Recherche Opérationnelle - Cours 6

Olivier Baudon

Université de Bordeaux

11 octobre 2022

Kruskal

Justification

Soit T_K un arbre obtenu par l'algorithme de Kruskal. Supposons que T_K ne soit pas de poids minimum. On note par $e_1, \dots e_{n-1}$ les arêtes de T_K telles que e_i est l'arête insérée à l'étape i.
Soit T un arbre de poids minimum tel que la première arête qui diffère entre T et T_K soit l'arête e_i avec i maximum.
Si on rajoute e_i à T, alors on crée un cycle fondamental C et donc il existe dans C une arête e_T n'appartenant pas à T_K. Cette arête a un poids $\omega(e_T) < \omega(e_i)$, sinon,

- si $\omega(e_T) > \omega(e_i)$, alors T ne serait pas optimal car on pourrait remplacer e_T par e_i ;
- si $\omega(e_T) = \omega(e_i)$, alors on pourrait remplacer e_T par e_i sans changer le poids de T et i ne serait maximum.

Donc quand on a choisit e_i, e_T était aussi candidate avec un poids inférieur et donc on aurait du choisir e_T.
Conclusion : T_K est de poids minimum.

Prim

Principe

L'algorithme de Prim construit un arbre T en rajoutant à chaque étape le sommet qui n'est pas encore dans l'arbre et qui possède l'arête de poids minimum parmi celles reliant les sommets de T aux sommets de $G - T$.

Énoncé

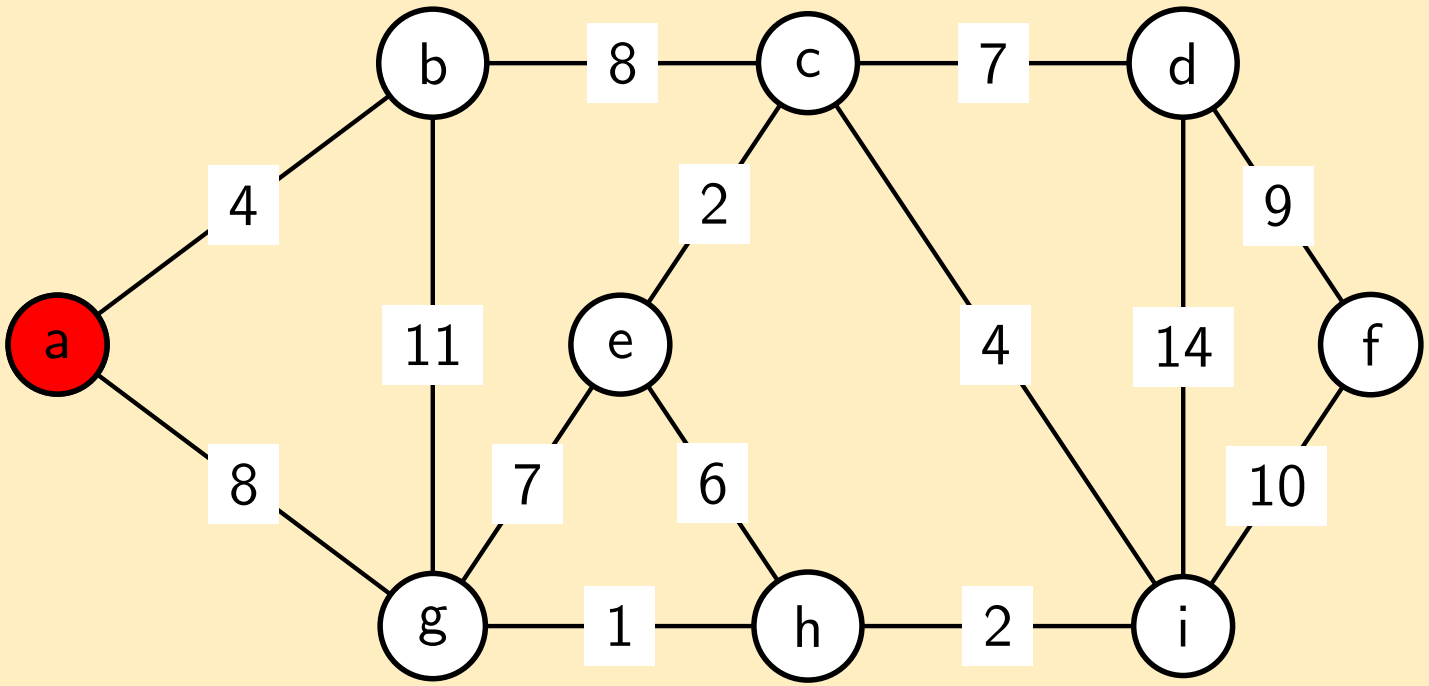
Voir le document "Algorithmes de graphes" page 11.

Complexité

L'algorithme de Prim est en $O(m + n \times log(n))$, à condition d'utiliser un tas de Fibonacci pour gérer les arêtes candidates à chaque étape.

