Méthode Ford & Fulkerson

tant qu'il existe une chaîne augmentante

faire

- **chercher une chaîne augmentante**
- **augmenter le flot sur la chaîne**

fait

A préciser :

- **comment chercher**
- **de combien augmenter**

Le théorème Flot-max Coupe-min

Théorème :

Soit *f* un flot dans un réseau G(V,E) de source *s* et puits *t*. Les conditions suivantes sont équivalentes :

1. ***f* est un flot maximum**
2. **le réseau résiduel G_f ne contient pas de chaîne augmentante**
3. **$|f| = c(S,S')$ pour une coupe (S,S') de G**

Preuve :

$3 \Rightarrow 1$

Comme $|f| \leq c(S,S')$ pour toute coupe (S,S') de G, si on a égalité, c'est un flot maximum.

$1 \Rightarrow 2$

ou plus exactement $\neg 2 \Rightarrow \neg 1$

Supposons que G_f contient une chaîne augmentante.

Ainsi, on peut augmenter f et donc f n'est pas un flot maximum.

$2 \Rightarrow 3$

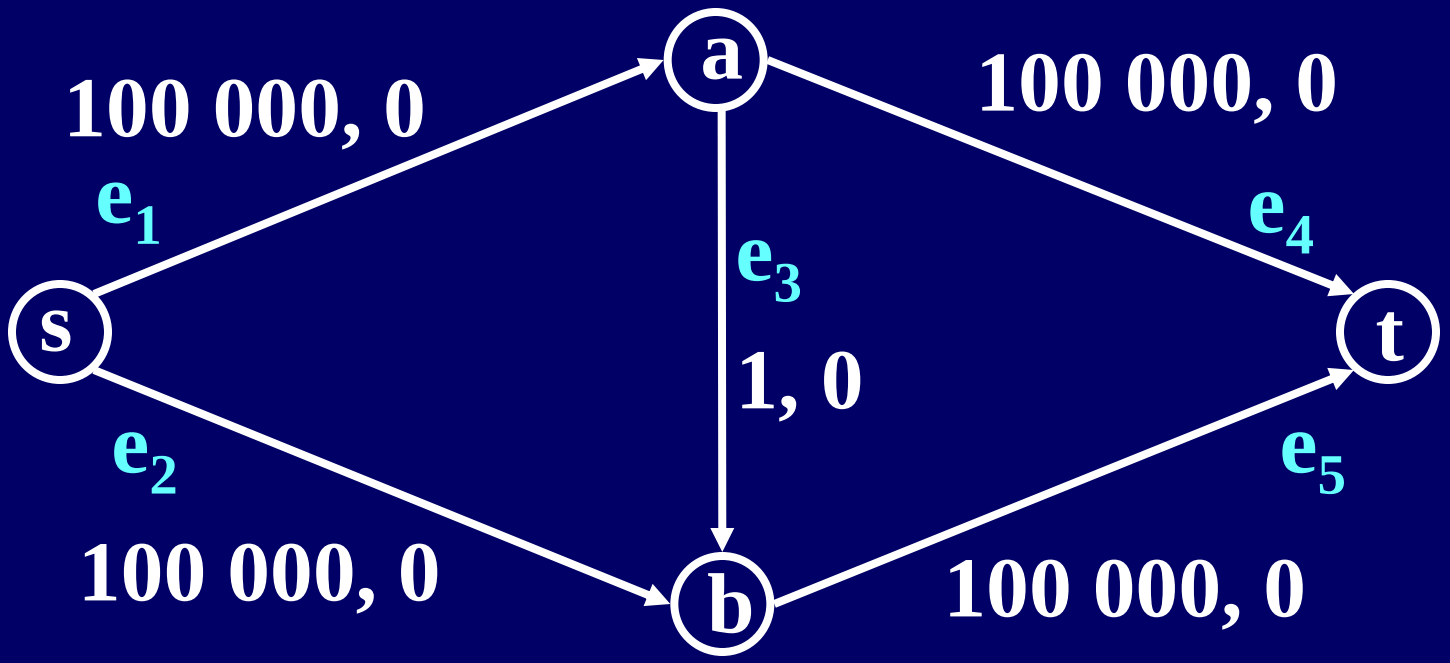
Si G_f ne contient pas de chaîne augmentante, c'est que G_f ne contient pas de chemin de s vers t. Soit $S=\{x \in V \mid$ il existe un chemin dans G_f de s vers $x\}$ et soit $S' = V-S$.

Ainsi (S, S') est une coupe qui sépare s de t ($s \in S$ et $t \notin S$) .

Soit uv un arc de G, avec $u \in S$ et $v \in S'$. Comme uv n'est pas dans G_f, nous devons avoir $f(uv)=c(uv)$.

