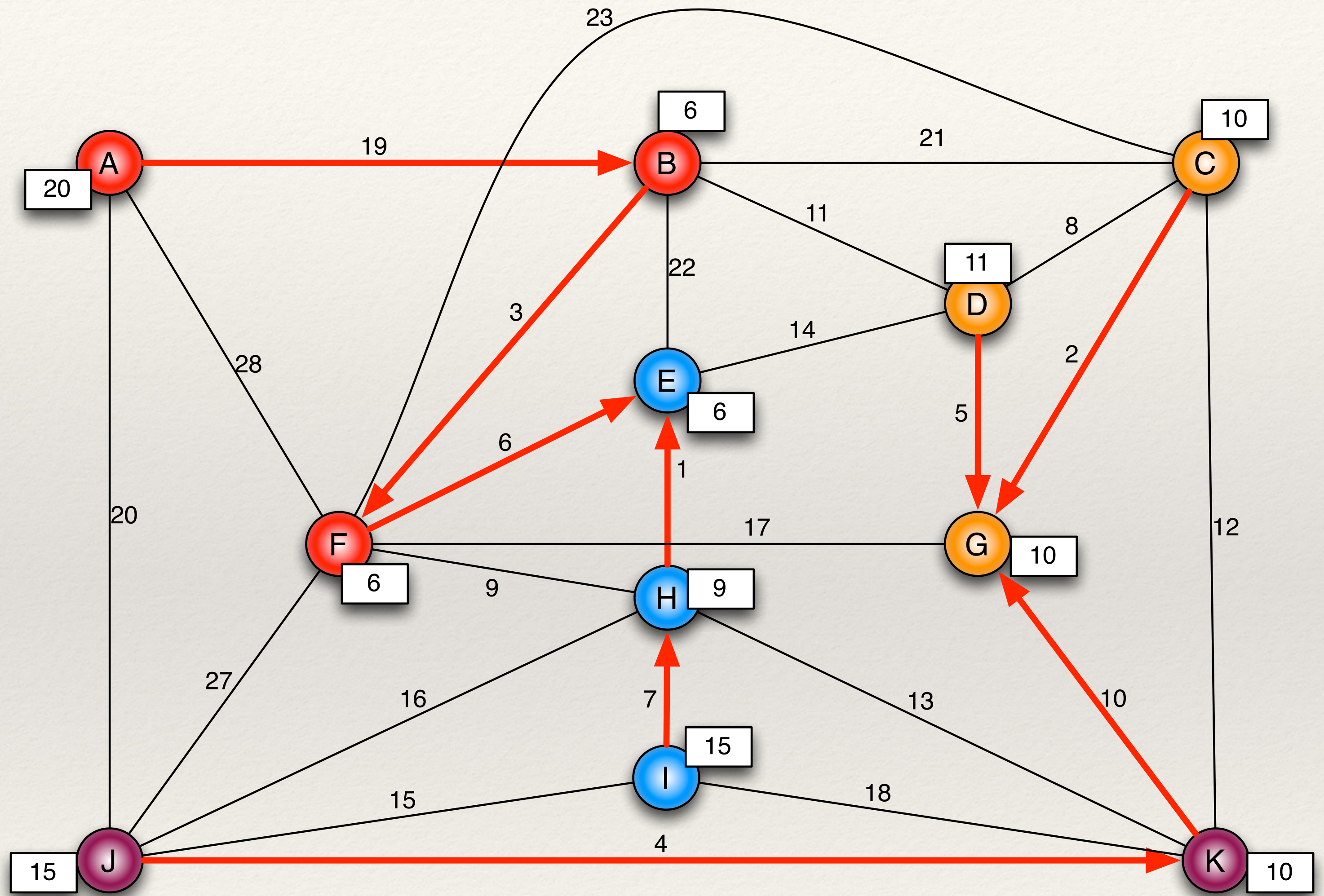
Nouvelles phases

- Il faut remettre à jour les fragments
  - Diffusion de l'identifiant à partir de la racine
- Collecter les nouvelles arêtes sortantes de poids minimum
- Fusion les fragments
  - ...