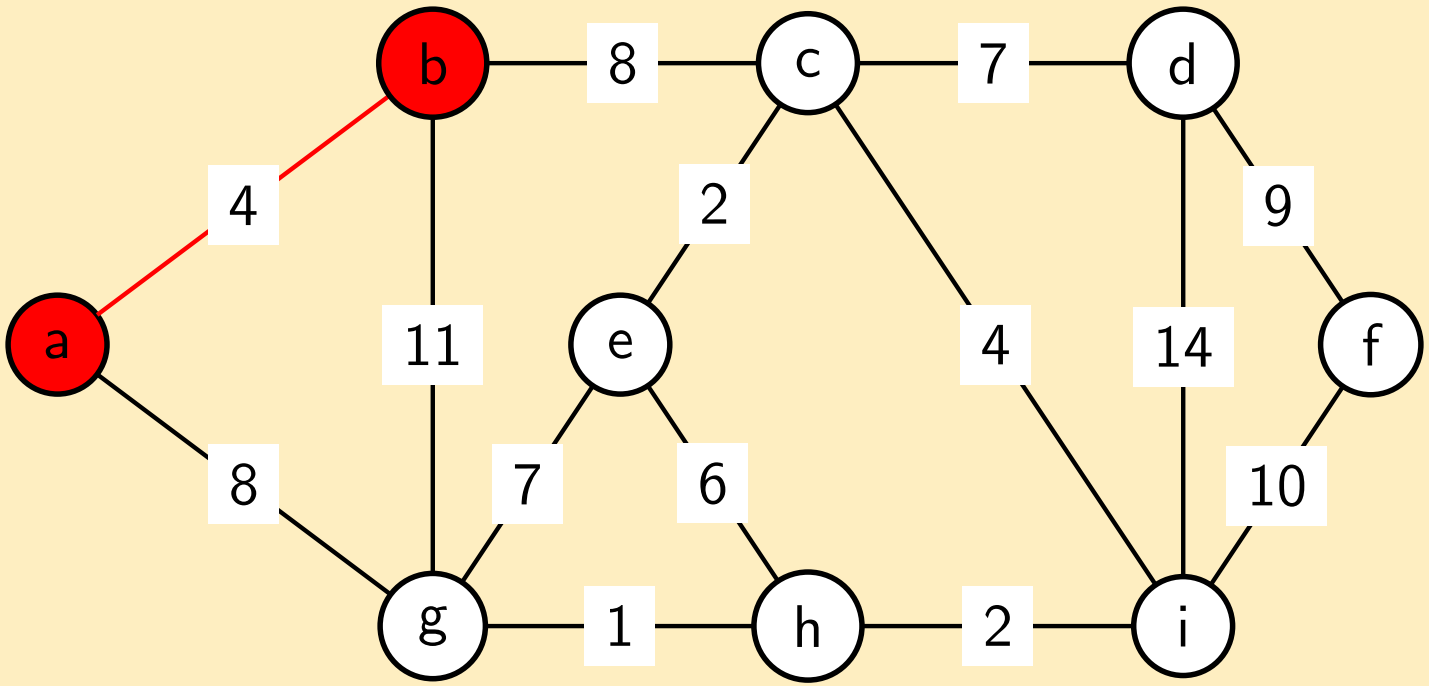
Prim

Exemple



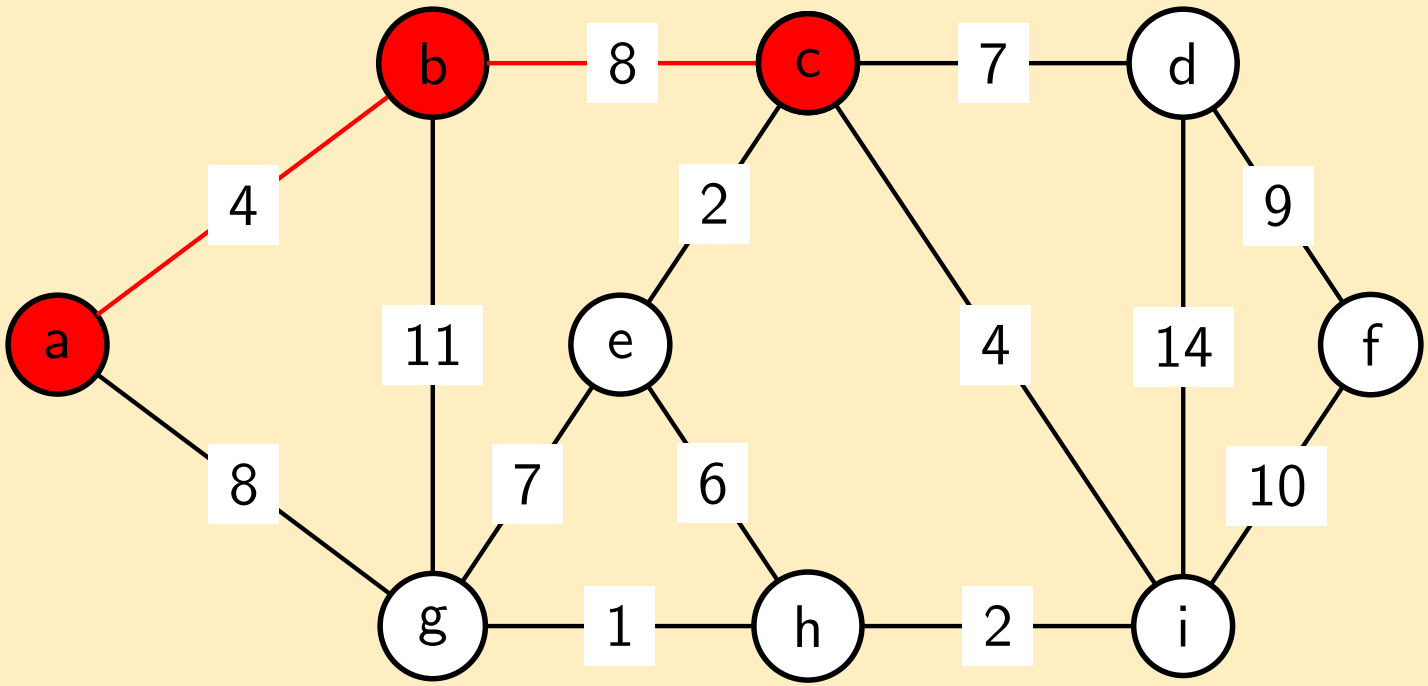
Prim

Exemple



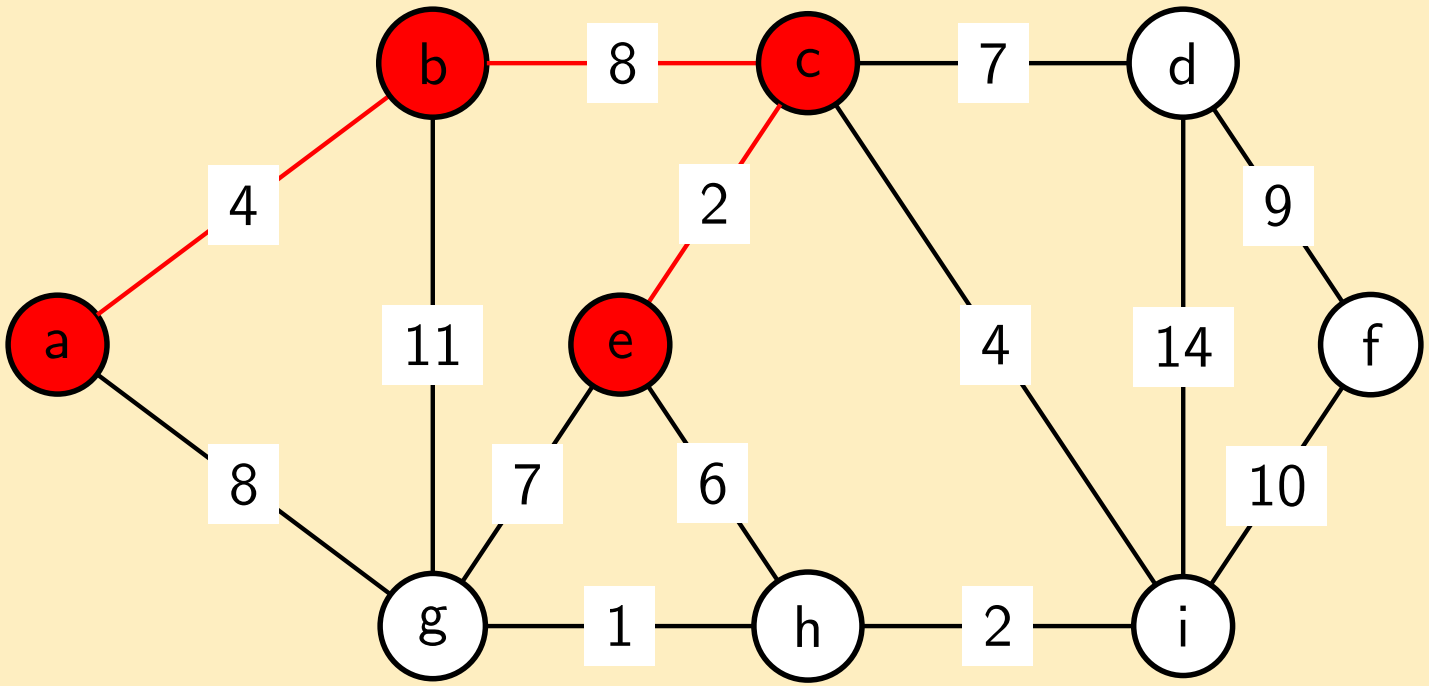
Prim

Exemple



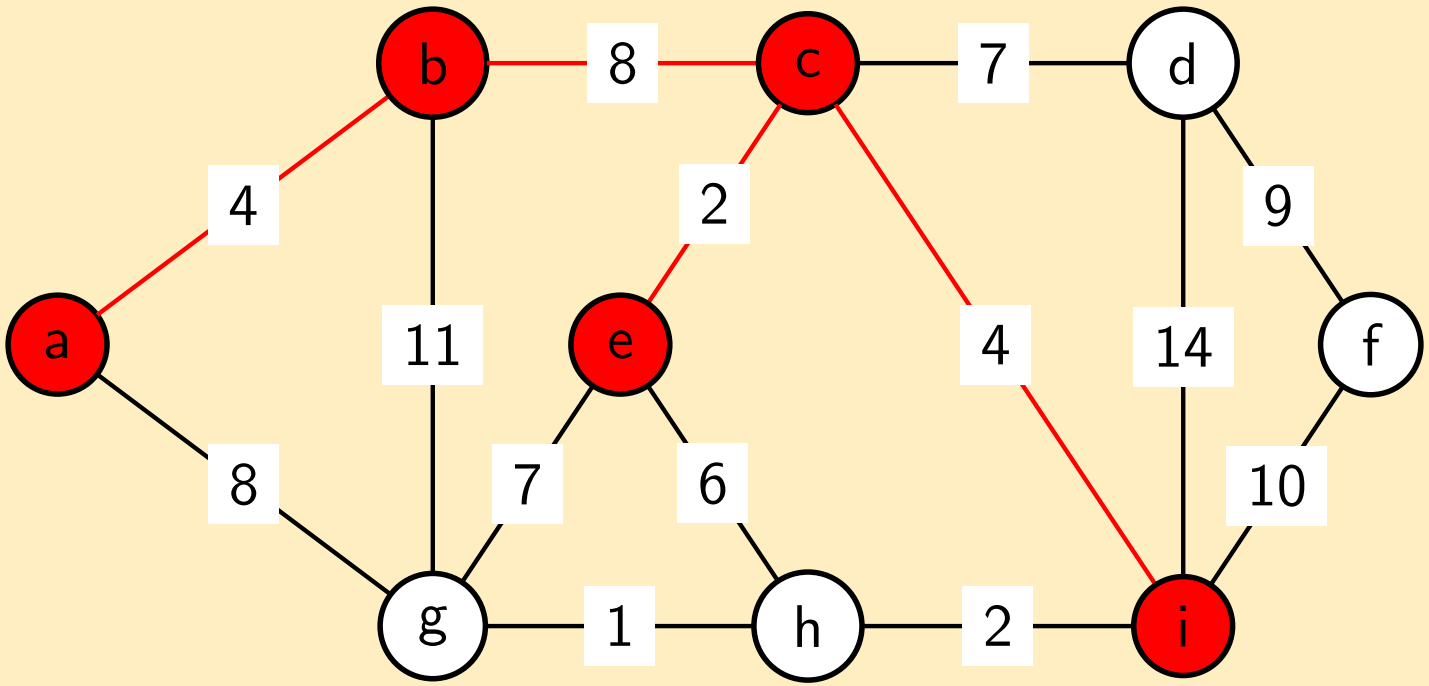
Prim

Exemple



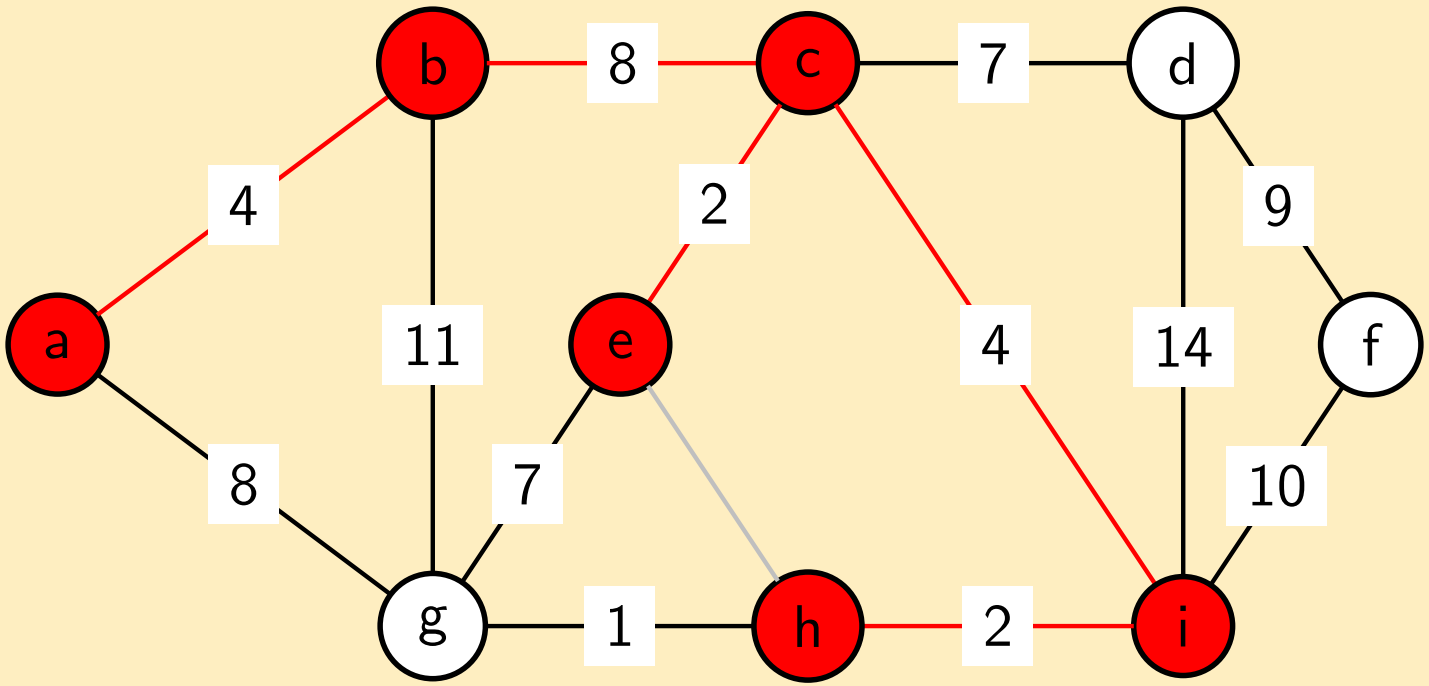
Prim

Exemple



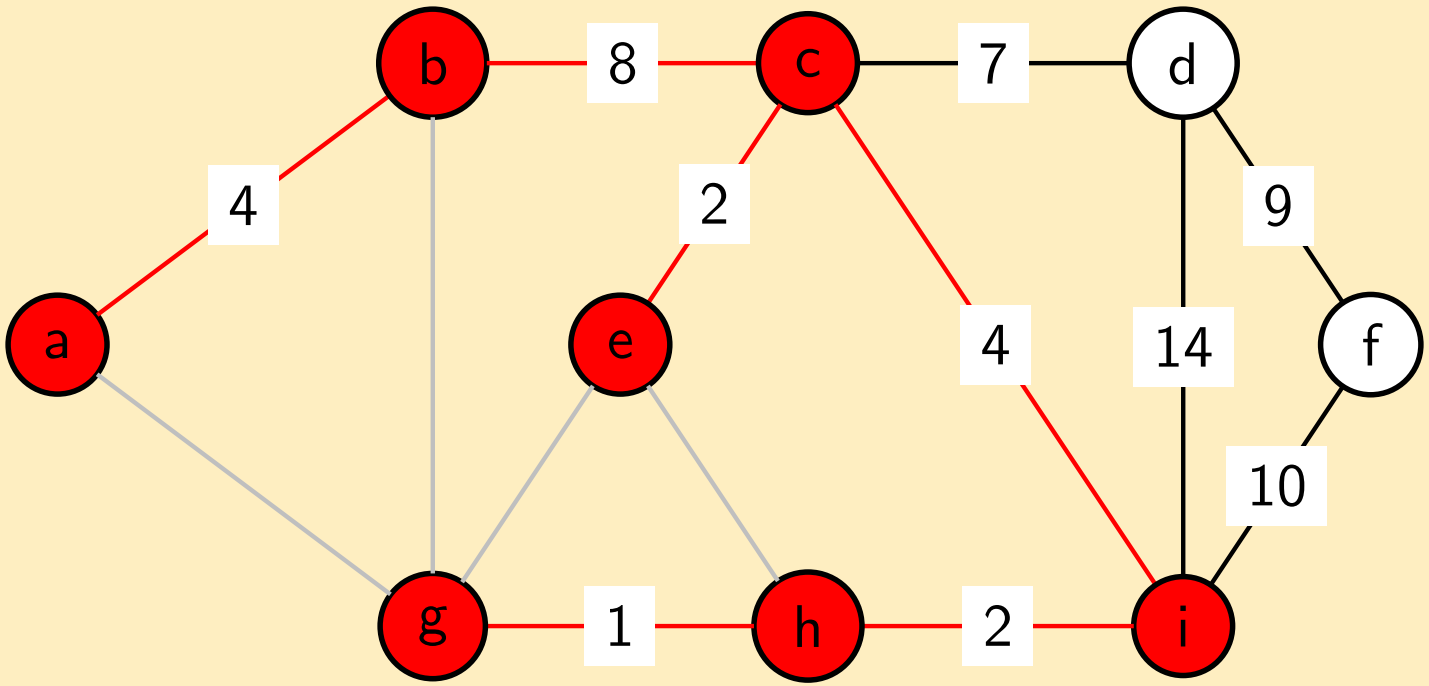
Prim

Exemple



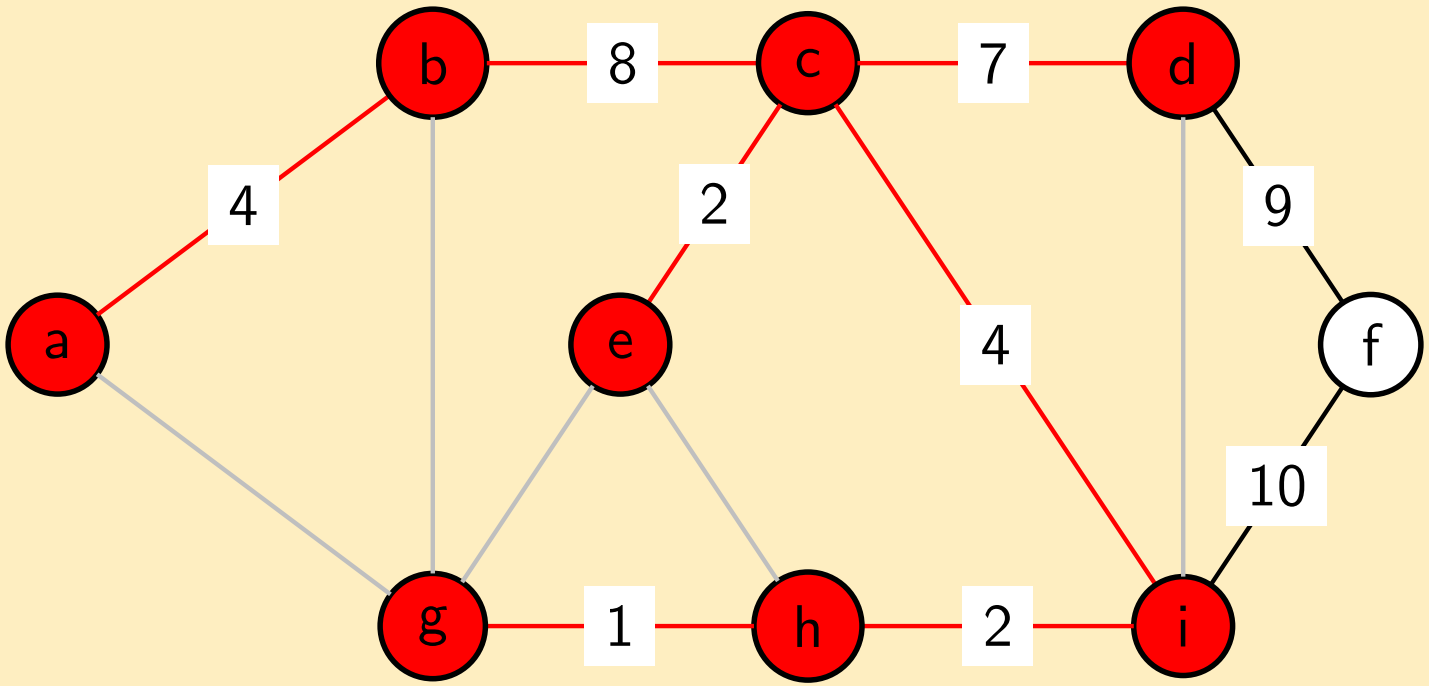
Prim

Exemple



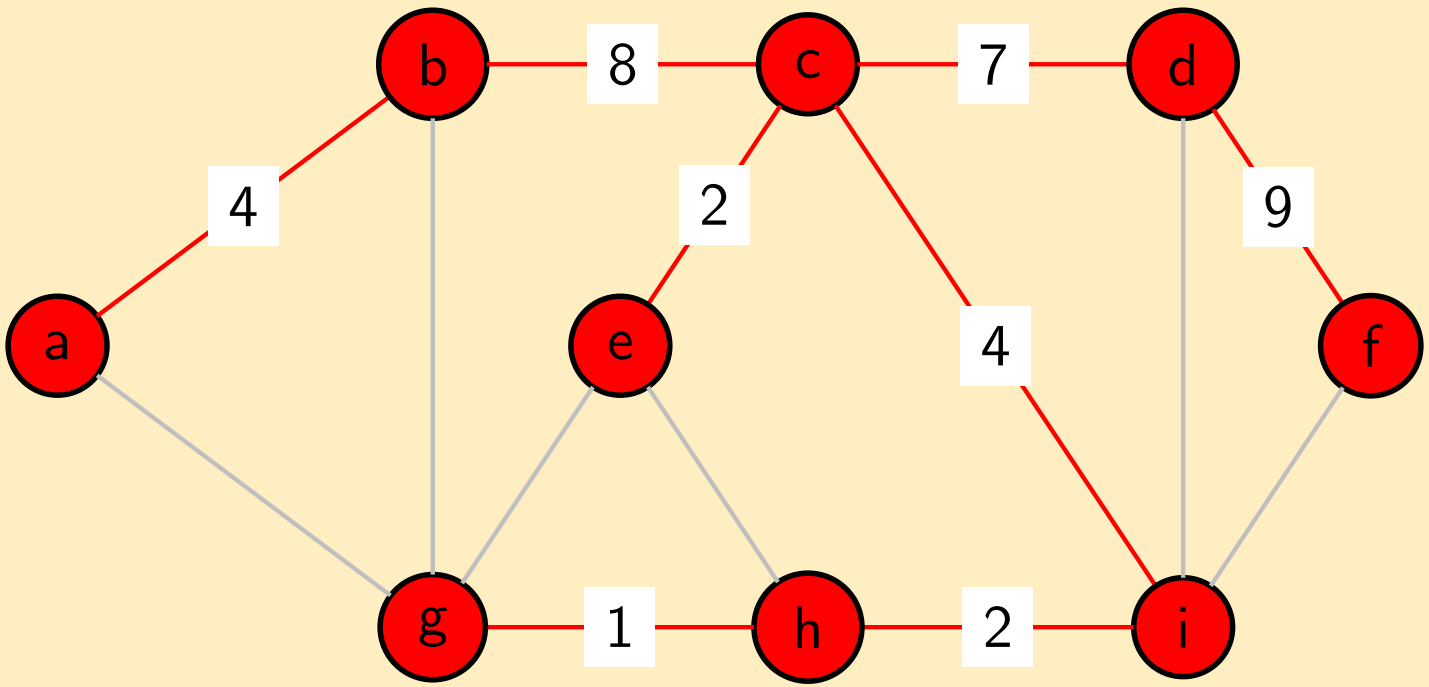
Prim

Exemple



Prim

Exemple



Poids total $= 4 + 8 + 2 + 7 + 9 + 4 + 2 + 1 = 37$

Prim

Justification

On construit un graphe connexe à n sommets et $n-1$ arêtes. Donc c'est un arbre.
Soit T_i l'arbre construit à l'étape i. On va montrer qu'il existe un arbre couvrant T^* de poids minimum tel que T^* contient T_i.
Pour $i = 0$, l'assertion est vraie.
Supposons qu'elle soit vraie à l'étape $i-1$. Soit T^* un arbre couvrant de poids minimum contenant T_{i-1} et notons A l'ensemble des sommets de T_{i-1}.
Soit $e = xy$ l'arête choisie par Prim à l'étape i, avec $x \in A$ et $y \notin A$.
Si $e \in T^*$, alors T_i est inclus dans T^*.
Sinon $\exists e' \in T^*$ avec $e' = x'y'$, $x' \in A$ et $y' \notin A$ et $e' \notin T_i$
Soit $T' = T^* + e - e'$. T' est un arbre de poids $\omega(T^*) + \omega(e) - \omega(e')$.
Or e a été choisie comme une arête de poids minimum entre A et $\bar{A}$.
Donc $\omega(e) \leq \omega(e')$ et donc $\omega(T') \leq \omega(T^*)$. Comme T^* est de poids minimum, on a $\omega(T') = \omega(T^*)$ et donc il existe bien un arbre de poids minimum contenant T_i.

Conclusion : T_{n-1} est de poids minimum.

Flots - problématique

Réseau

Un réseau est un quadruplet (G, s, t, c) où :

- G est un graphe orienté ;
- s et t sont deux sommets de G, appelés respectivement source et puits ;
- c est une fonction de $E(G) \to \mathbb{R}^+$, qui donne la capacité de chaque arc.

Flots - problématique

Flot

Un flot f dans un réseau (G, s, t, c) est une fonction de $E(G) \to \mathbb{R}^+$ telle que :