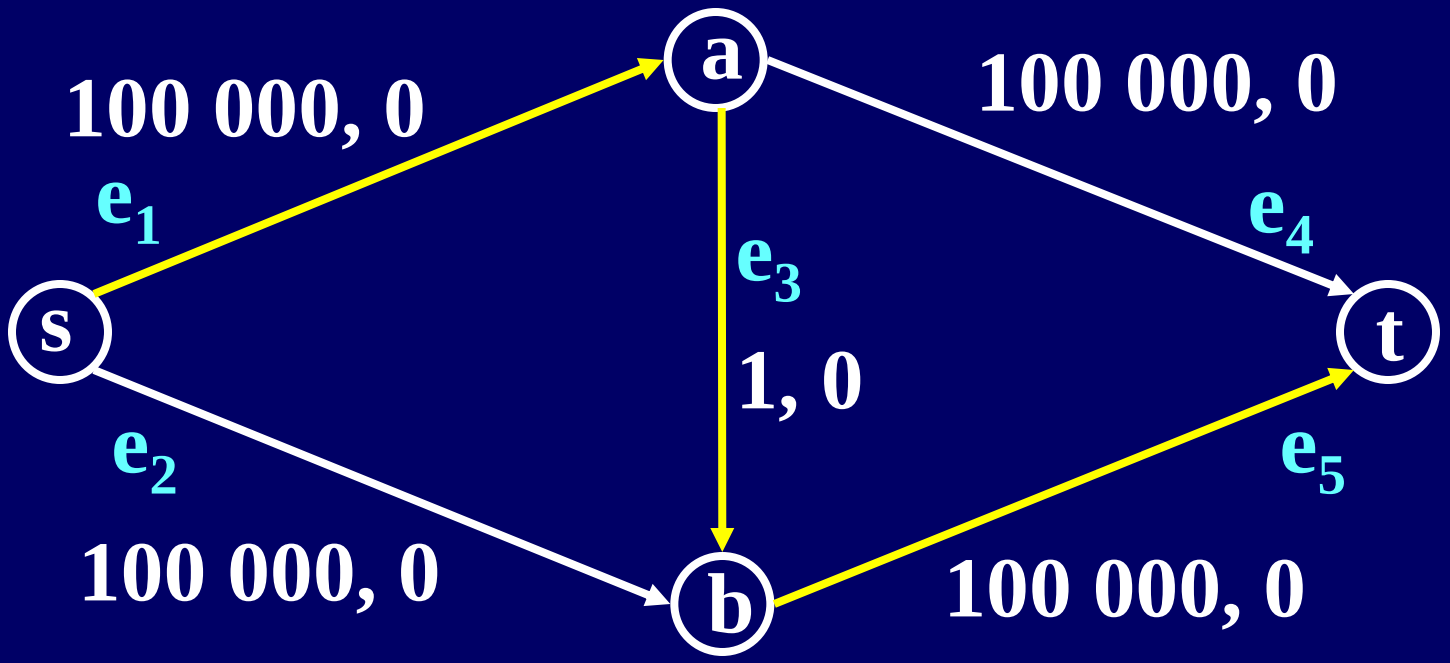
Ainsi $|f| = c(S,S')$. **CQFD**

Un petit problème (due à Karp)



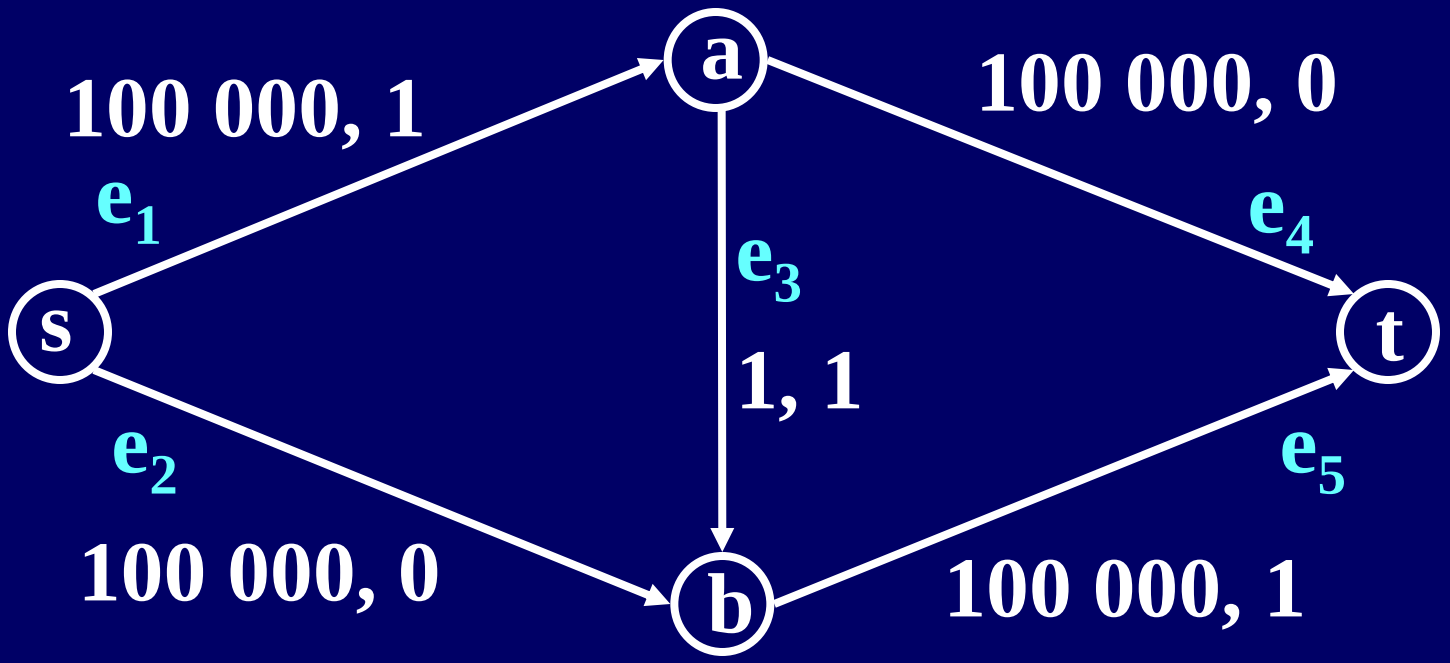
c, f

Un petit problème (due à Karp)



