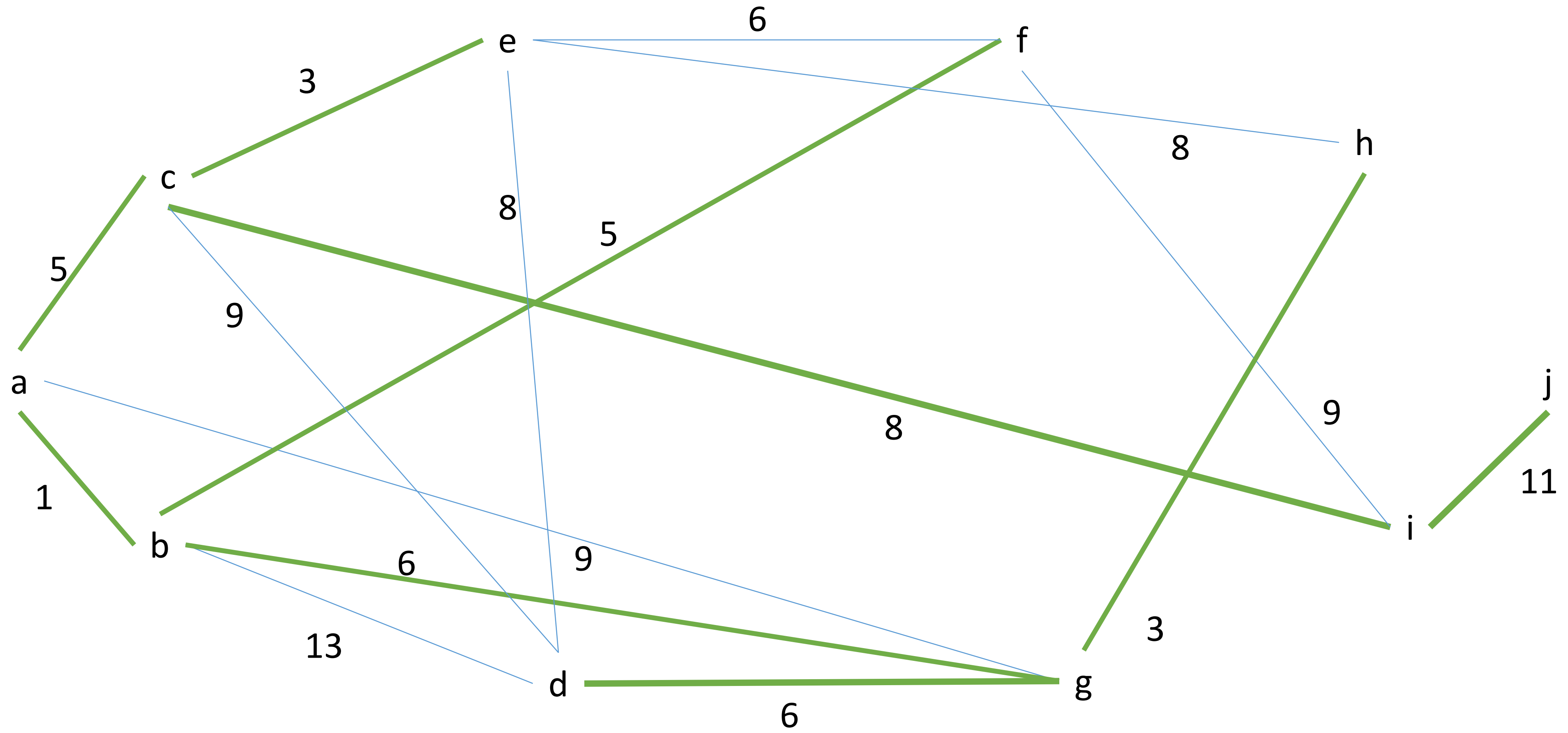
Algorithmique des Graphes
TD 4

Propositions de Solutions

Alain Cournier

Exercice 1



Exercice 1 Prim

- Voici un ordre possible pour l'insertion des arêtes si nous prendrons a comme sommet de départ : ab, ac, ce, bf, bg, gh, gd, ci, ij.
- Attention cet ordre n'est pas unique en voici un autre : ab, bf, ac, ce, bg, gh, gd, ci, ij