$$\forall e \in E(G), c(e) \geq f(e) \geq 0$$

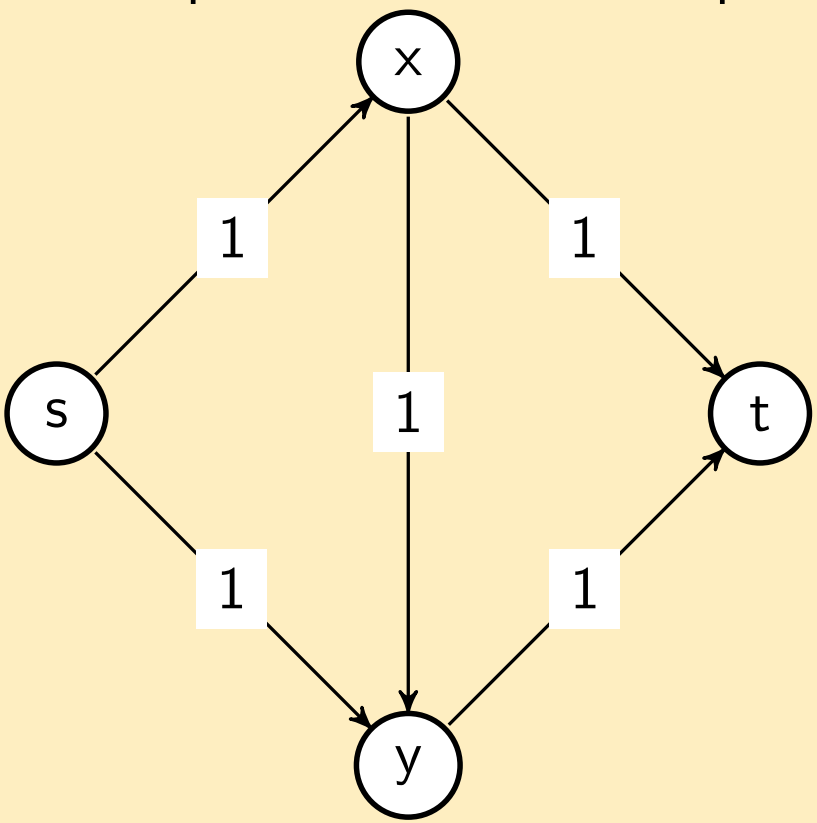
$$\forall v \in V(G) \backslash \{s, t\} \quad \sum_{e=uv \in E(G)} f(e) = \sum_{e=vw \in E(G)} f(e)$$

Remarque : la deuxième contrainte est appelée loi de Kirchhoff, en référence à la conservation d'énergie dans les circuits électriques, et s'exprime généralement par "tout ce qui rentre est égal à tout ce qui sort".
La valeur du flot, notée $val(f)$ sera égale à $\sum_{e=sv \in E(G)} f(e)$.
Remarque : on a également $val(f) = \sum_{e=vt \in E(G)} f(e)$.
L'objectif est de trouver pour un réseau (G, s, t, c) son flot de valeur maximum.

Flots - problématique

Exemple de réseau

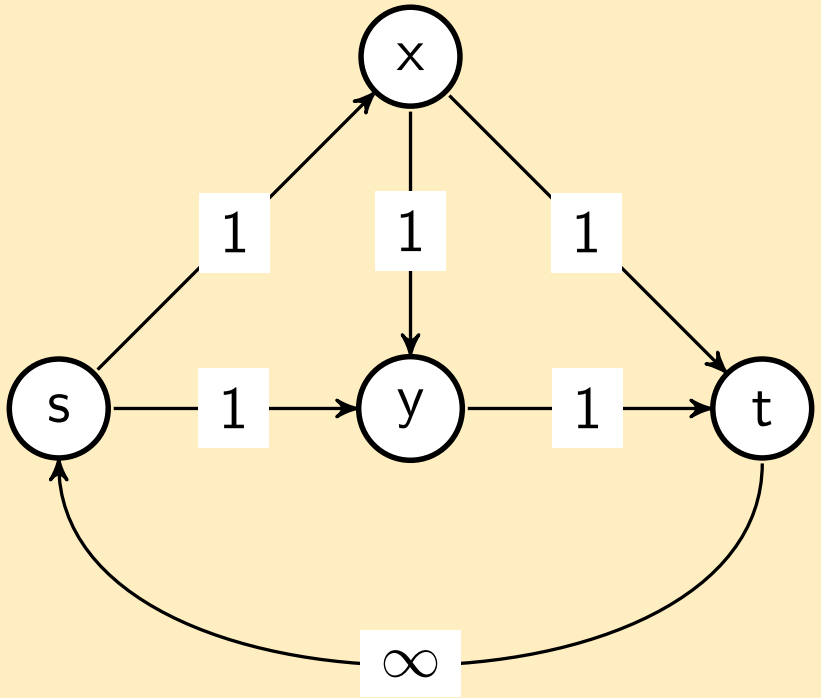
Les étiquettes sur les arcs représentent la capacité de chaque arc.



Flots - problématique

Arc retour

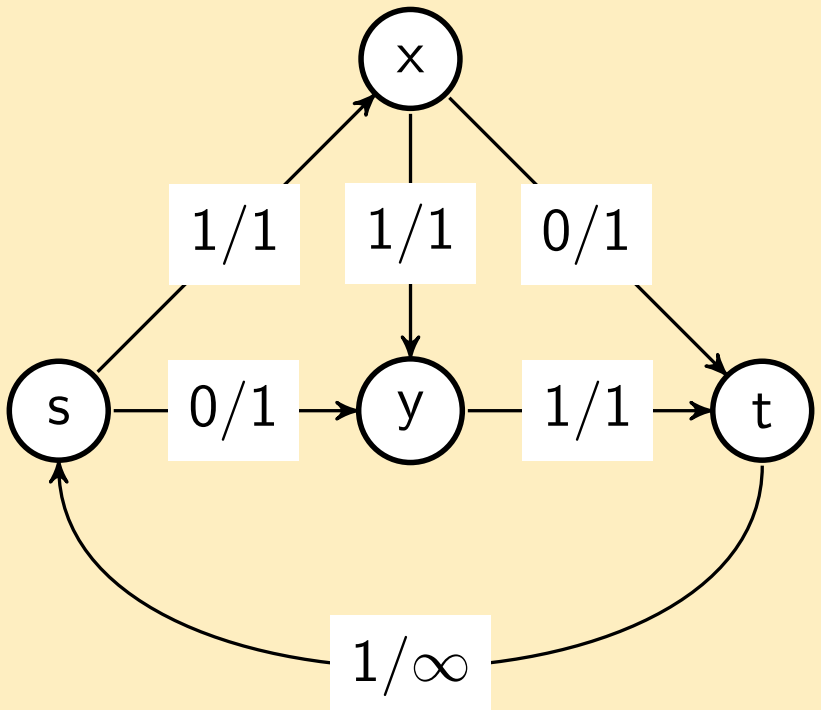
Si f est un flot, alors
$\sum_{e=sv \in E(G)} f(e) = \sum_{e=vt \in E(G)} f(e) = val(f)$.
Il est habituel de rajouter un arc, dit "de retour", de t vers s, de capacité infinie, et dont la valeur du flot est $val(f)$. Ainsi, la loi de Kirchhoff sera respectée par tous les sommets.



Flots - problématique

Exemple de flot réalisable

Sur chaque arc e, on note deux valeurs "$f(e)/c(e)$" représentant respectivement la valeur du flot et la capacité de l'arc e.
Le flot ci-dessous est de valeur 1.



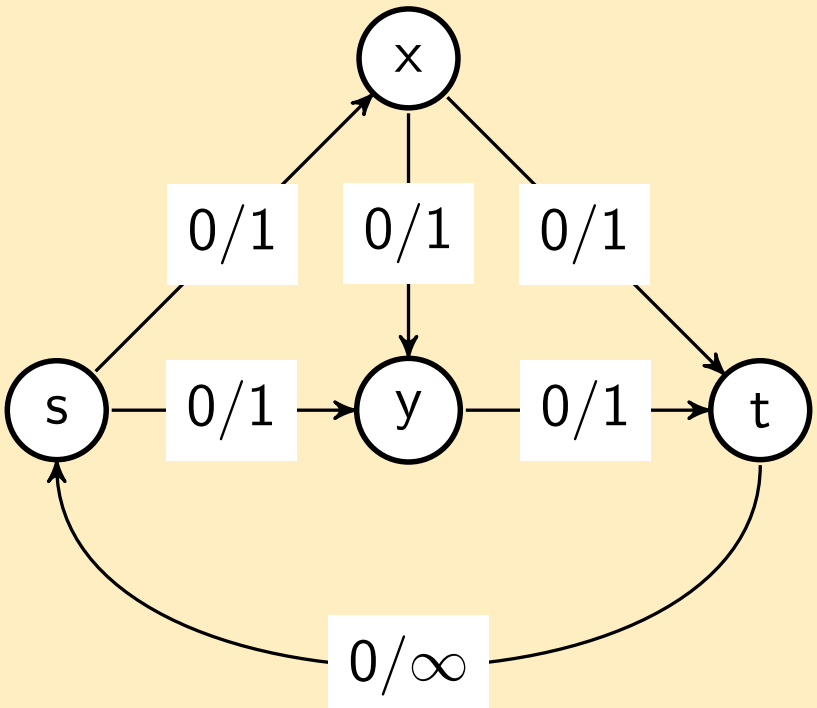
Flots - résolution

Formulation sous forme d'un programme linéaire

Le problème de flots peut s'exprimer à l'aide d'un programme linéaire : $max\{f(ts)|0 \leq f \leq c, Bf = 0\}$ où B est la matrice d'incidence de G augmenté de l'arc retour, f et c des vecteurs indicés par $E(G) \cup \{ts\}$ contenant respectivement la valeur du flot sur chaque arc et les capacités des arcs.
Rappel : $B_{ij} = -1$ si l'arc j est un arc sortant de i, 1 si j est un arc entrant de i, 0 sinon.
On peut donc résoudre le problème de flots à l'aide d'un algorithme de résolution d'un programme linéaire, tel que l'algorithme du simplexe.