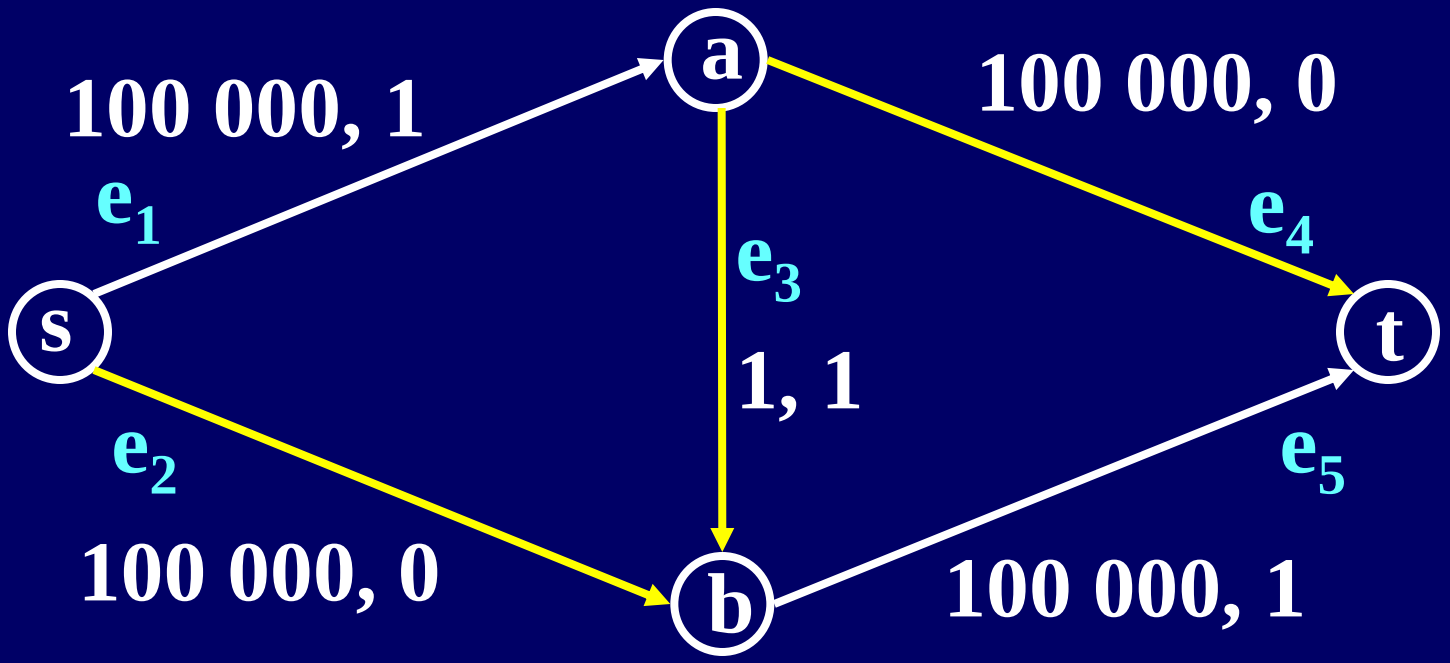
c, f

Un petit problème (due à Karp)



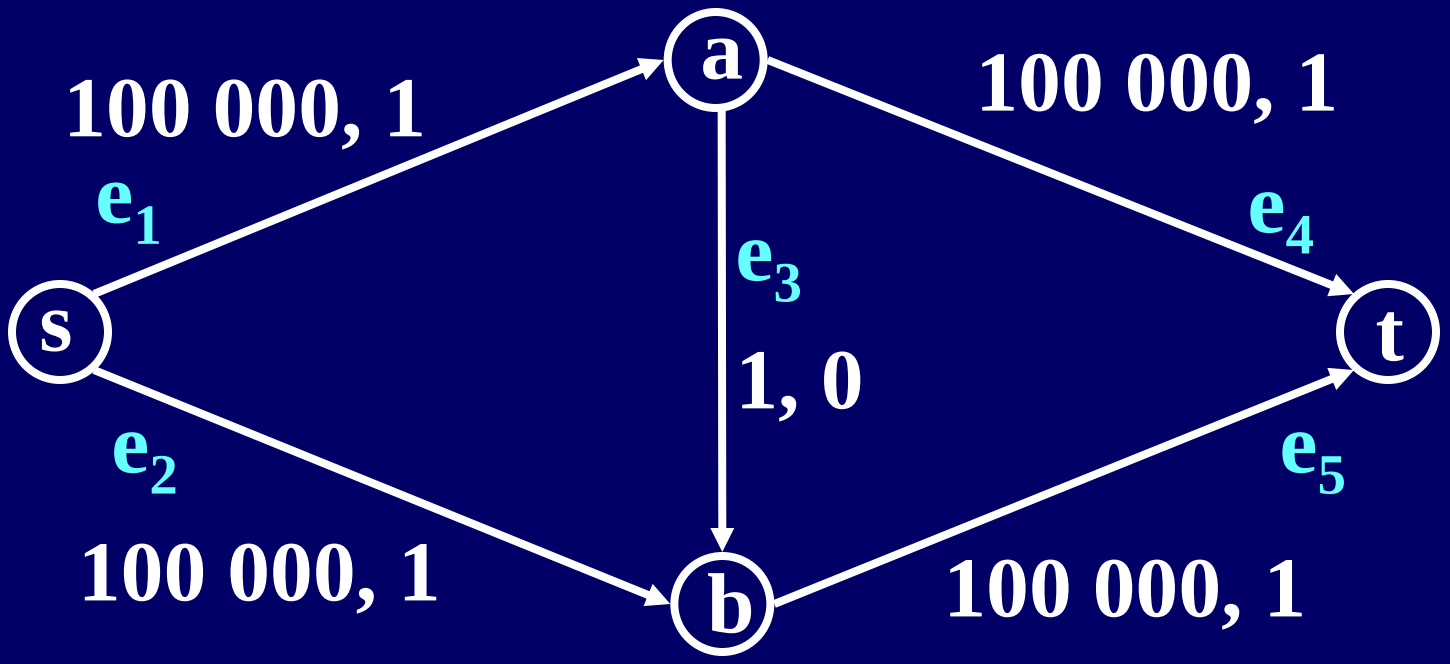
c, f

Un petit problème (due à Karp)