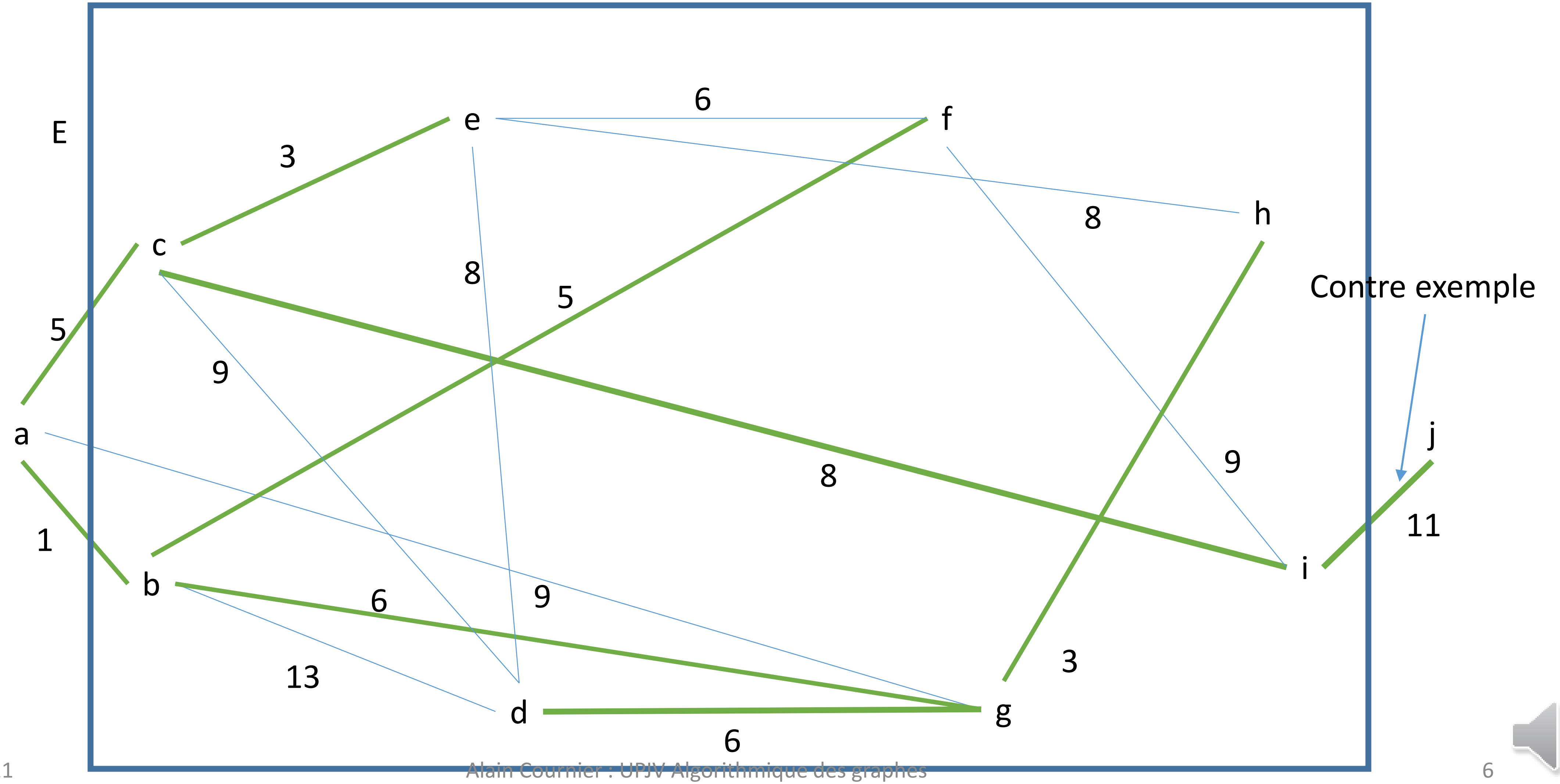
Eléments

Set	Inter	Choix
a	(b,ab,1);(c,ac,5);(g,ag,9)	(b,ab,1)
a,b	(d,bd,13);(c,ac,5);(g,bg,6);(f,bf,5)	(c,ac,5)
a,b,c	(d,cd,9);(e,ce,3);(g,bg,6);(f,bf,5);(i,ci,8)	(e,ce,3)
a,b,c,e	(d,ed,8);(h,eh,8);(g,bg,6);(f,bf,5);(i,ci,8)	(f,bf,5)
a,b,c,e,f	(d,ed,8);(h,eh,8);(g,bg,6);(i,ci,8)	(g,bg,6)
a,b,c,e,f,g	(d,gd,6);(h,gh,3);(i,ci,8)	(h,gh,3)
a,b,c,e,f,g,h	Je vous laisse finir	

Exercice 1 Kruskal

- Le tri des arêtes donne ab,ce,gh,bf,ac,bg,gd,ed,ci,eh,fi,cd,ag,ij
- Voici un ordre possible pour l’insertion des arêtes : ab, ce, gh, bf, ac, bg, gd, ci, ij.
- Attention cet ordre n’est pas unique en voici un autre : ab, gh, ce, ac, bf, gd, bg, ci, ij.

Exercice 2



Exercice 3

- L'algorithme de Kruskal tient sa complexité du tri des arêtes.
- Si toutes les valeurs sont entières Il est possible de réaliser ce tri en temps linéaire par un tri par base.
- La complexité devient alors liée à la gestion des composantes connexes : $n+m\ \alpha(n,m)$