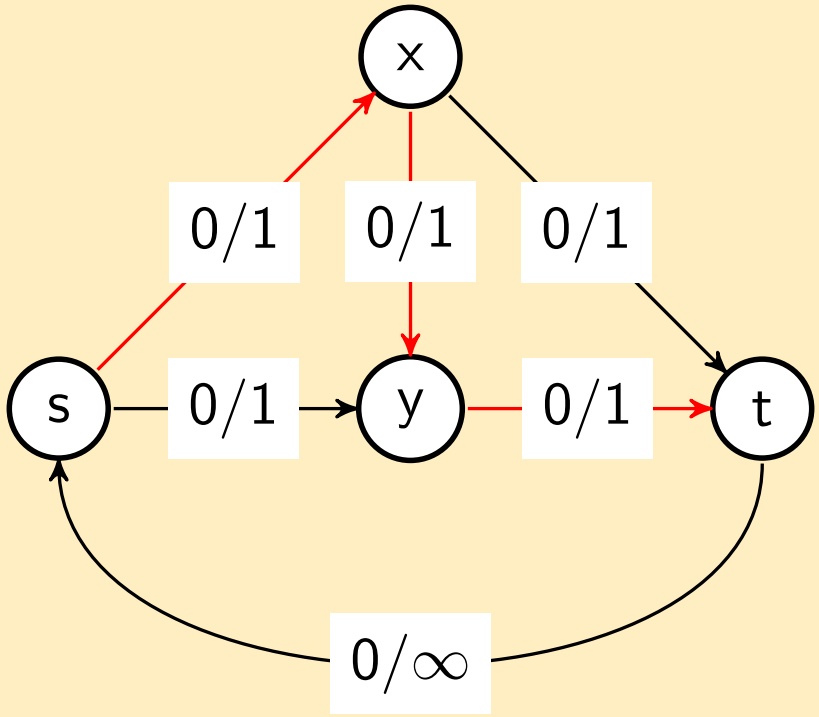
Flots - résolution

Recherche d'une chaîne augmentante par marquage direct



Flots - résolution

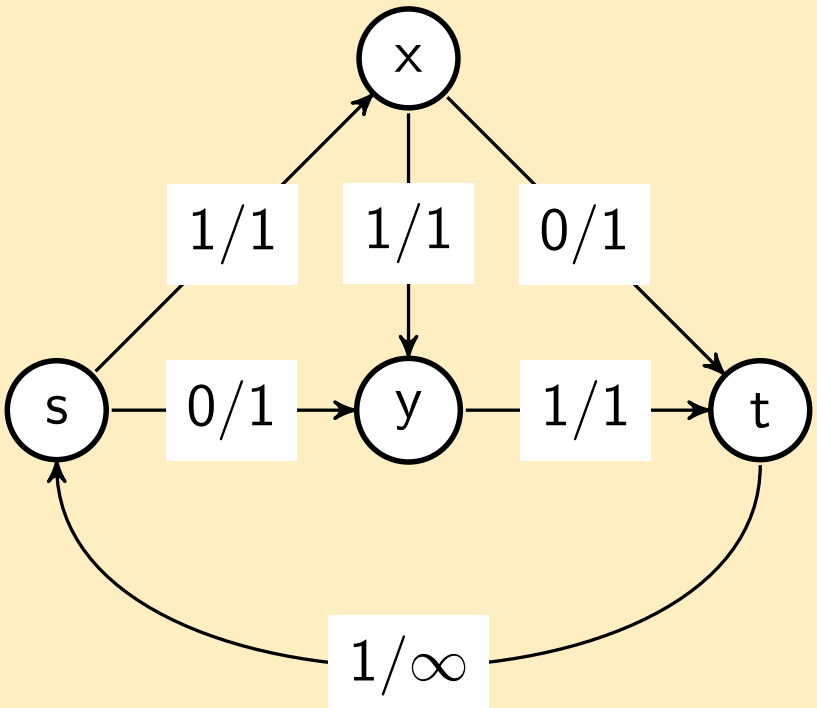
Recherche d'une chaîne augmentante par marquage direct



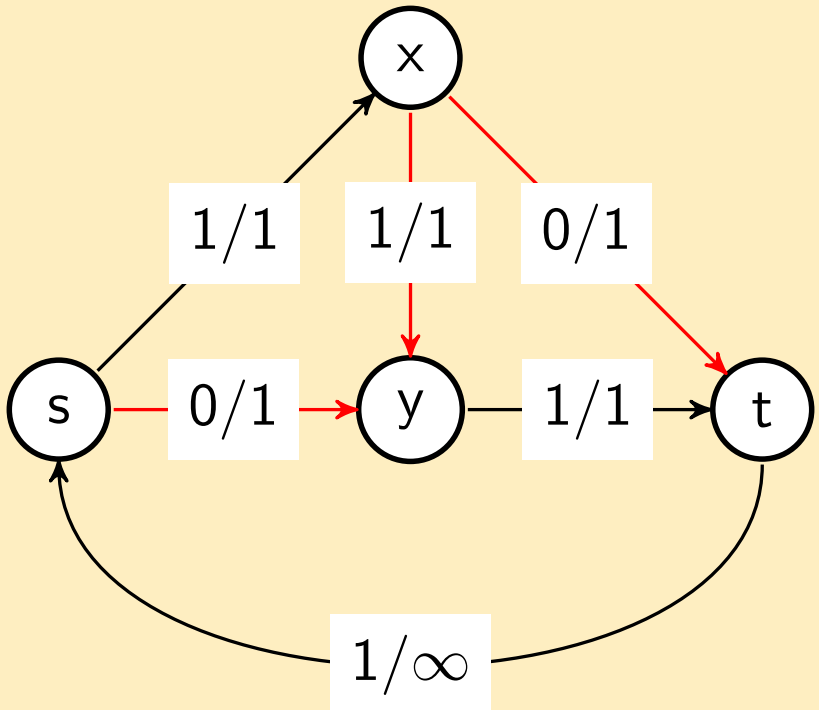
Chaîne augmentante : s, x, y, t. Augmentation : $+1$.

Flots - résolution

Recherche d'une chaîne augmentante par marquage indirect



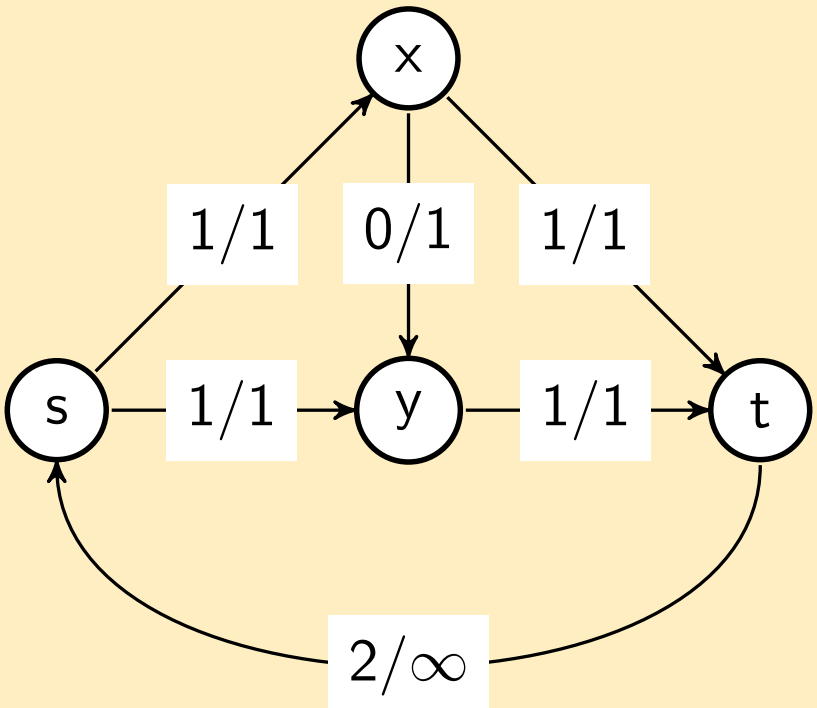
Recherche d'une chaîne augmentante par marquage indirect



Chaîne augmentante : s, y, x, t. Augmentation : $+1$.

Flots - résolution

Recherche d'une chaîne augmentante par marquage indirect



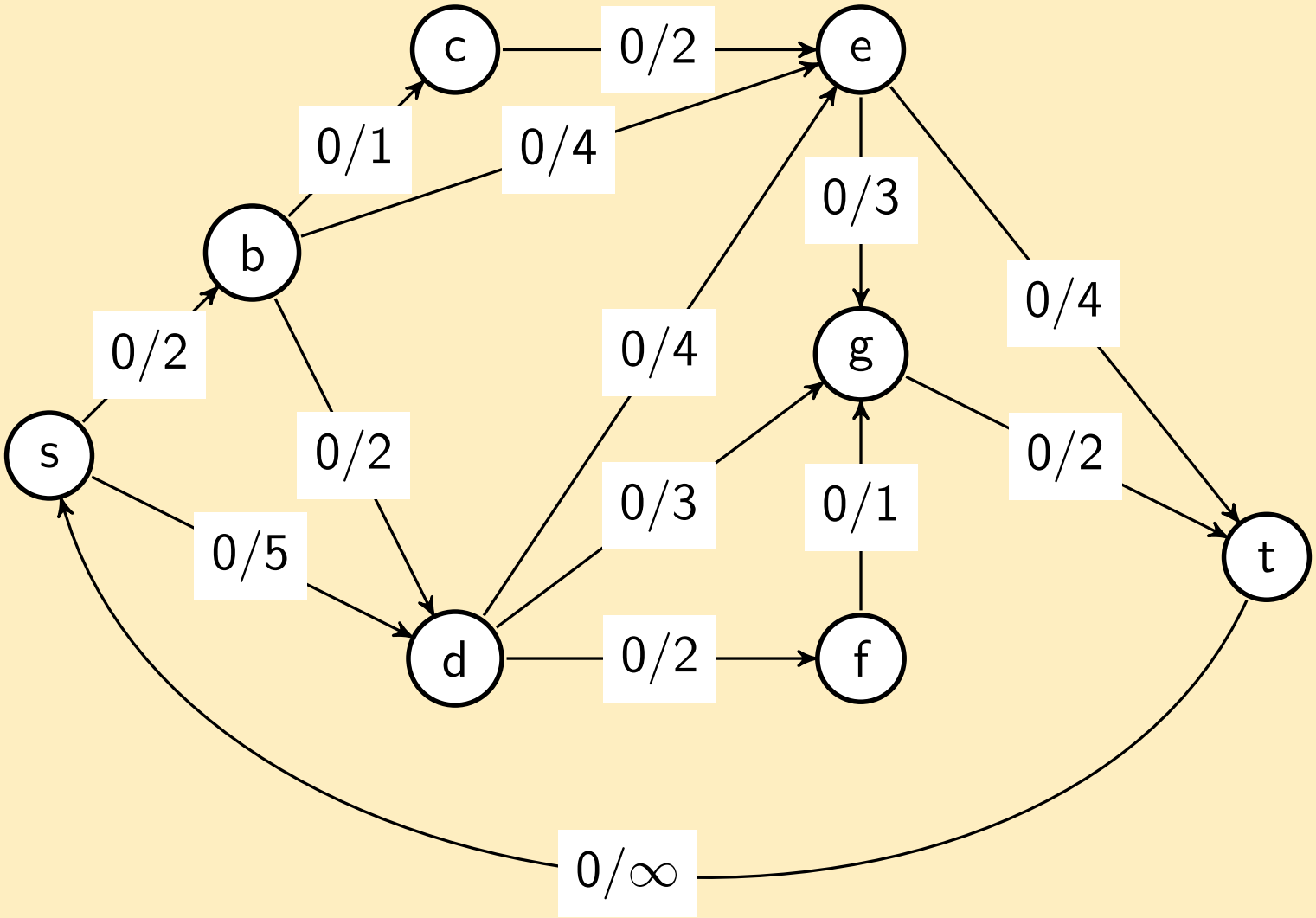
Flots - résolution

Algorithme de Ford-Fulkerson

Voir document "Algorithmes des graphes" pages 12 et 13.

Flots - résolution

Exemple (Ford-Fulkerson)

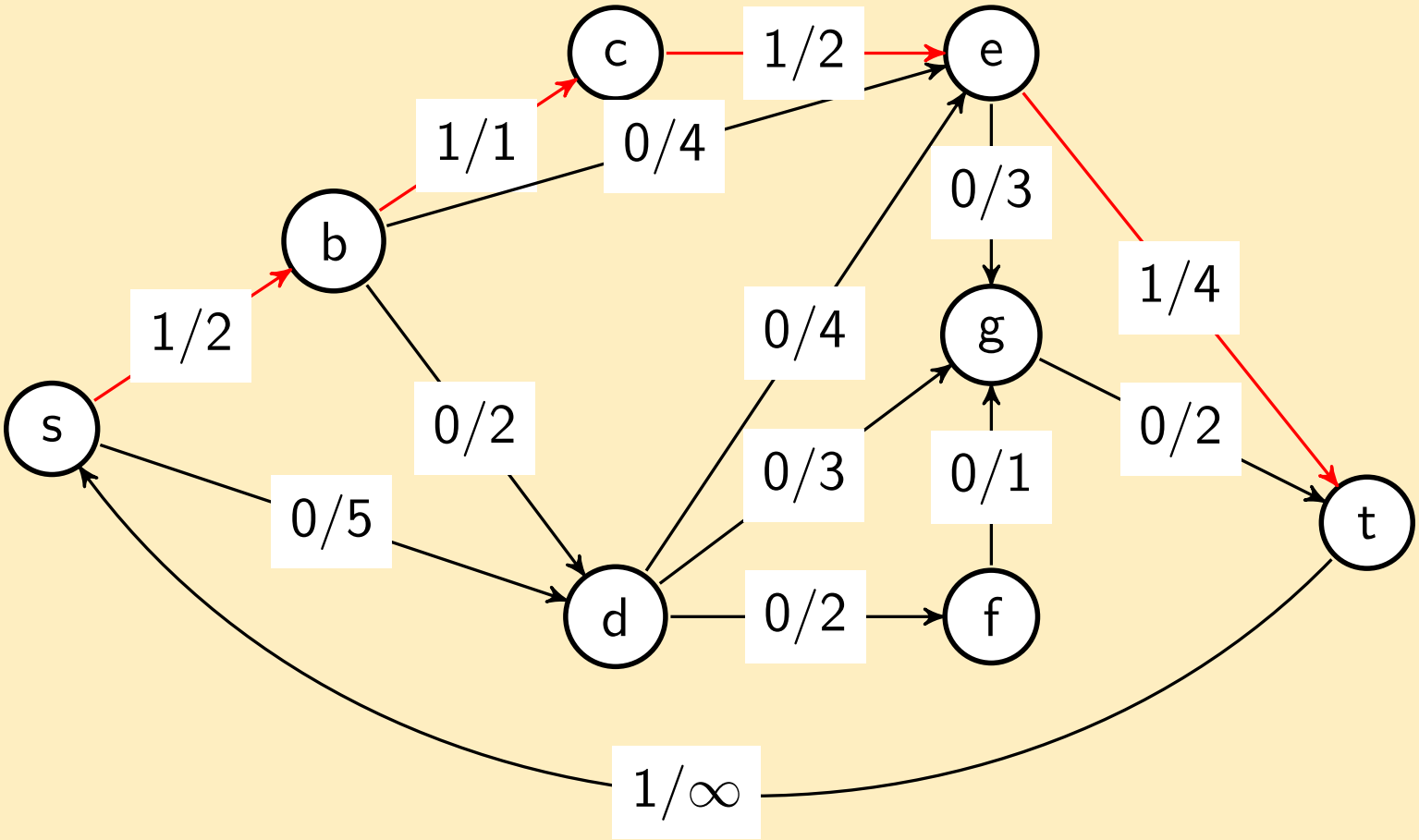


Flots - résolution

Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	b	c	e	t
	$+2$	$+1$	$+1$	$+1$