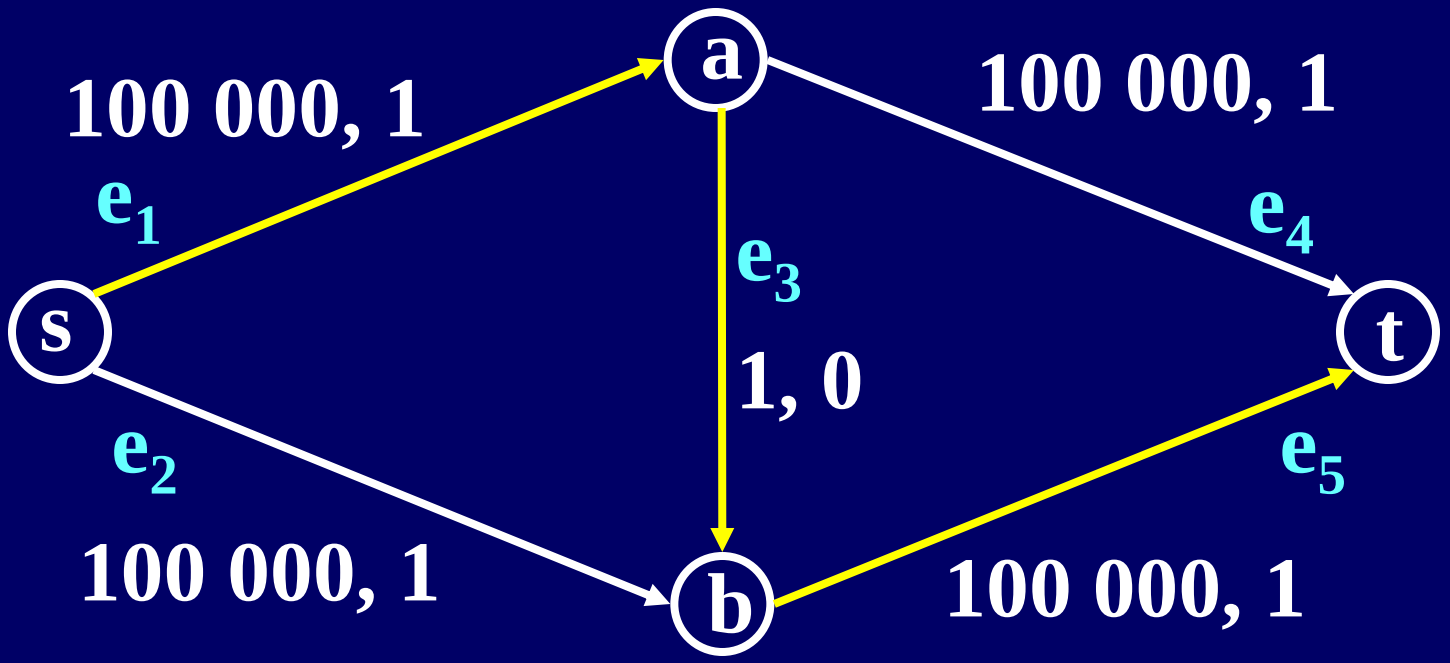
c, f

Un petit problème (due à Karp)



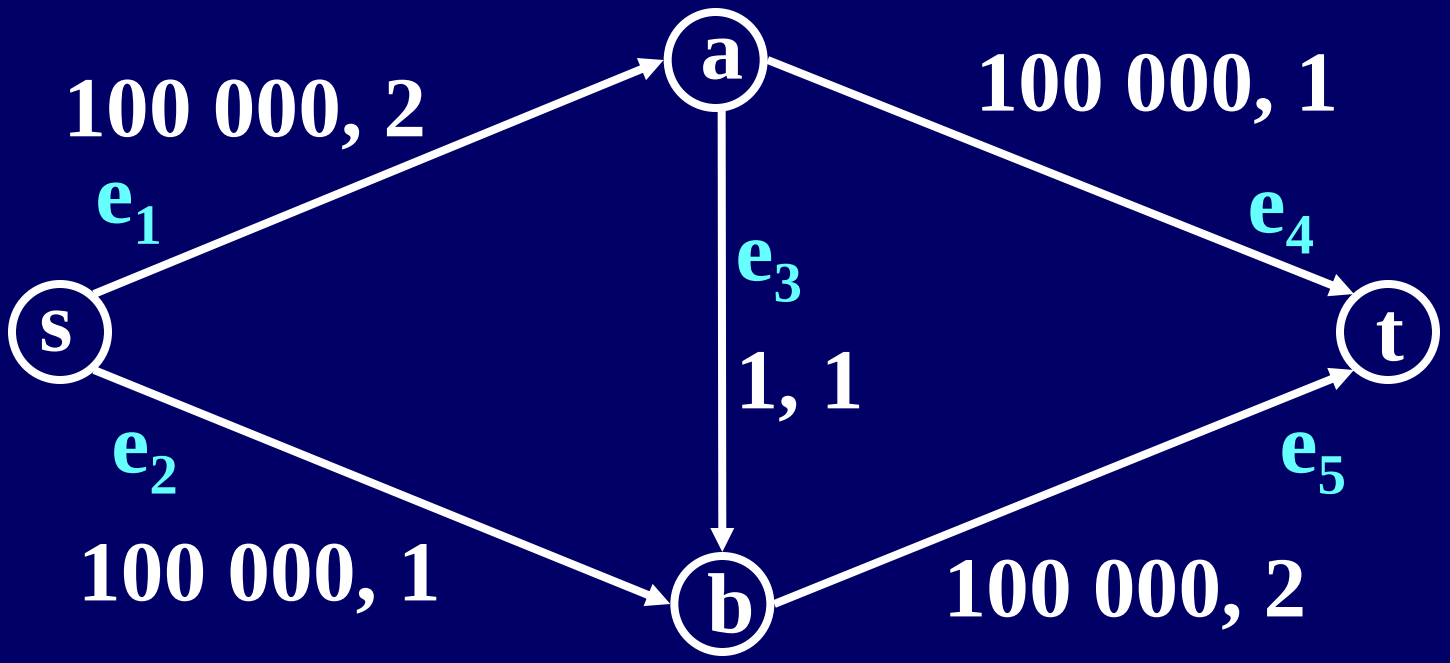
c, f

Un petit problème (due à Karp)