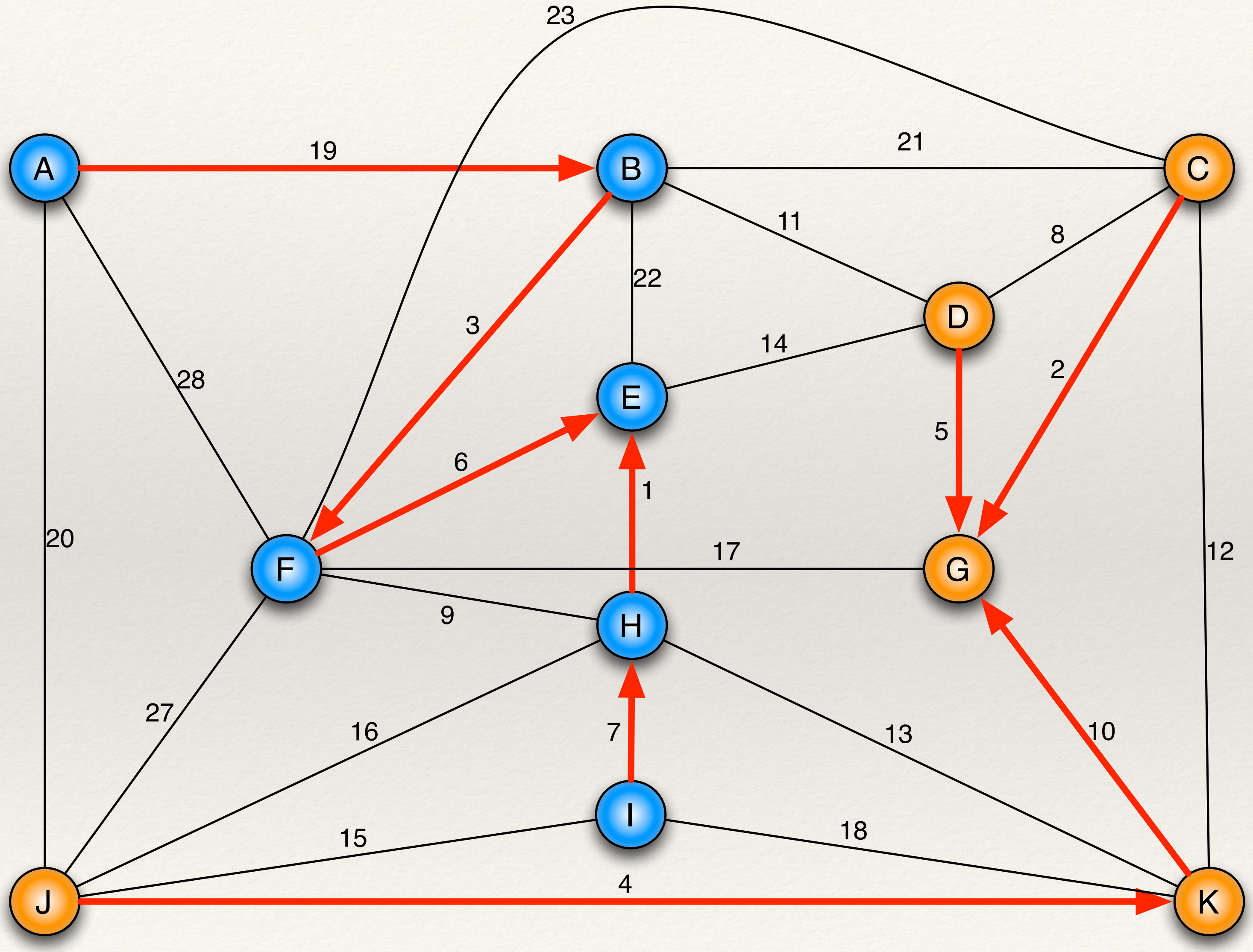
Exemple


23
6
10
19
21
A
B
C
20
11
8
11
22
D
3
14
28
E
2
6
5
6
1
20
17
12
F
G
10
9
6
H
9
27
16
10
13
7
15
15
18
I
4
J
K
15
10

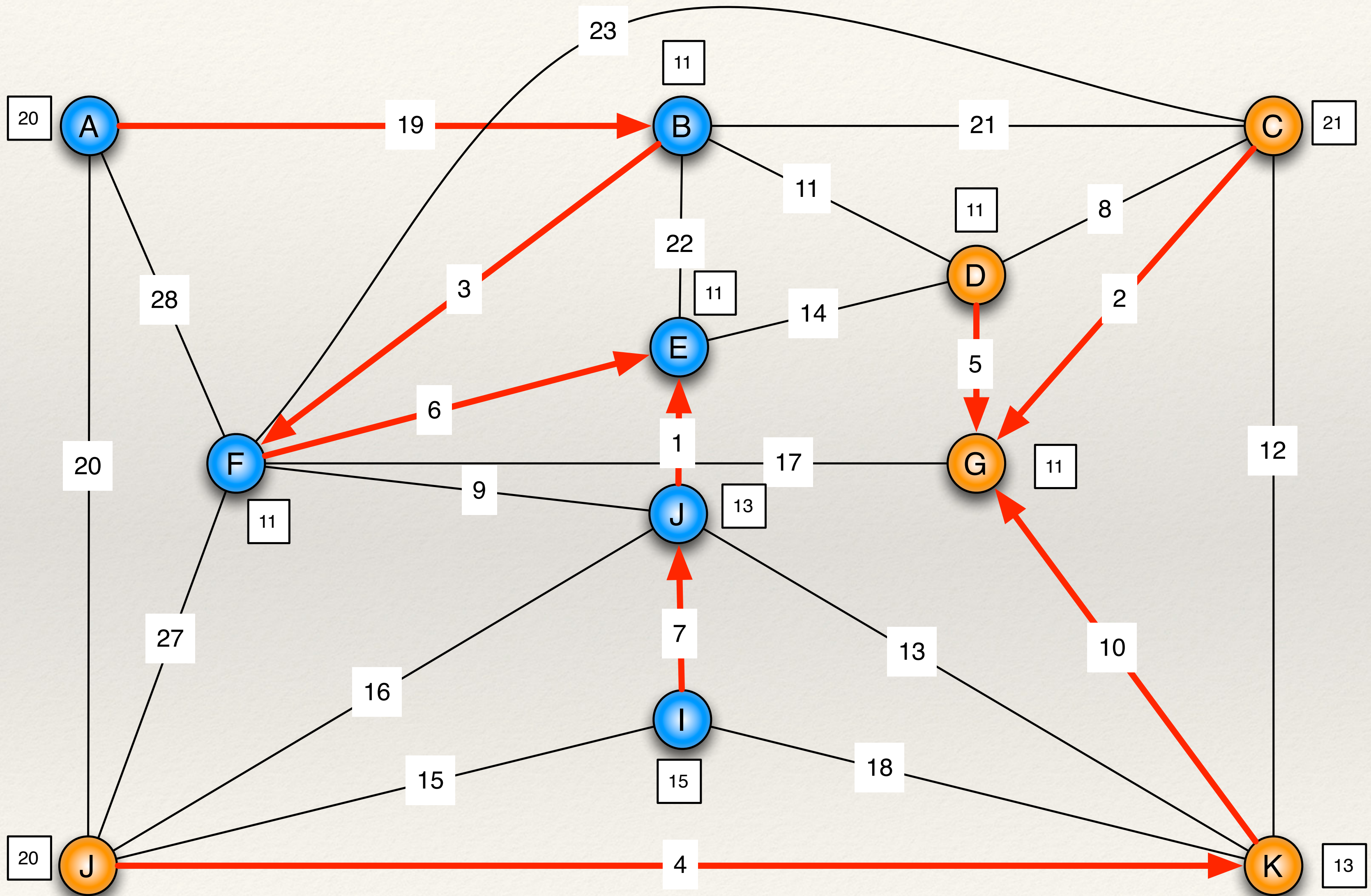
Exemple


23
6
10
A
19
B
21
C
20
11
8
22
11
3
D
14
28
E
2
6
5
6
1
20
17
12
F
G
10
6
9
H