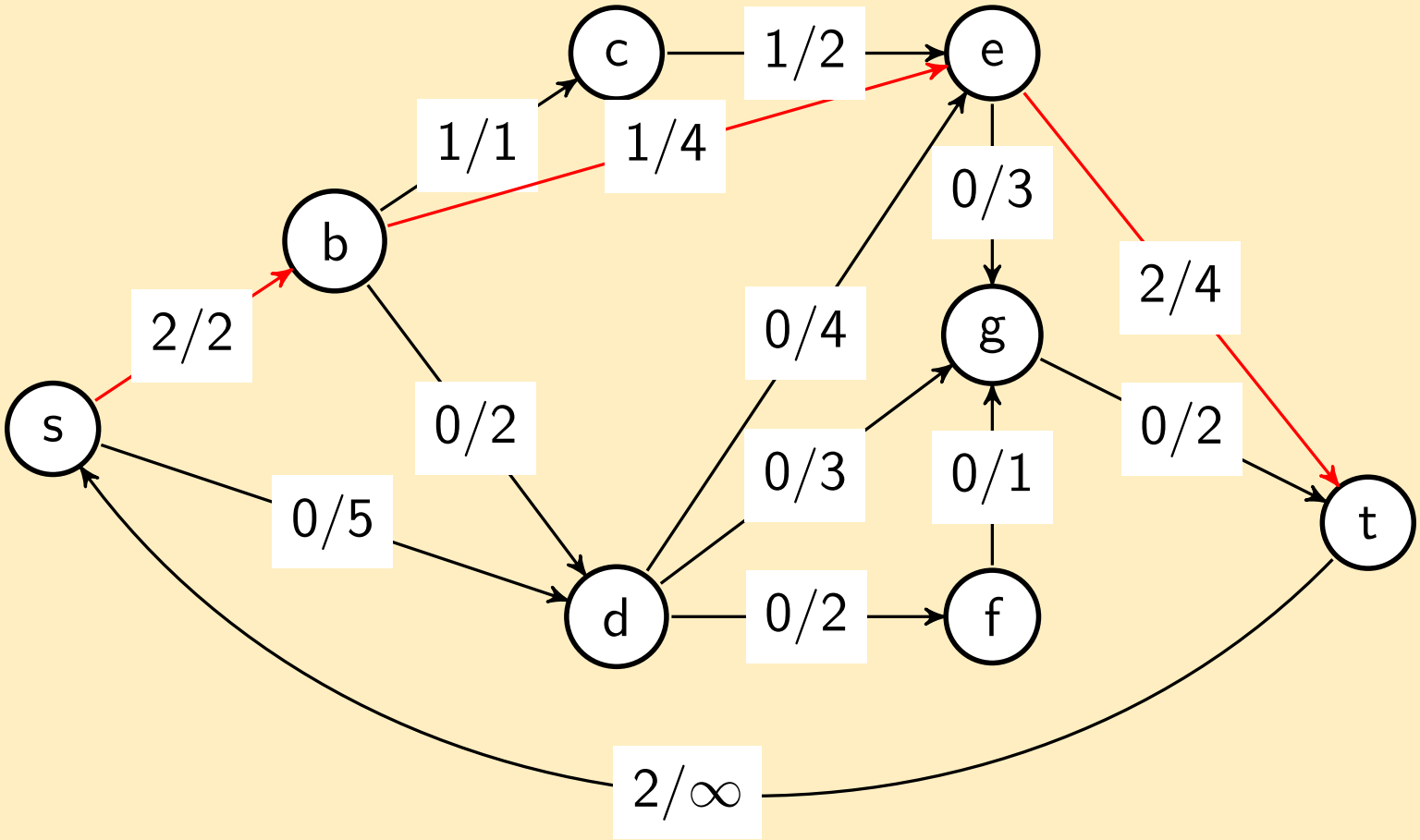


Flots - résolution

Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	b	e	t
	$+1$	$+1$	$+1$

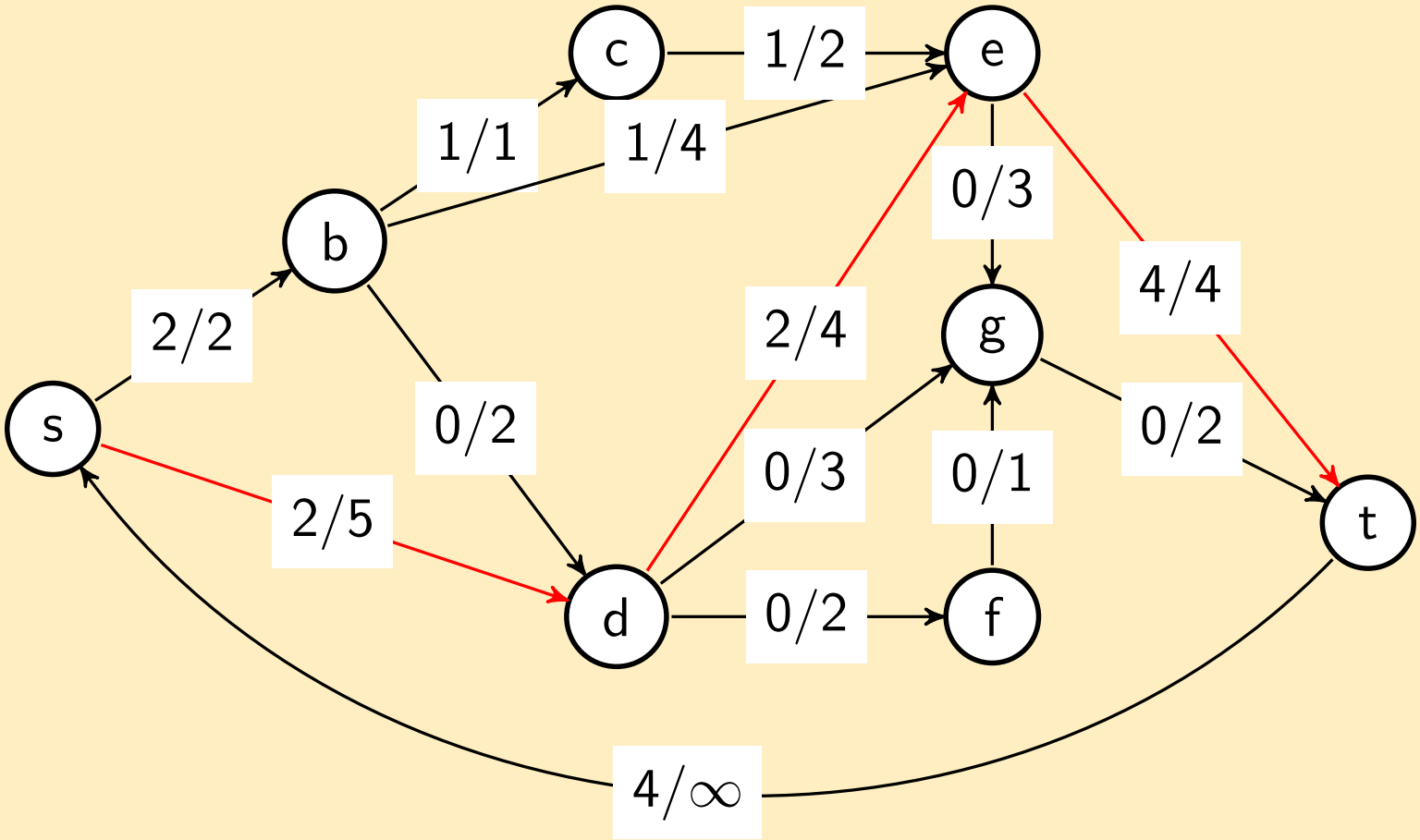


Flots - résolution

Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	d	e	t
	$+5$	$+4$	$+2$

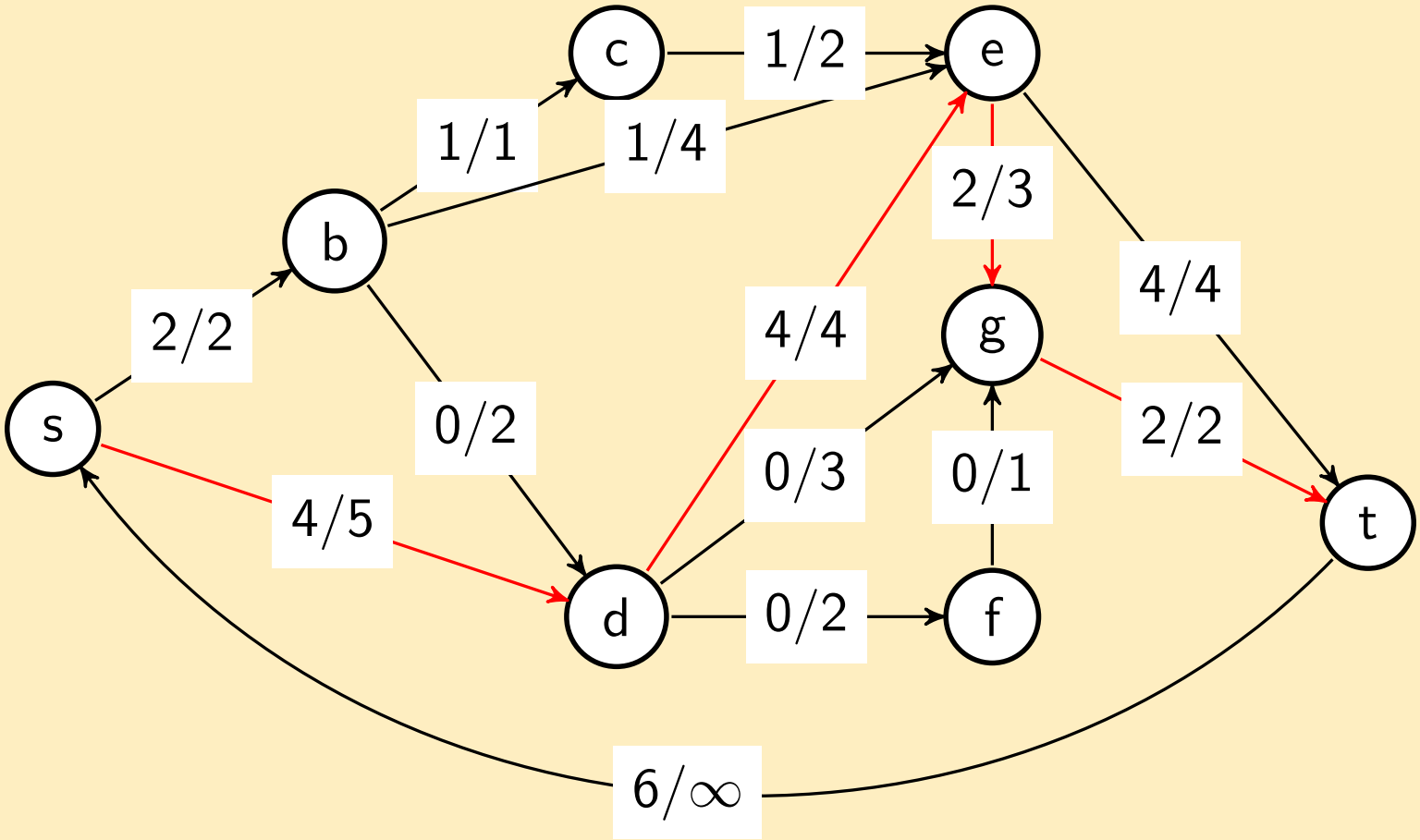


Flots - résolution

Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	d	e	g	t
	$+3$	$+2$	$+2$	$+2$