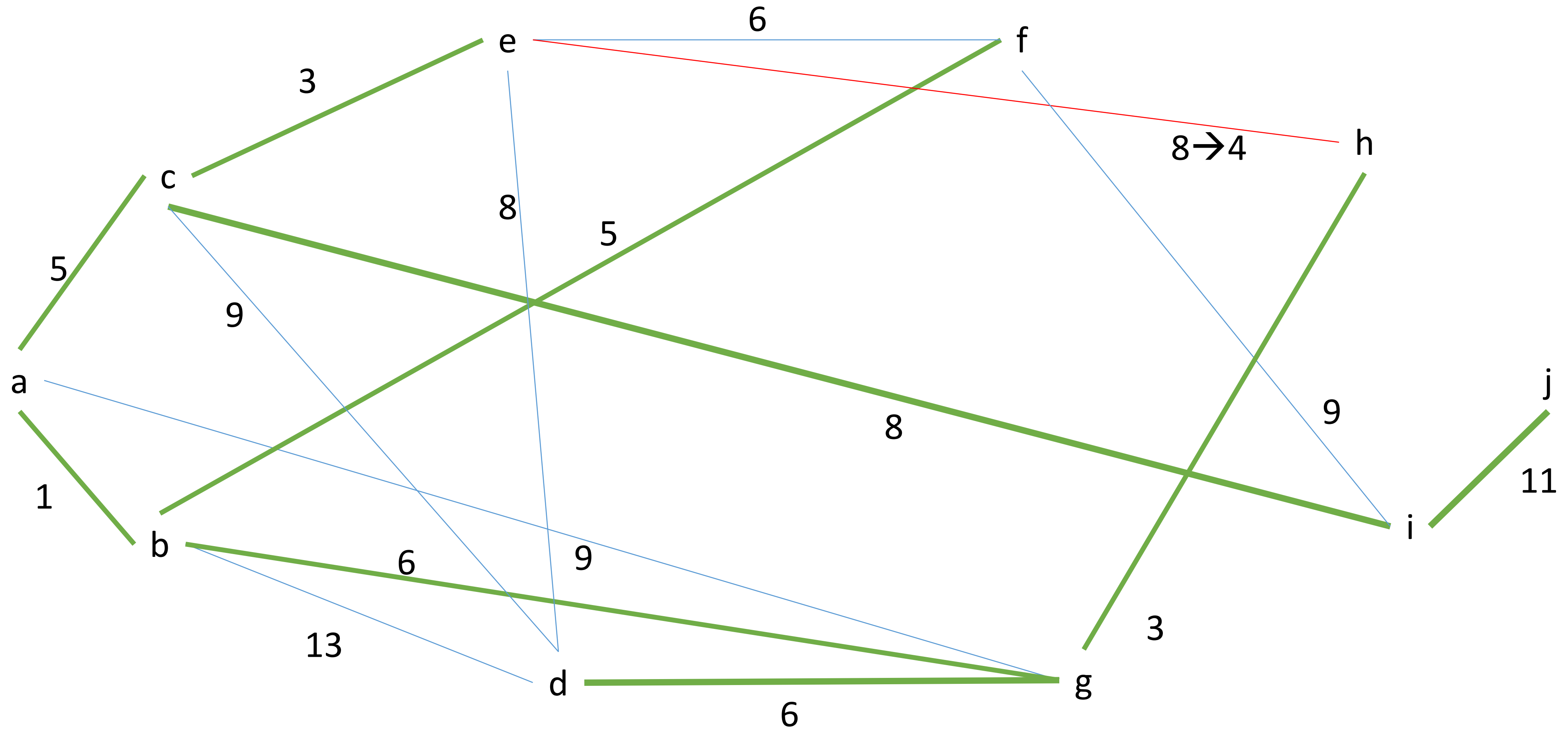
Exercice 4

- Soit ab l’arête modifiée V1(ab) son poids avant et V2(a,b) après
- Il faut distinguer 4 cas :
  - ab est dans l’arbre couvrant et V1(ab) >= V2(ab) RAS
  - ab n’est pas dans l’arbre couvrant et V2(ab) >= V1(ab) RAS
  - ab n’est pas dans l’arbre couvrant et V1(ab) >= V2(ab) Cas 3
  - ab est dans l’arbre couvrant et V2(ab) >= V1(ab) Cas 4

Exercice 4 : Cas 3

- ab n’est pas dans l’arbre couvrant et V1(ab) >= V2(ab)
  - Si on ajoute ab dans l’arbre on crée un cycle et sur ce cycle on retire l’arête de poids maximal (voir cours principe 2) O(n) dans le pire des cas
  - Un moyen faire un parcours en profondeur à partir de a dans l’arbre couvrant que l’on arrête dès que l’on rencontre b, dans la pile nous avons toutes les arêtes de la chaîne