9
27
16
7
13
10
15
15
I
18
4
15
J
K
10

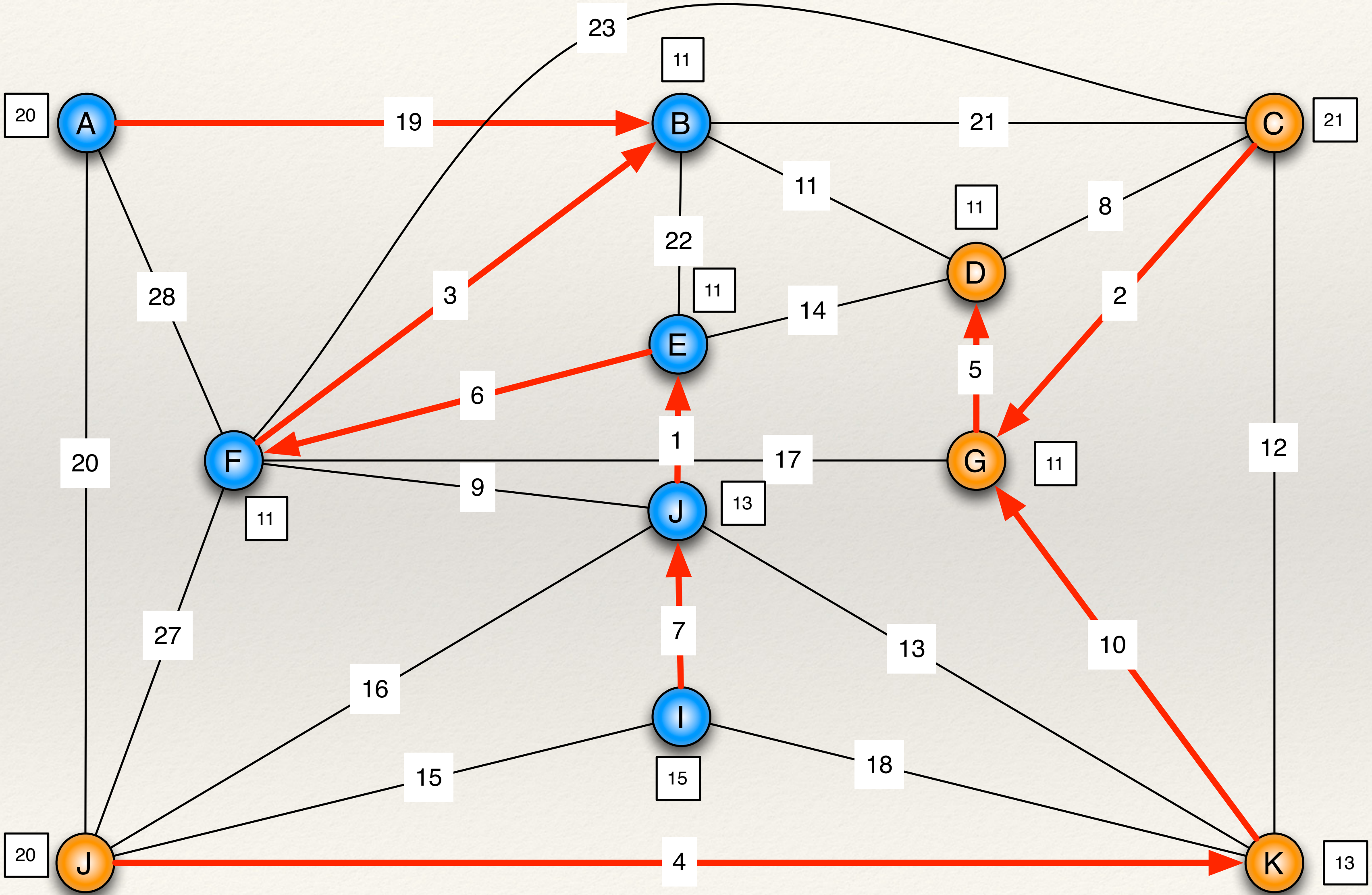
Exemple


23
19
21
A
B
C
11
8
22
D
3
14
28
E
2
5
6
1
20
17
12
F
G
9
H
27
16
7
13
10
I
15
18
4
J
K

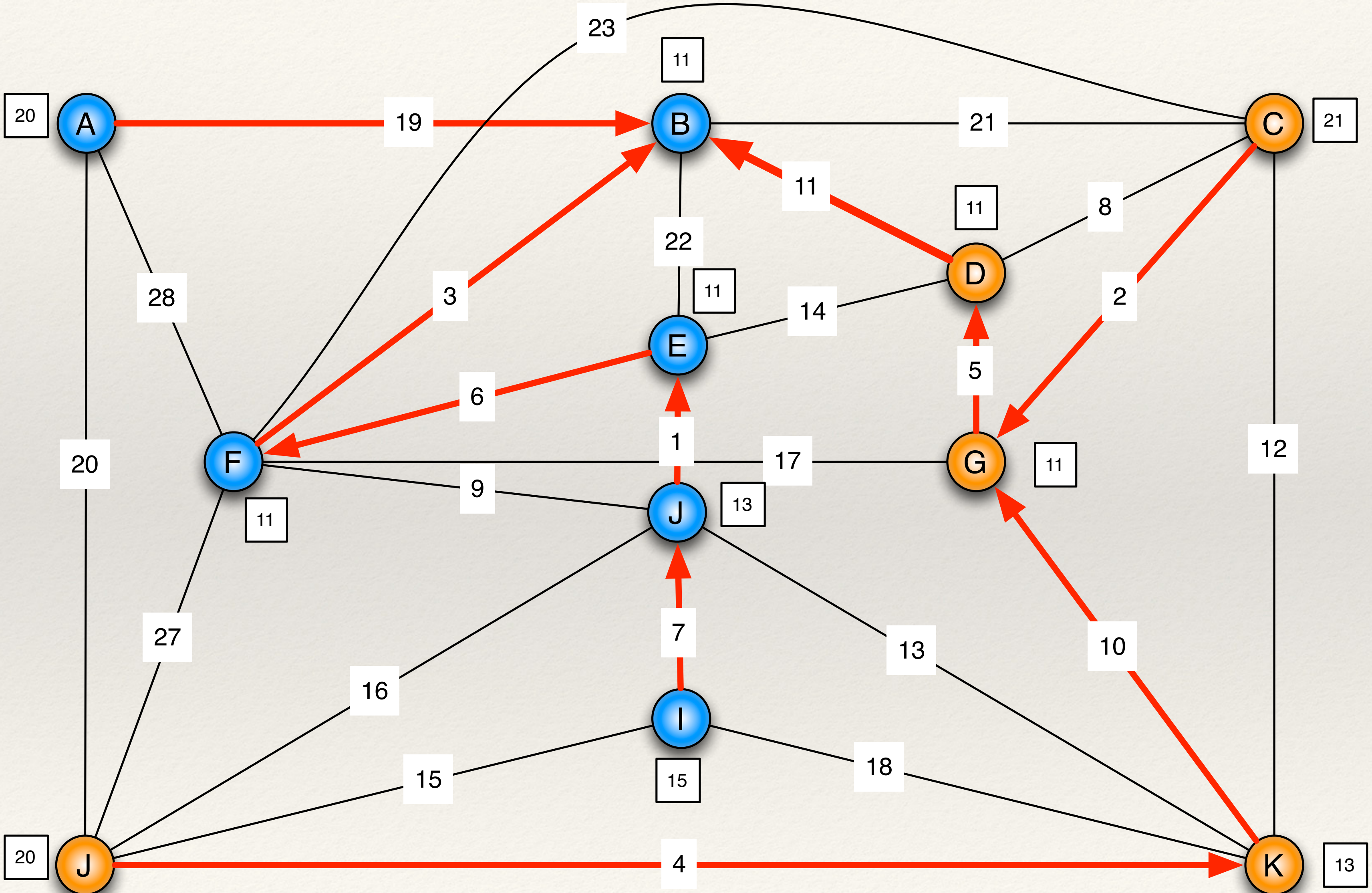
Exemple


23
11
20
A
19
B
21
C
21
11
11
8
22
D
28
3
11
2
14
E
5
6
1
17
G
11
20