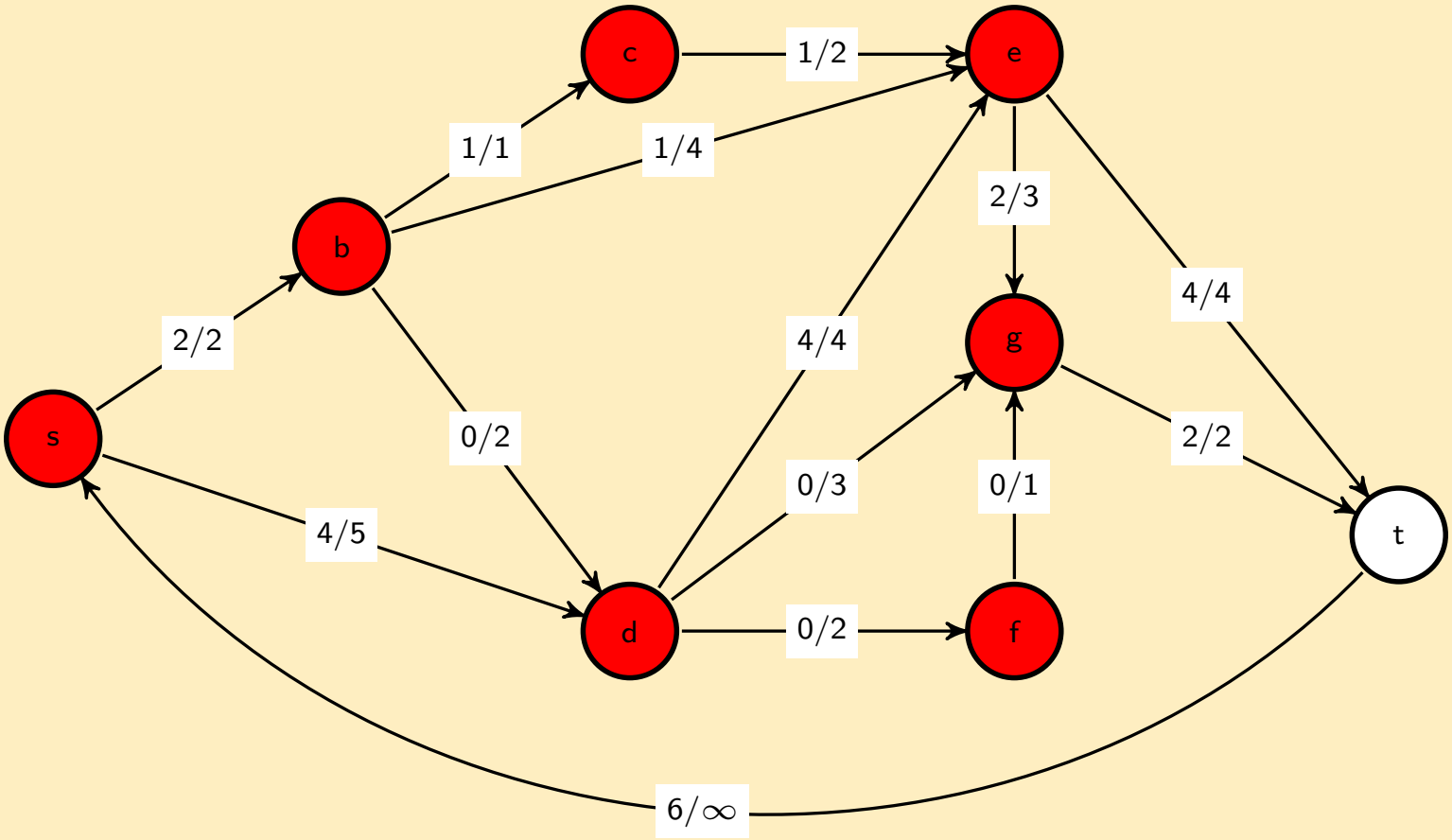


Flots - résolution

Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	d	g	e	c	b
	$+1$	$+1$	$+1$	$+1$	$+1$
s	d	f			
	$+1$	$+1$			



Flots - résolution

Notion de coupe

Une coupe est un sous-ensemble C de $V(G)$ contenant s et ne contenant pas t.
La valeur d'une coupe C, $val(C)$, est définie par :

$$val(C) = \sum_{e=uv, u \in C, v \notin C} c(e)$$

.

Théorème

Soit f un flot réalisable sur un réseau $R = (G, s, t, c)$ et C une coupe de R. Alors $val(f) \leq val(C)$

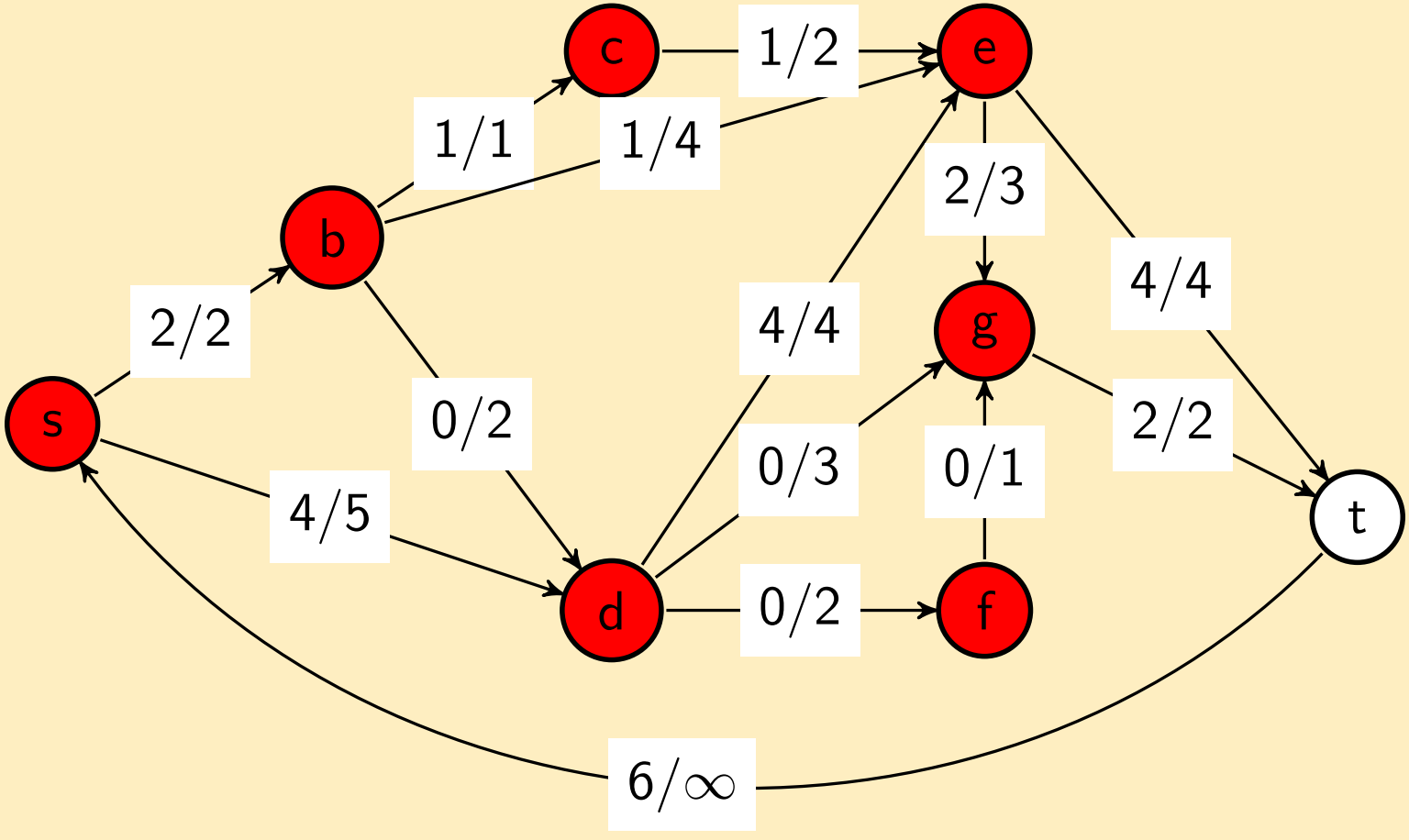
Théorème "Flot max = coupe min"

Soit f un flot maximum sur un réseau $R = (G, s, t, c)$ et C une coupe minimum de R. Alors $val(f) = val(C)$

Flots - résolution

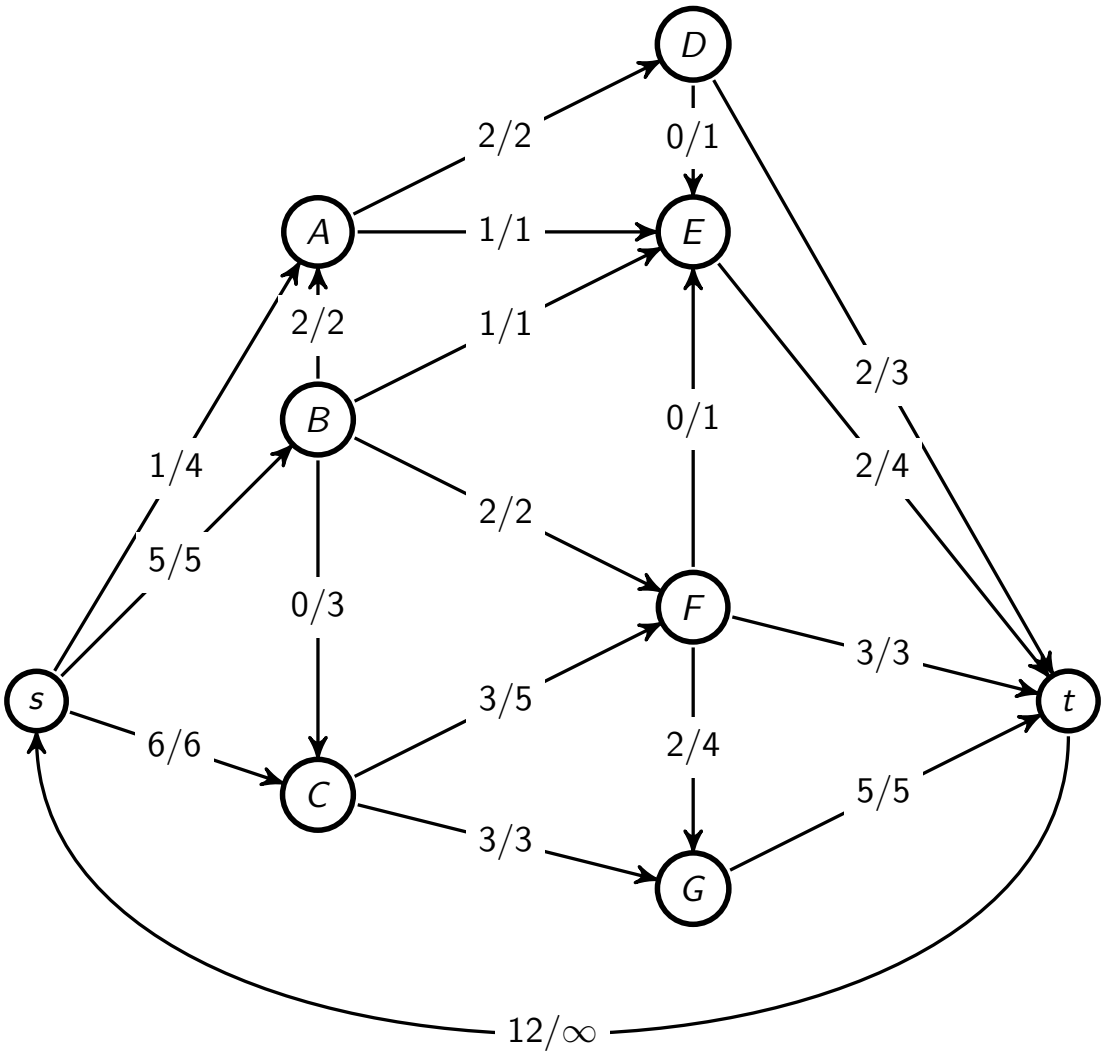
Exemple (Ford-Fulkerson)

$Y = \{s, b, c, d, e, f, g\}$. $val(Y) = c(et) + c(gt) = 6 = val(f)$



Flots - exercice

Résoudre le problème du flot maximum sur le graphe ci-dessous et donner la coupe minimum correspondante.

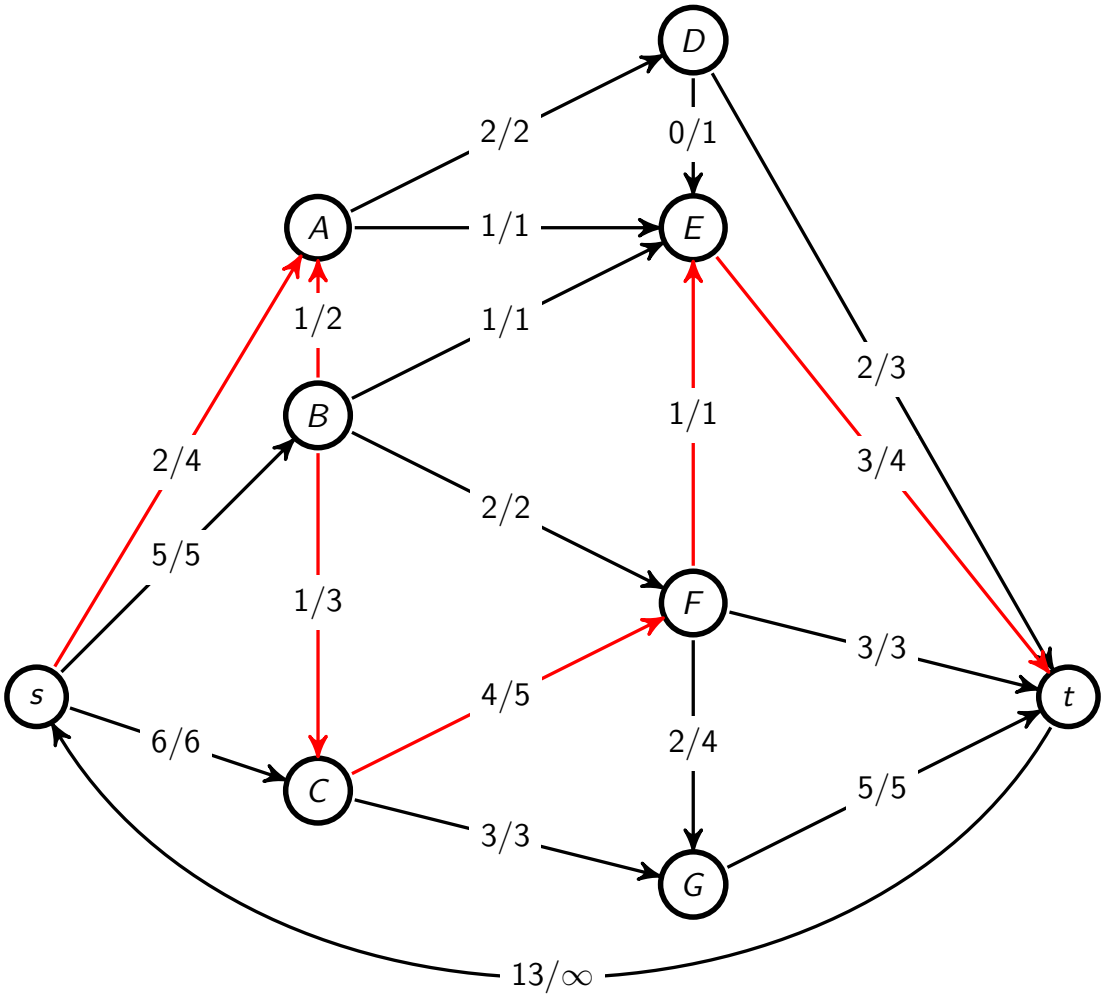


Flots - exercice

Chaîne augmentante (arcs en rouge avec la nouvelle valeur de f) :