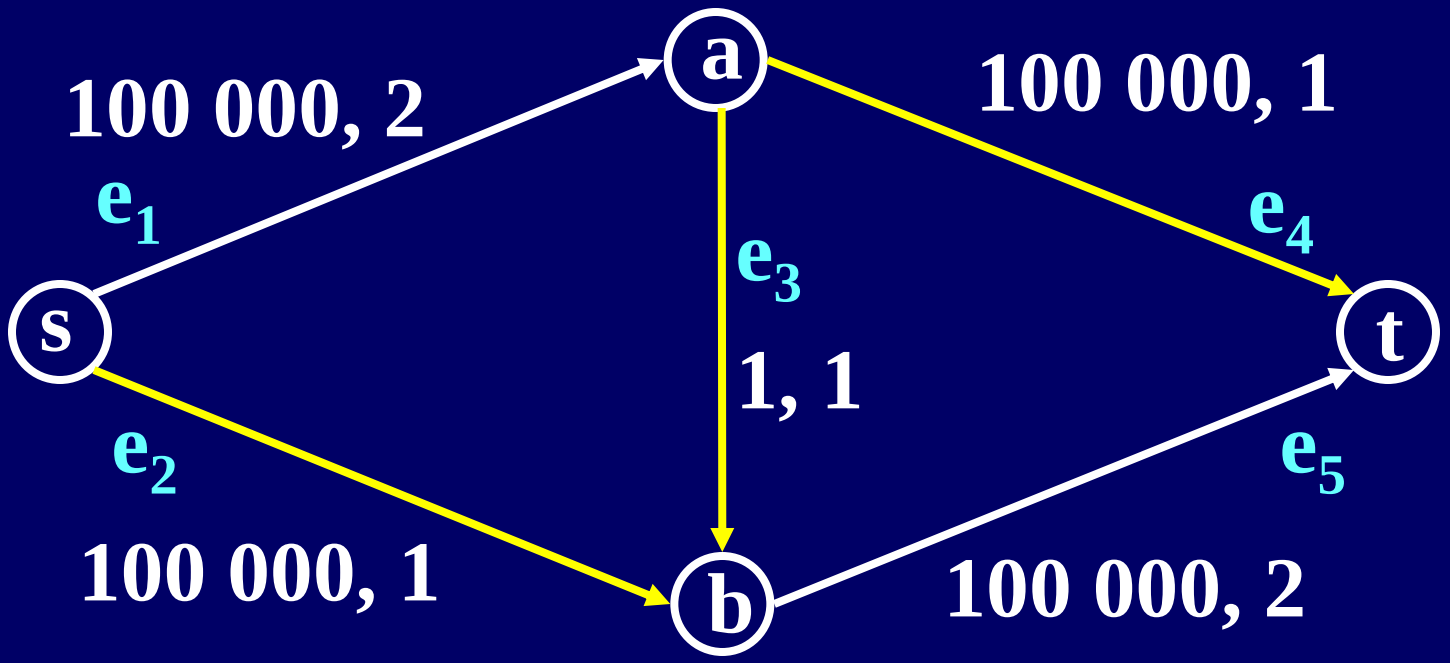
c, f

Un petit problème (due à Karp)