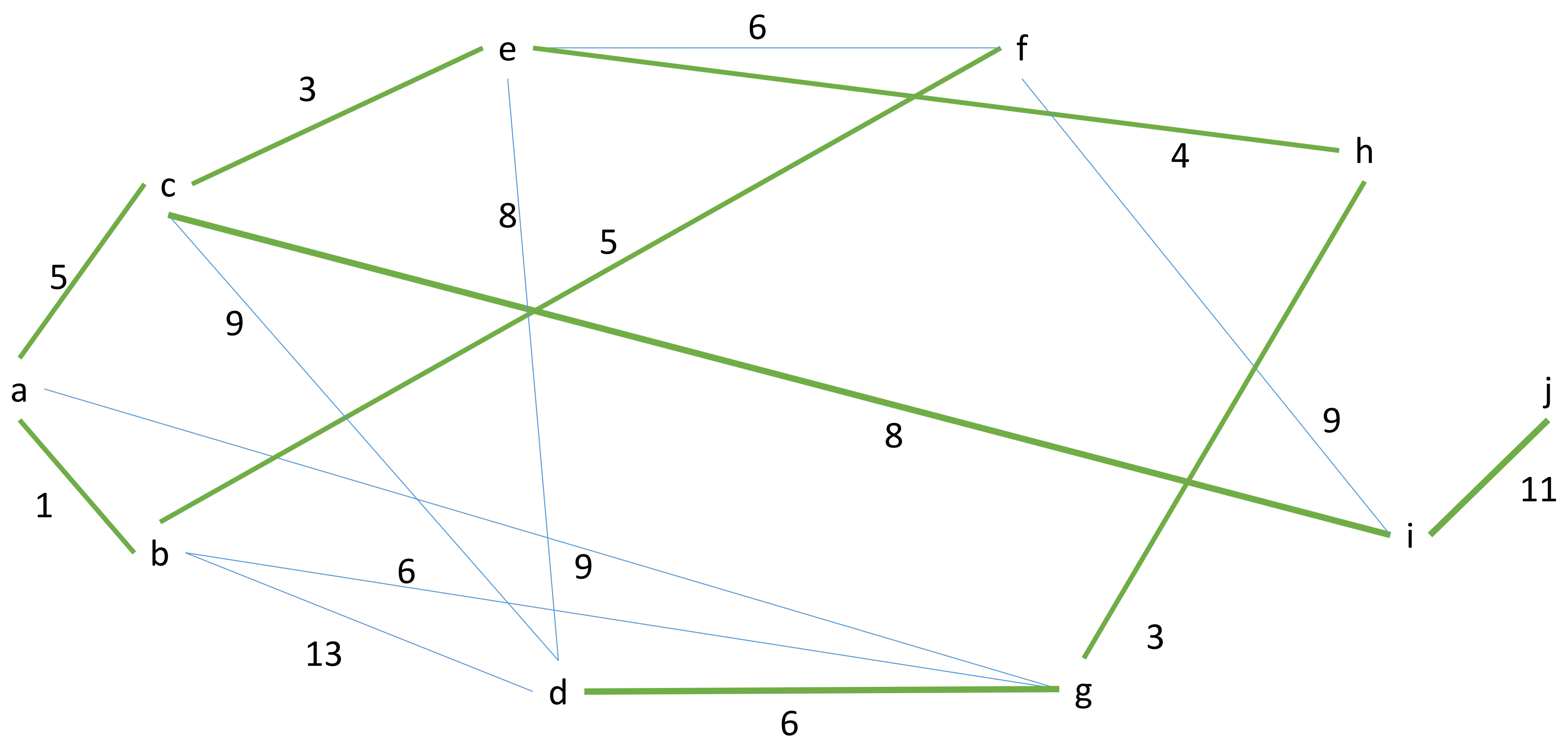
Exercice 4 cas 3



Exercice 4 : cas 3

- L’arête eh (qui n’est pas dans l’arbre) modifie sa valeur de 8 à 3
- On ajoute eh dans l’arbre on crée ainsi un cycle e,h,g,b,a,c,e
- Sur ce cycle bg est l’arête de poids maximal (poids 6)
- On retire bg de l’arbre pour obtenir le nouvel arbre couvrant