v	s	A	B	C	F	E	t
δ	∞	$+3$	$+2$	$+2$	$+2$	$+1$	$+1$

$C+ = \{ts, sA, BC, CF, FE, Et\}$ $C- = \{AB\}$

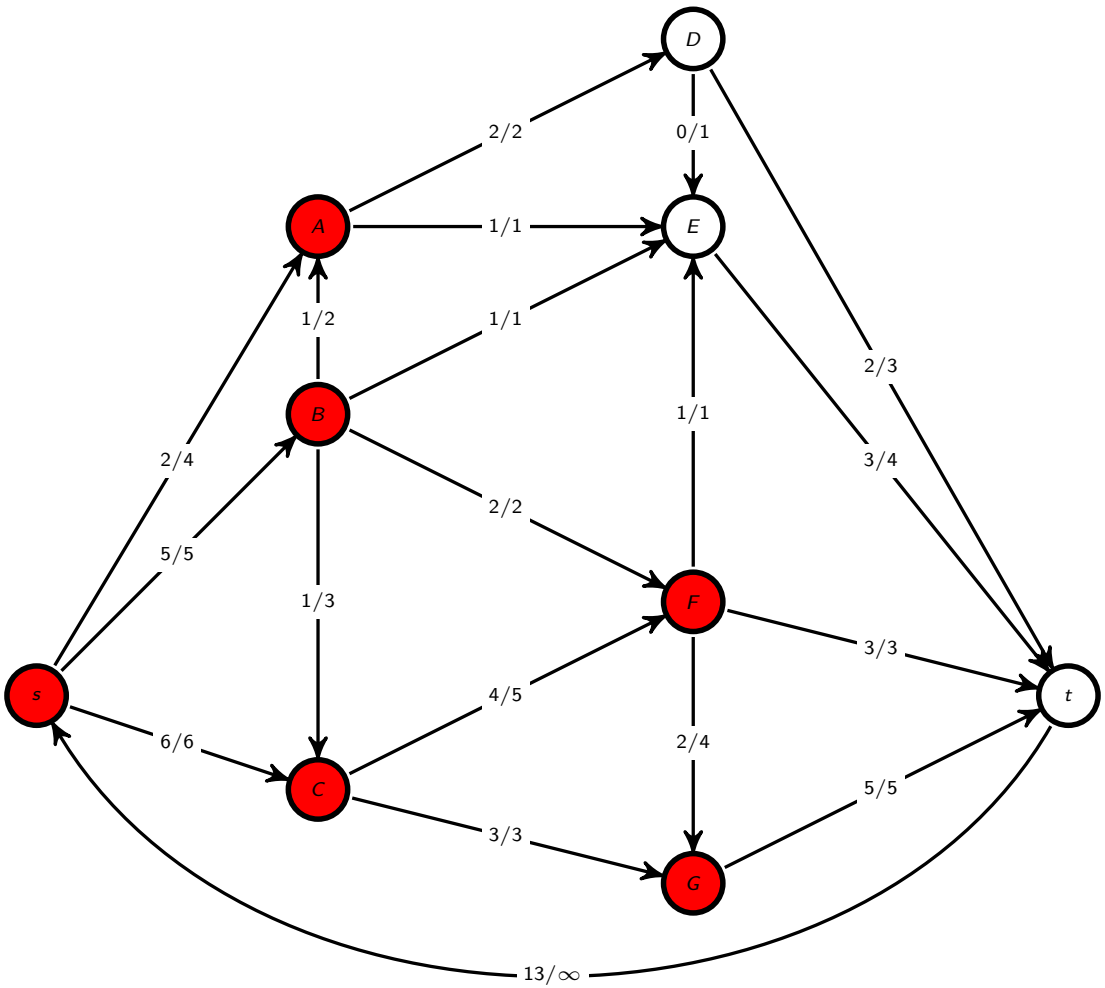


Flots - exercice

Recherche d'un chaîne augmentante (sommets de Y en rouge) :

v	s	A	B	C	F	G	
δ	∞	$+2$	$+1$	$+1$	$+1$	$+1$	

On n'a pas réussi à atteindre t, donc $Y = \{s, A, B, C, F, G\}$ doit être une coupe minimum.
$val(Y) = c(AD) + c(AE) + c(BE) + c(FE) + c(Ft) + c(Gt) = 2 + 1 + 1 + 1 + 3 + 5 = 13 = val(f)$



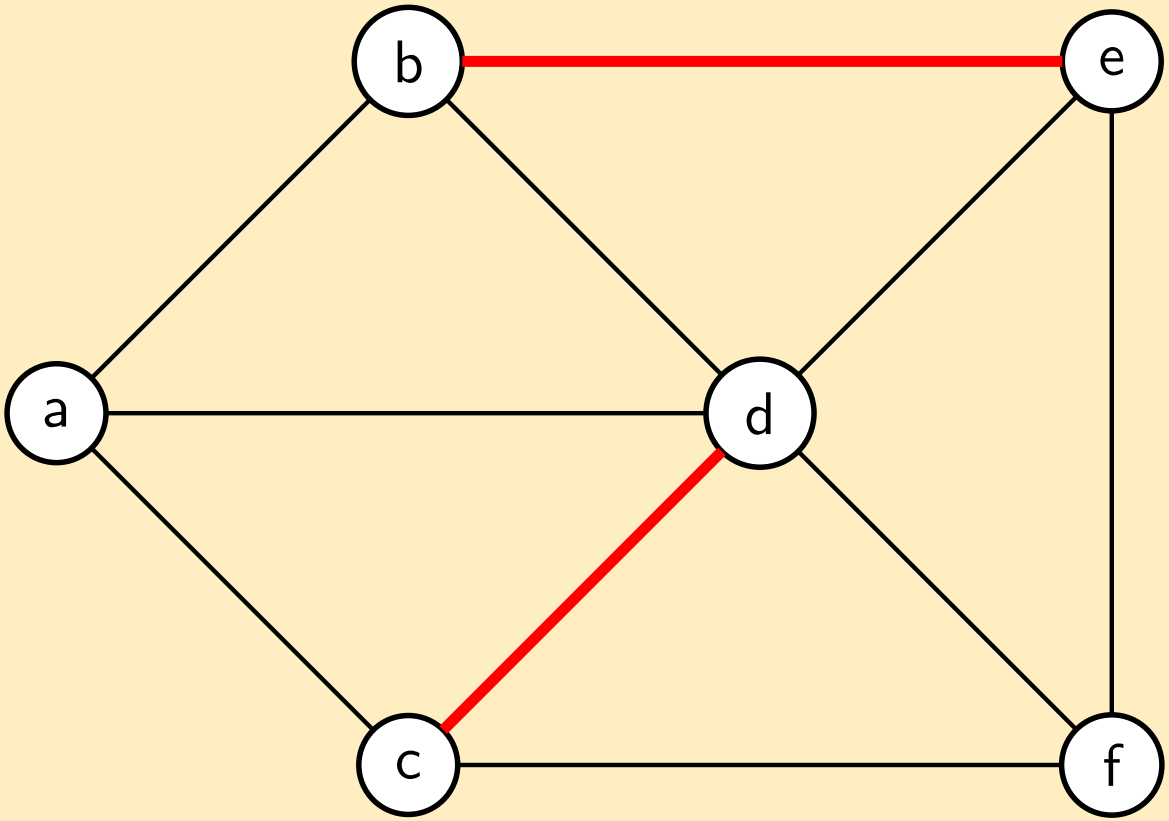
Algorithme de Ford et Fulkerson

Complexité

La complexité de l'algorithme de Ford et Fulkerson dépend de la valeur du flot maximum à trouver, notée f_{max}.
La complexité de Ford et Fulkerson est en $0(m \times f_{max})$

Couplage

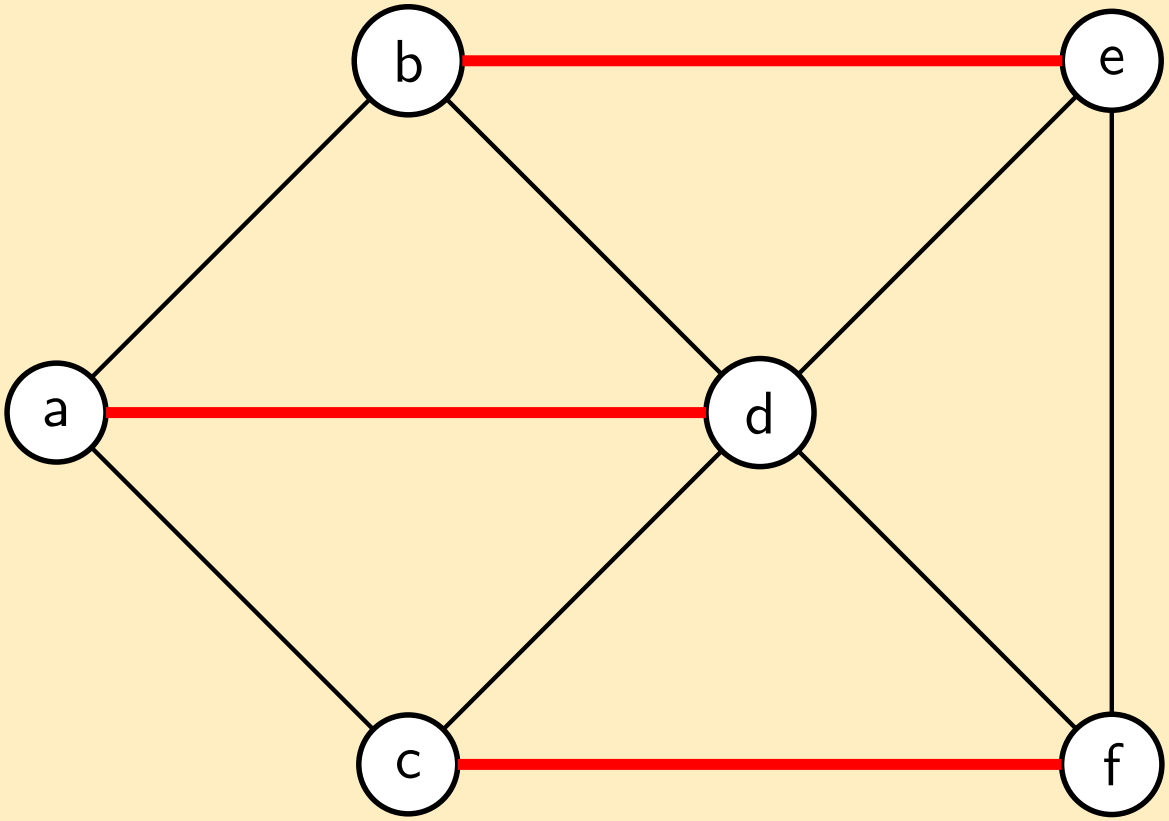
Un couplage dans un graphe G non orienté est un ensemble d'arêtes n'ayant aucun sommet en commun.



Remarque : ce couplage est maximal. Question : est-il maximum ?

Flots - applications

Exemple de couplage maximum



Remarque : un couplage qui couvre tous les sommets du graphe est appelé un couplage parfait.

Couplage maximum dans un graphe biparti

Graphe biparti

Un graphe non orienté $G = (V, E)$ est biparti si son ensemble de sommets peut être partitionné en deux sous-ensembles disjoints A et B tels que les sous-graphes induits par A et B soient des stables (ne contiennent aucune arête). On notera alors le graphe G par $G = (A \cup B, E)$.
Rappel (vu en TD) : un graphe G est biparti si et seulement si il ne contient aucun cycle impair.

Couplage maximum dans un graphe biparti

Exemple

Vous travaillez dans une agence matrimoniale et après une série d'entretiens, vous en déduisez les couples de personnes que vous jugez "compatibles". Cet ensemble est décrit par le tableau suivant :

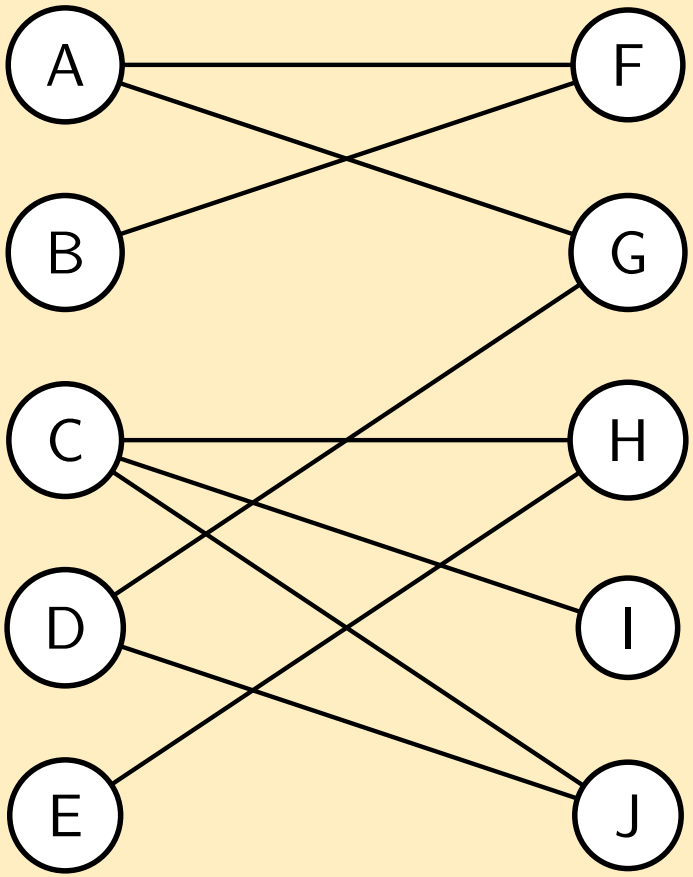
	Alice	Béatrice	Charlotte	Dominique	Eliane
François	•	•	-	-	-
Gérard	•	-	-	•	-
Henri	-	-	•	-	•
Ivan	-	-	•	-	-
Julien	-	-	•	•	-

Vous souhaitez organiser le maximum de rencontres possibles simultanées entre personnes compatibles.