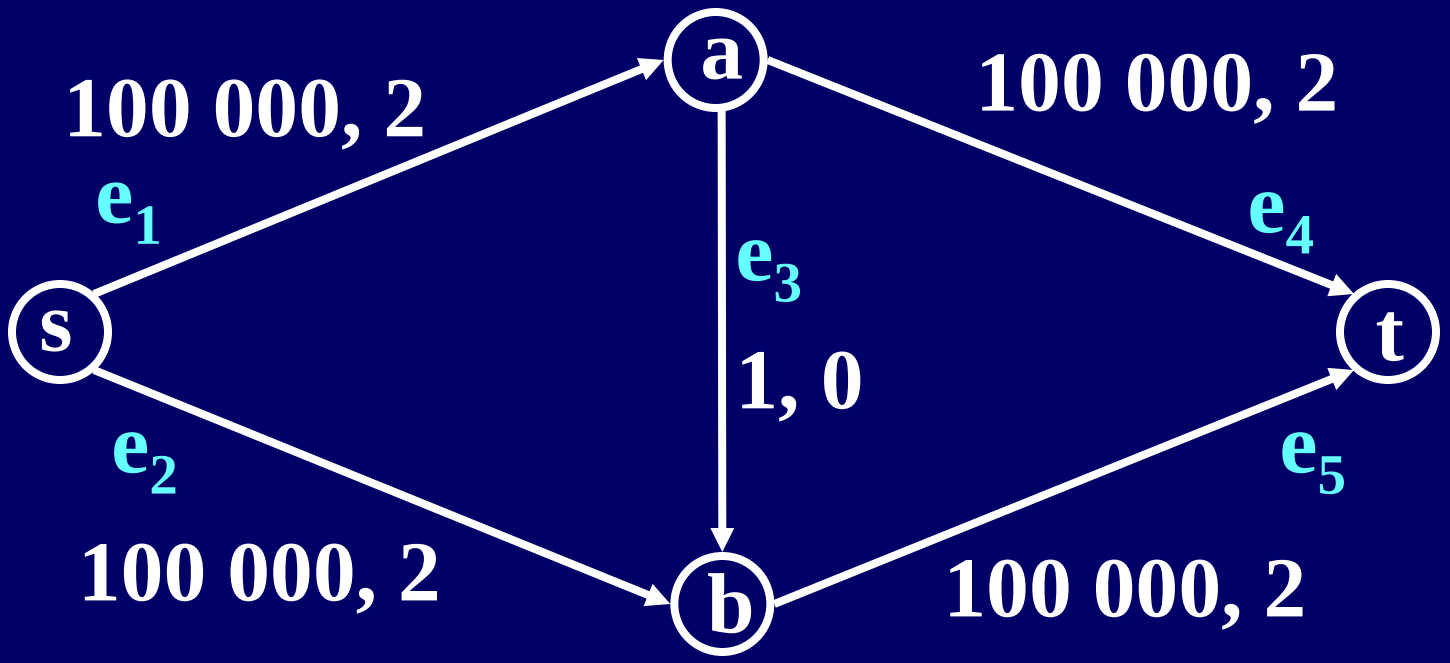
c, f

Un petit problème (due à Karp)



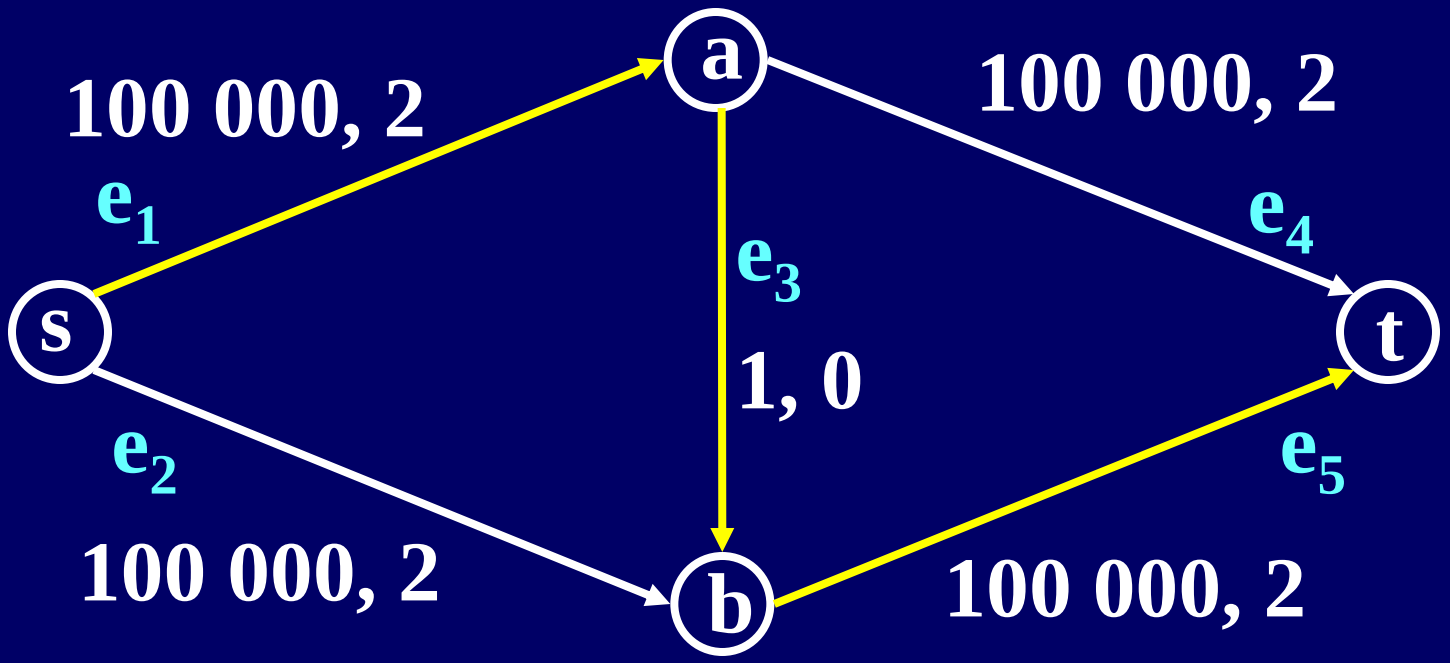
c, f

Un petit problème (due à Karp)