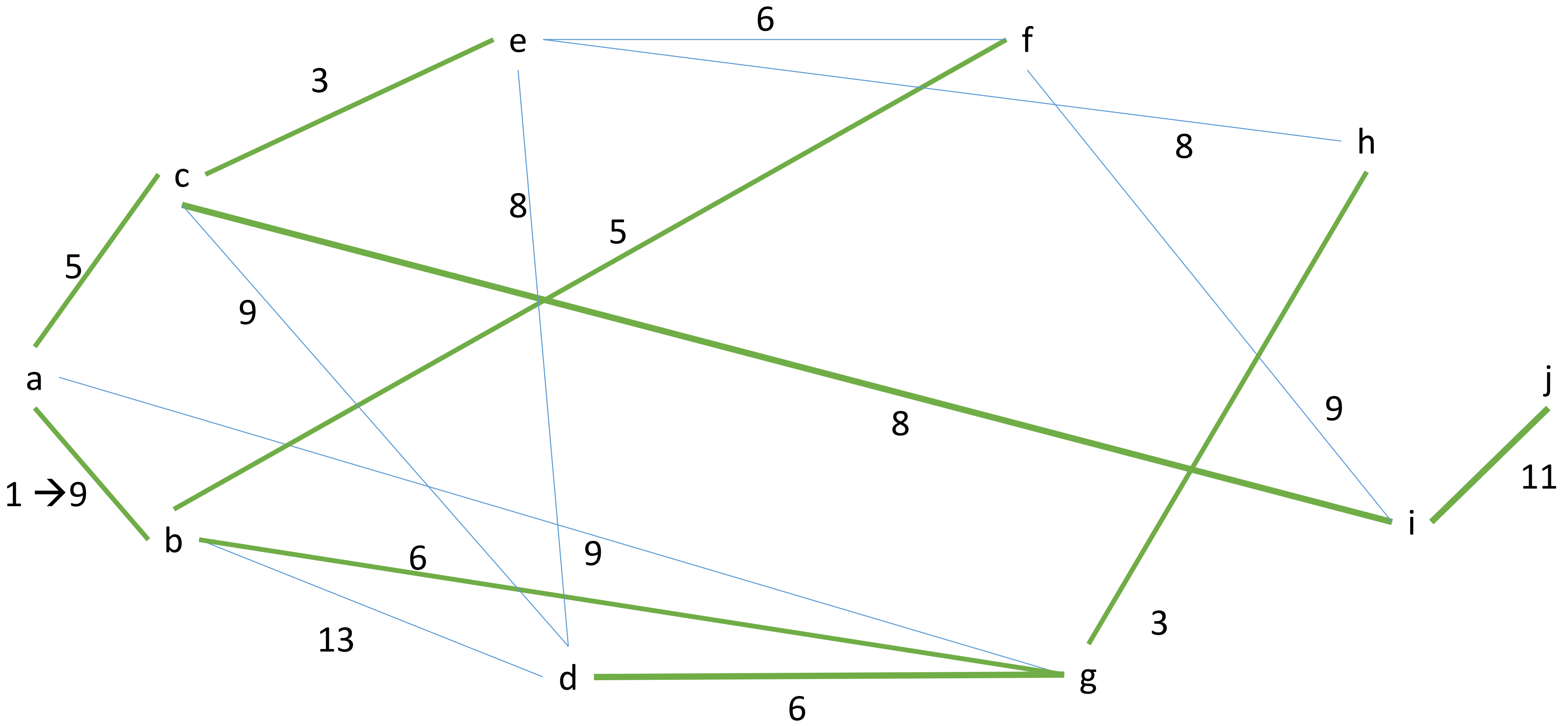
Exercice 4 cas 3



Exercice 4 Cas 4

- ab est dans l’arbre couvrant et V2(ab) >= V1(ab) Cas 4
  - On retire l’arête ab de l’arbre on partage l’arbre en deux composantes connexes CC1 et CC2 par un parcours
  - Dans le graphe on cherche le cocycle de CC1 CoCyCC1
  - Parmi les arêtes de CoCyCC1 on cherche l’arête de poids minimal uv
  - On insère uv dans l’arbre O(m)