F
9
J
13
12
11
27
7
13
10
16
I
18
15
15
20
J
4
K
13

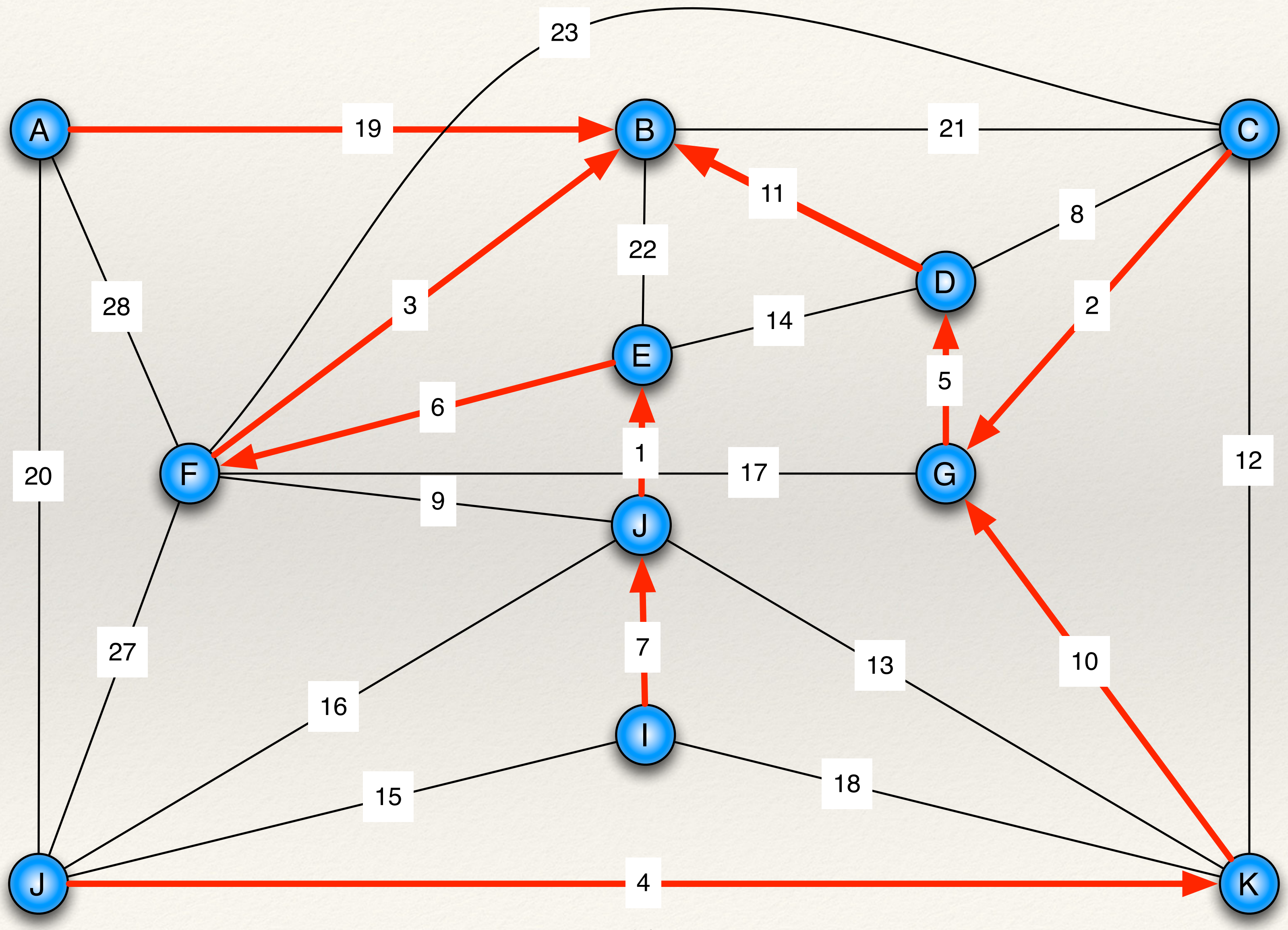
Exemple


23
11
20
A
19
B
21
C
21
11
11
8
28
3
22
D
11
2
14
E
6
5
1
20
F
17
G
11
12
9
J
13
11
27
7
13
10
16
I
15
18
15
20
J
4
K
13

Exemple


23
11
20
A
19
B
21
C
21
11
11
8
28
3
22
11
D
2
14
E
5
6
1
17
G
11
12
20
F
9
J
13
11
27
7
13
10
16
I
18
15
15
20
J
4
K
13

Exemple


23
A
19
B
21
C
11
8
28
22
3
D
14
2
E
6
5
1
20
F
17
G
12
9
J
27
7
13
10
16
I
15
18
J
4
K

Complexité

- L'algorithme de Gallager, Humlet et Spira utilise
  - $5n\log_2 n+2m$ messages
  - n est le nombre de noeuds et m le nombre de liens
  - Chaque message est de taille $O(\log_2 n)$

Remarques

- Cet algorithme fonctionne en asynchrone.
- Dans la version présenté les subtilités liés à l'asynchrone on été masqués.
- Il faut notamment que les fragment « grossissent » à la même « vitesse » pour minimiser le nombre de message.