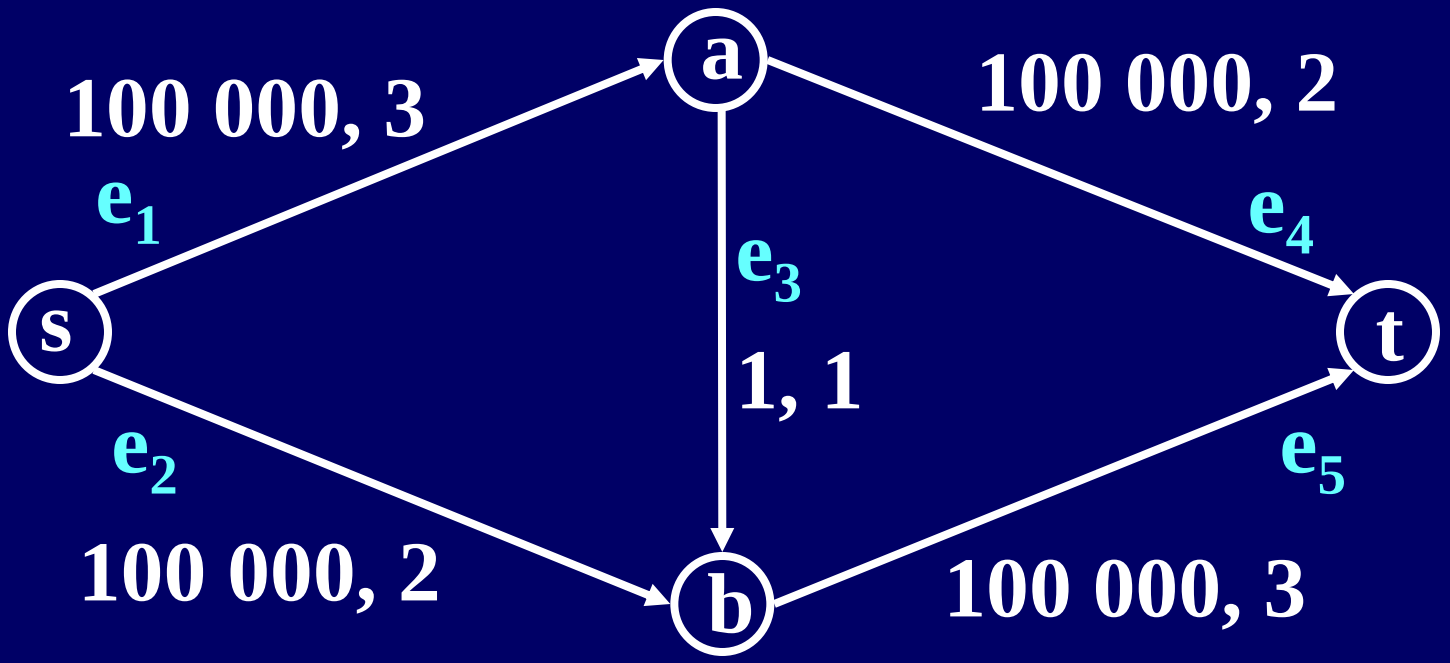
c, f

Un petit problème (due à Karp)



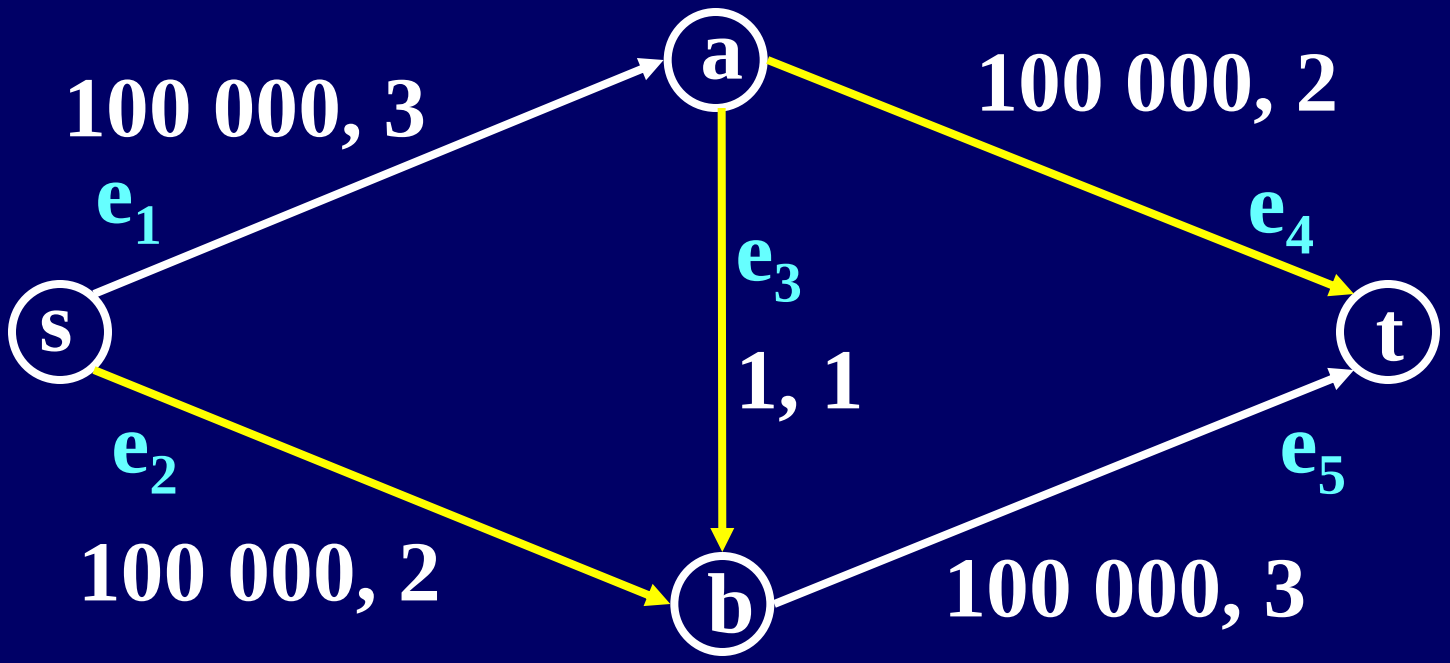
c, f

Un petit problème (due à Karp)