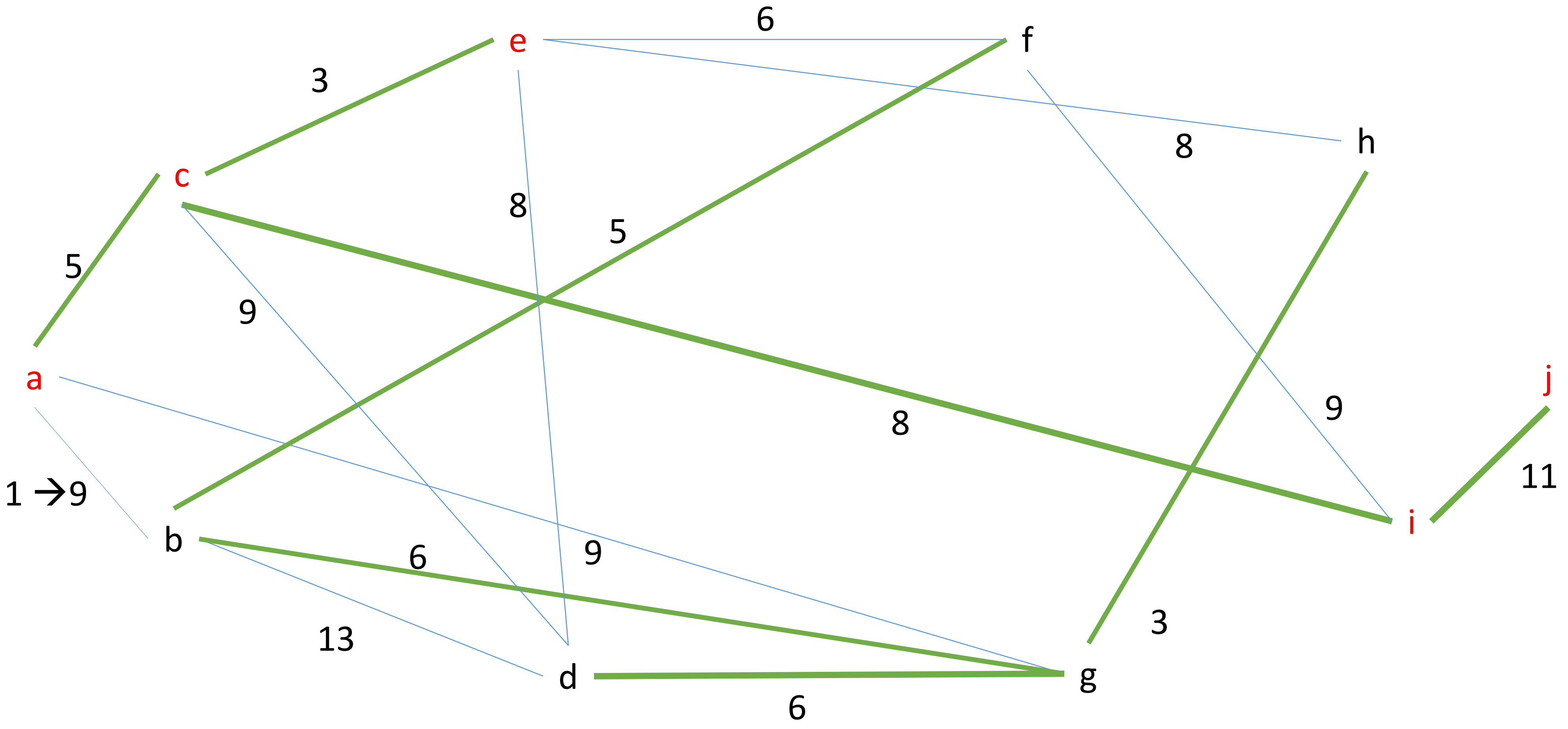
Exercice 4 cas 4



Exercice 4 : cas 3

- L’arête (qui est dans l’arbre) modifie sa valeur de 1 à 9
- On retire ab de l’arbre on crée ainsi deux composantes connexes {a,c,e,i,j} et {b,f,g,h,d}