Couplage maximum dans un graphe biparti

Exemple

Voici le graphe issu du tableau. Les prénoms ont été remplacés par l'initiale correspondante.



Votre problème est maintenant de trouver un couplage maximum dans le graphe ci-dessus.

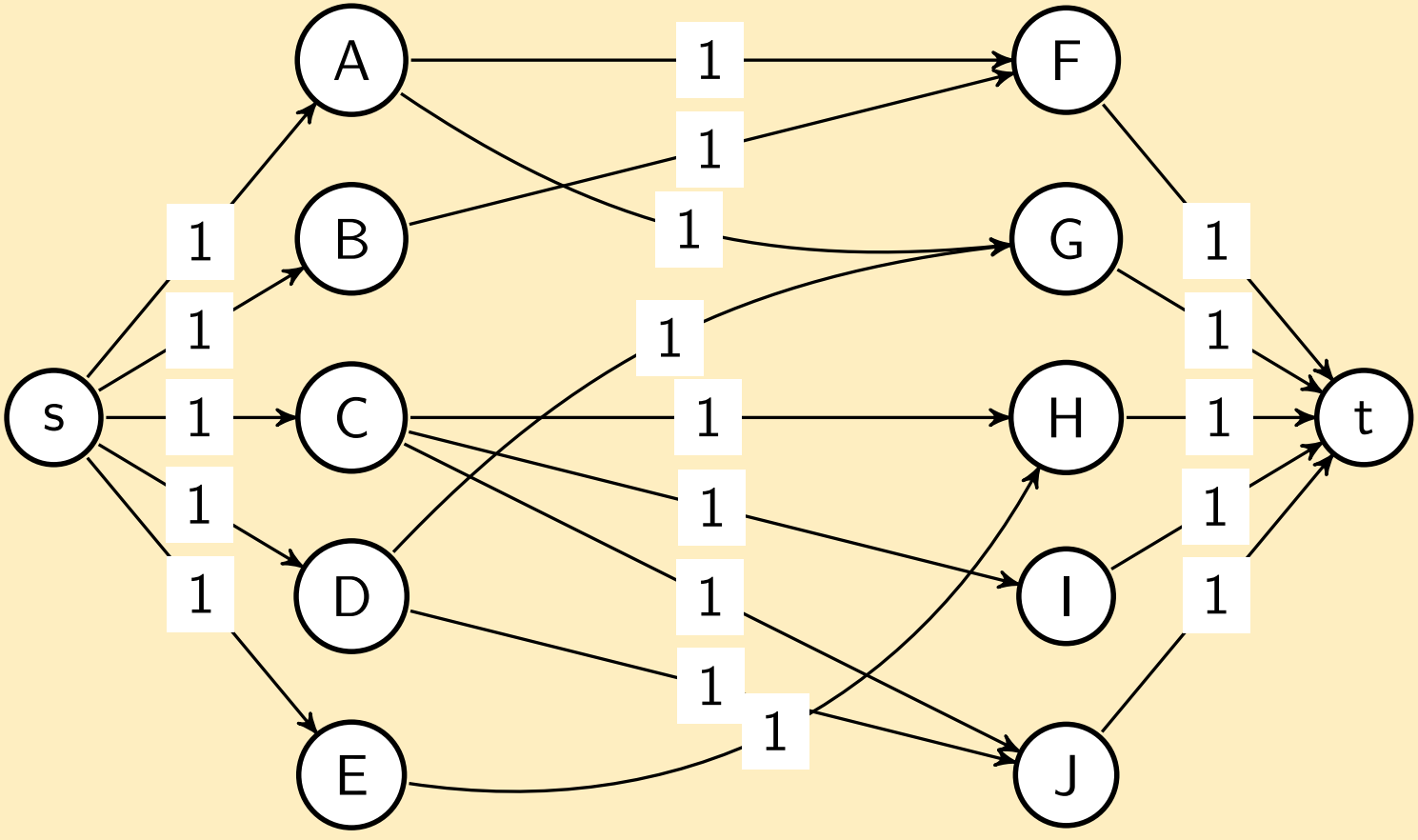
Couplage maximum dans un graphe biparti

Exemple

Pour transformer le problème de recherche d'un couplage maximum dans un graphe biparti $G = (A \cup B, E)$ en un problème de flot, on rajoute deux sommets à $V(G)$: s et t. On rajoute des arcs de s vers chaque sommets de A, des arcs de chaque sommet de B vers t et chaque arête de $E(G)$ est orientée de A vers B. Tous les arcs sont munis d'une capacité de 1.

Flots - applications

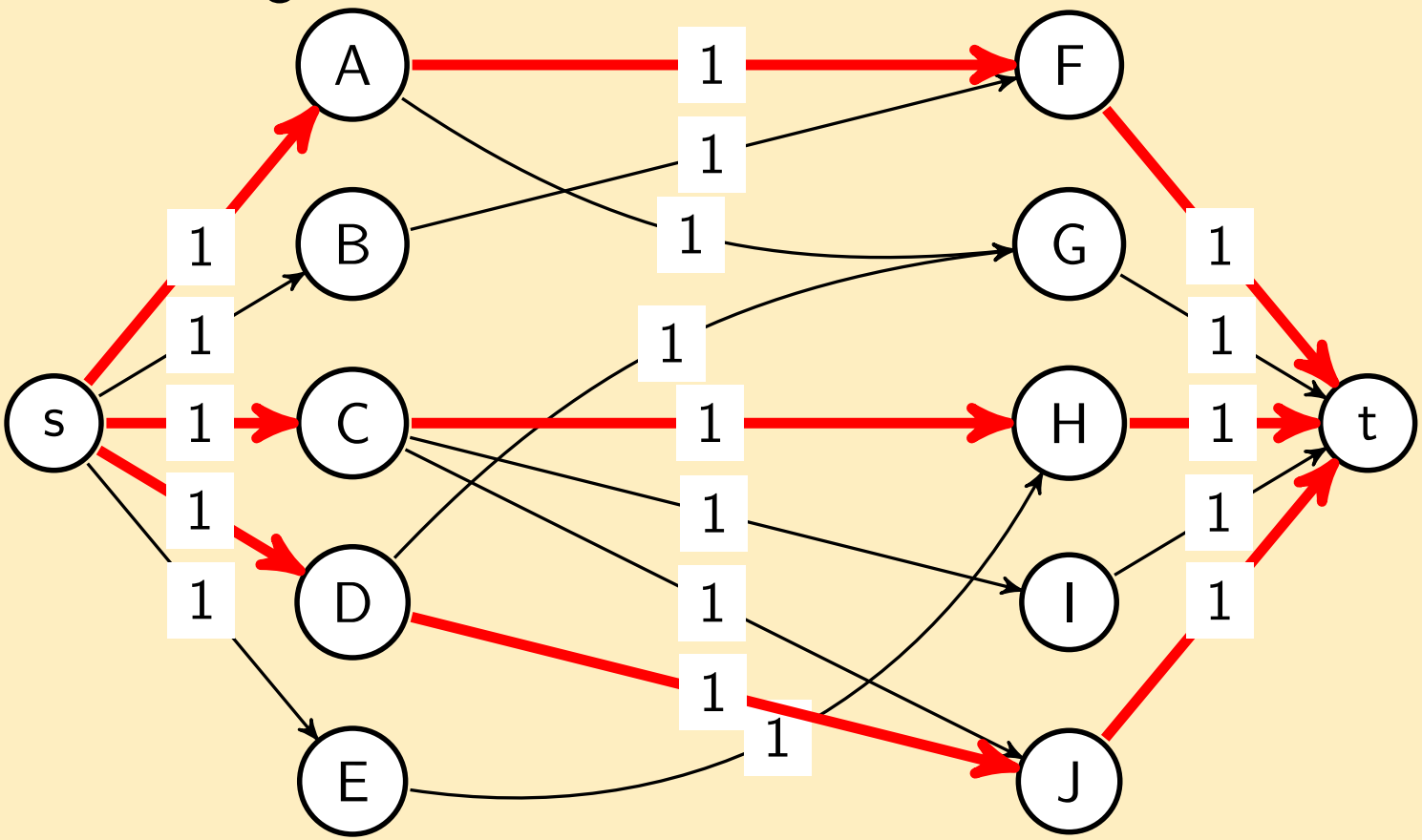
Exemple d'application



Couplage maximum dans un graphe biparti

Exemple

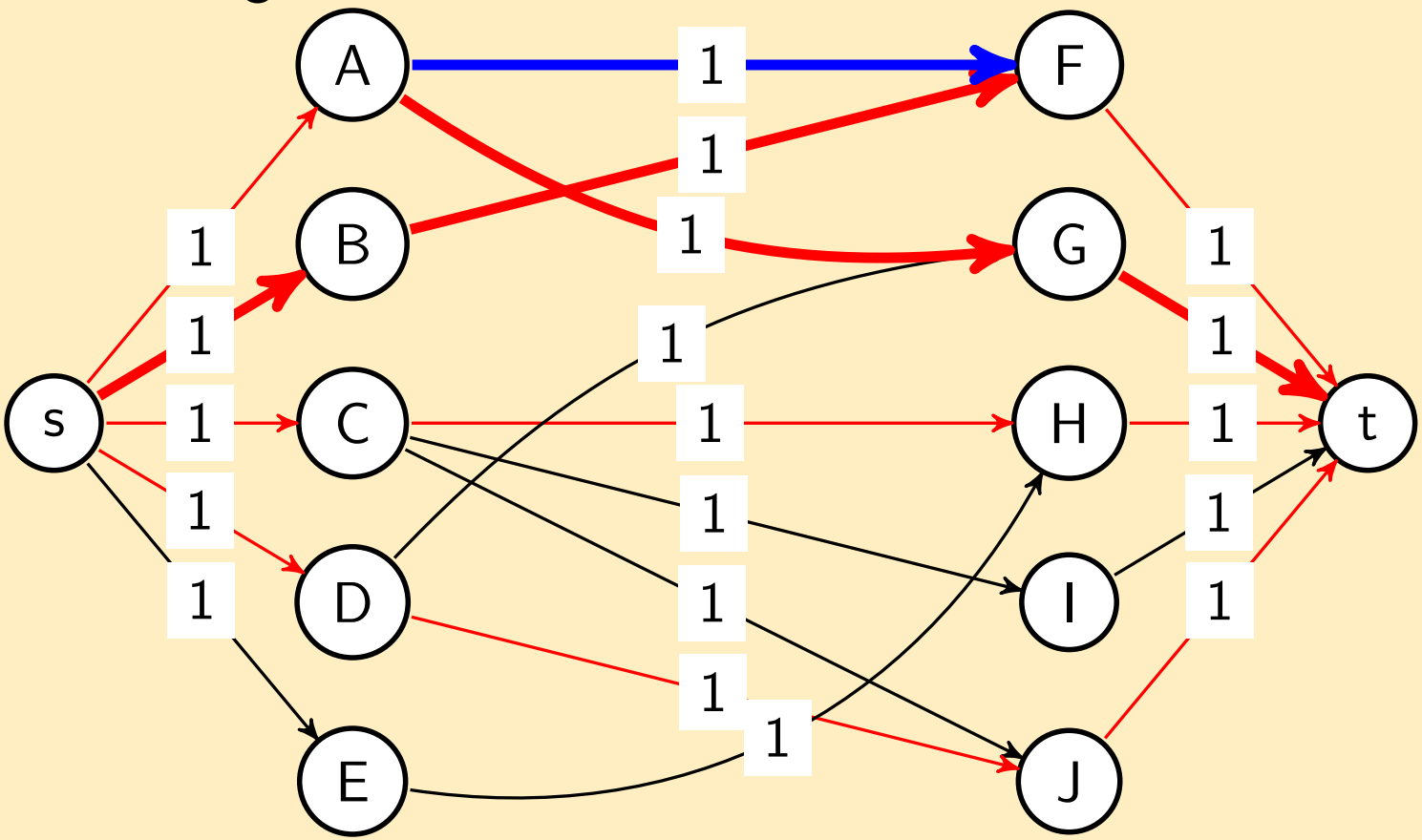
Chaînes augmentantes : s, A, F, t + s, C, H, t + s, D, J, t



Couplage maximum dans un graphe biparti

Exemple

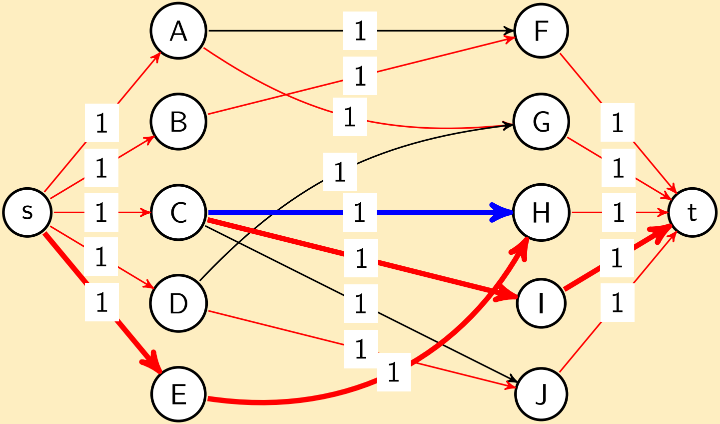
Chaîne augmentante : s, B, F, A, G, t



Couplage maximum dans un graphe biparti

Exemple

Chaîne augmentante : s, E, H, C, I, t



Couplage maximum dans un graphe biparti

Remarque

La recherche d'un chaîne augmentante consiste à chercher entre les sommets du stable A et les sommets du stable B une chaîne alternée d'arêtes n'appartenant pas (quand on va de A vers B) ou appartenant (quand on va de B vers A) au couplage. Si on termine sur B, la chaîne aura une arête hors couplage de plus que d'arêtes appartenant au couplage et donc en inversant les arêtes hors couplage et dans le couplage, on augmente de 1 la taille du couplage.