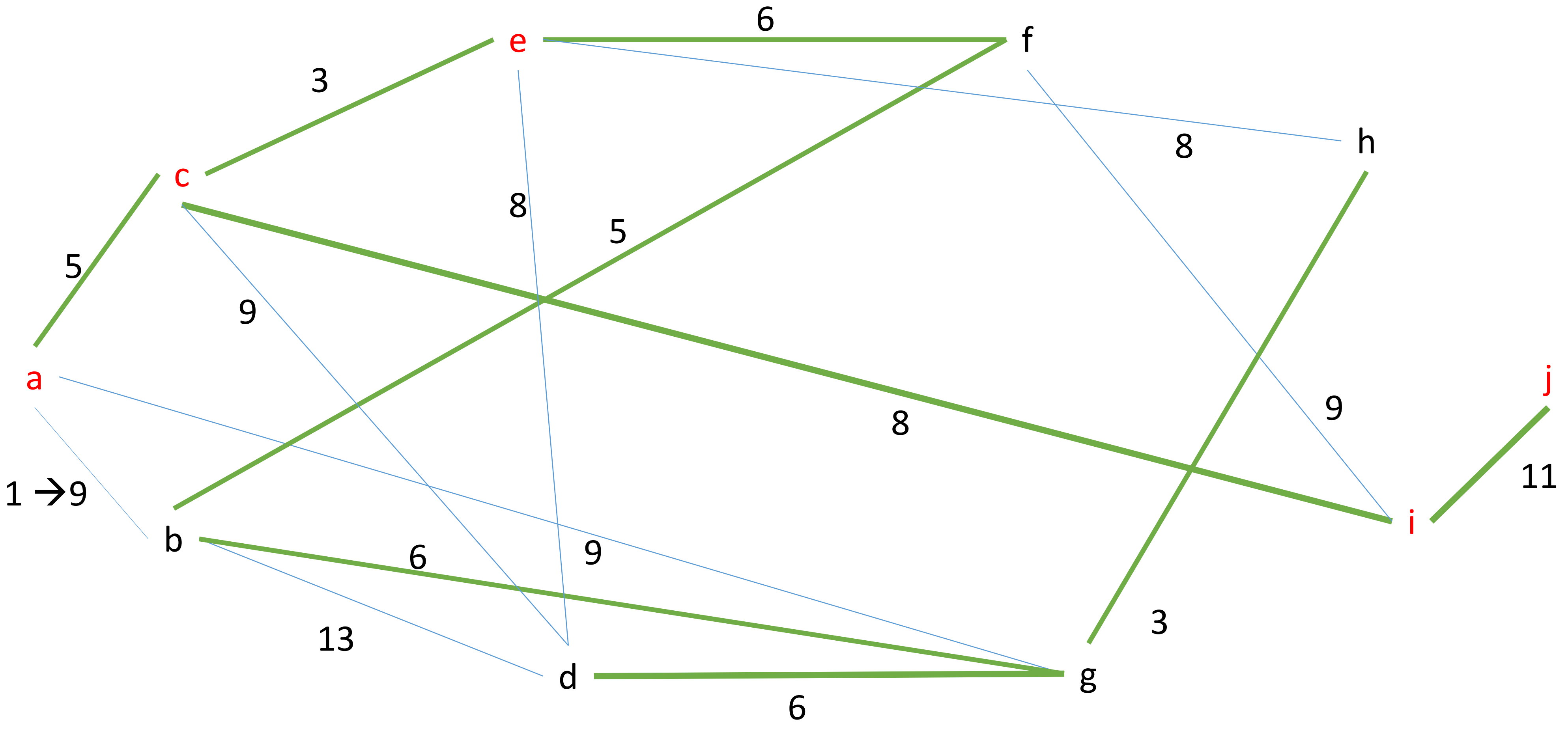
Exercice 4 cas 4



Exercice 4 : cas 3

- On cherche l'arête de poids minimal dans le cocycle de {a,c,e,i,j}
- C'est l'arête ef qui est minimale
- On ajoute cette arête ef dans l'arbre

Exercice 4 cas 4



Exercice 5 : CSS

- Chaque trou est modélisé par un sommet du graphe.
- Le coût (en gruyère) d’une galerie creusée entre deux trou est proportionnel à la distance qui sépare ces deux trous.
- On obtient ainsi un graphe complet (avec toutes les arêtes possibles) où chacune de ces arêtes possède un poids

- Les souris doivent alors construire l’arbre couvrant de poids minimal

Exercice 6 :

- Soit G=(X,U,V) le graphe initial sur lequel nous souhaitons construire un arbre couvrant de poids maximal.
- Construisons le graphe G’ = (X,U,V’) tel que :
  - Pour tout arête xy de U, v’(xy) = -v(xy)

- Je vous laisse faire la preuve que A l’arbre couvrant de poids minimal de G’ est aussi l’arbre de poids maximal de G
- Idée : Max(a,b,c) = - Min(-a,-b,-c)