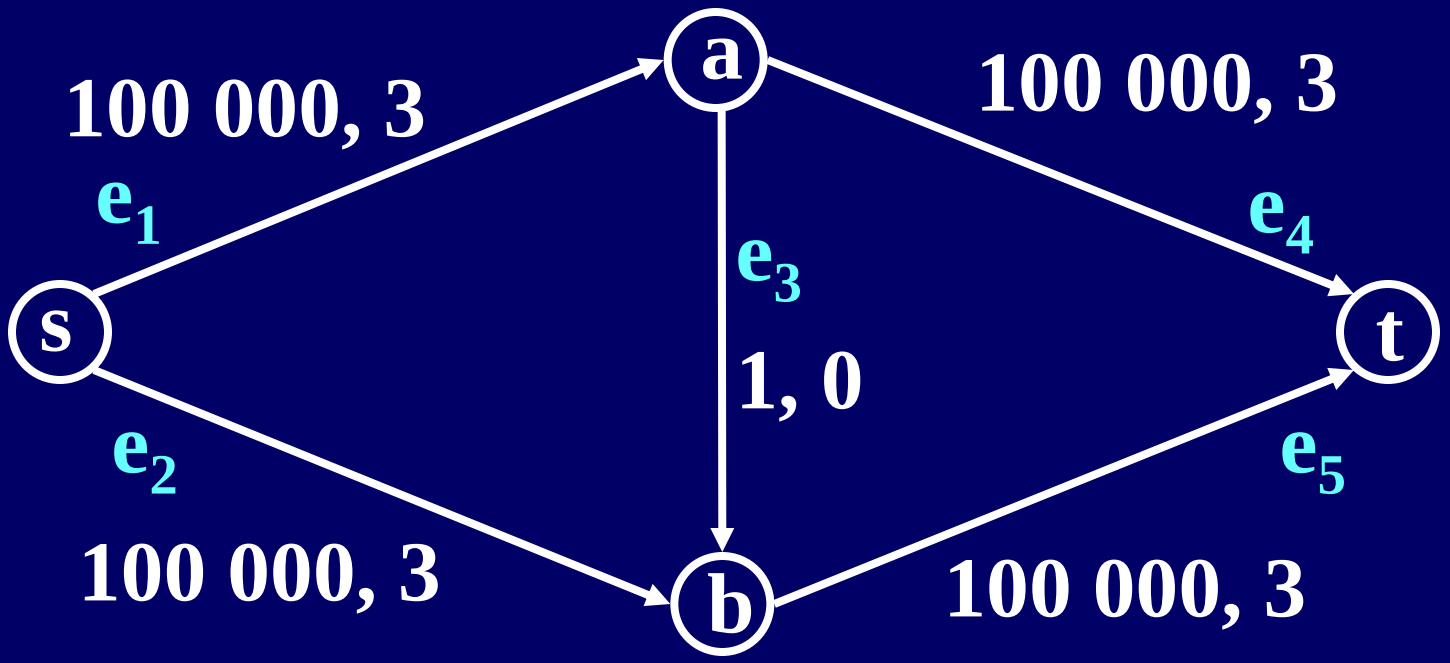
c, f

Un petit problème (due à Karp)



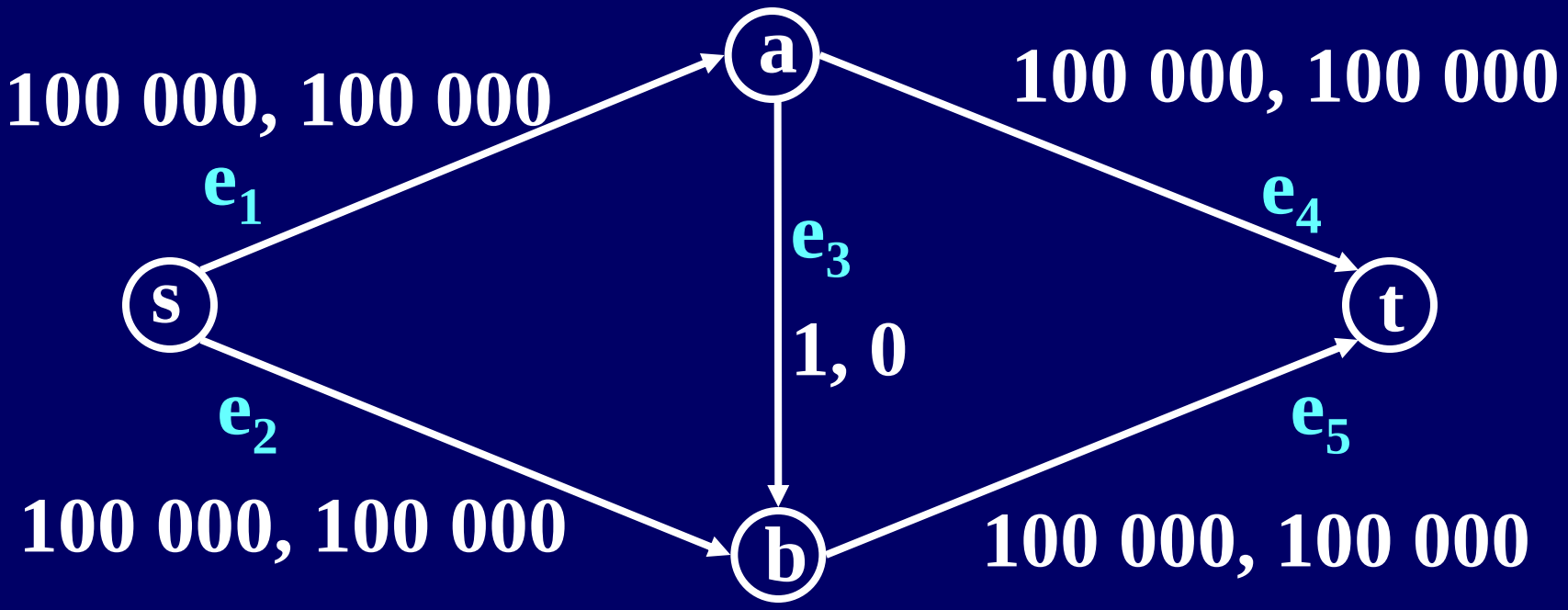
c, f

Un petit problème (due à Karp)



c, f

Le résultat final
(après 200 000 augmentations)



c, f

Conclusion

Tel quel l'algorithme n'est pas polynomial !

Que pouvons nous dire quand même ?

$$O(f_{max} |E|)$$

Pourquoi ?

Car si les capacités sont entières, alors toutes les valeurs sont toujours entières !

Version Edmonds-Karp

Il s'agit de choisir une plus courte chaîne augmentante.

Intérêt ?

Complexité en $O(n\,m^2)$

avec n=|V| et m=|E|

Idée de la preuve

Notons $dist_f(x,y)$ la distance du sommet x au sommet y dans G_f.

Remarque (non triviale) :

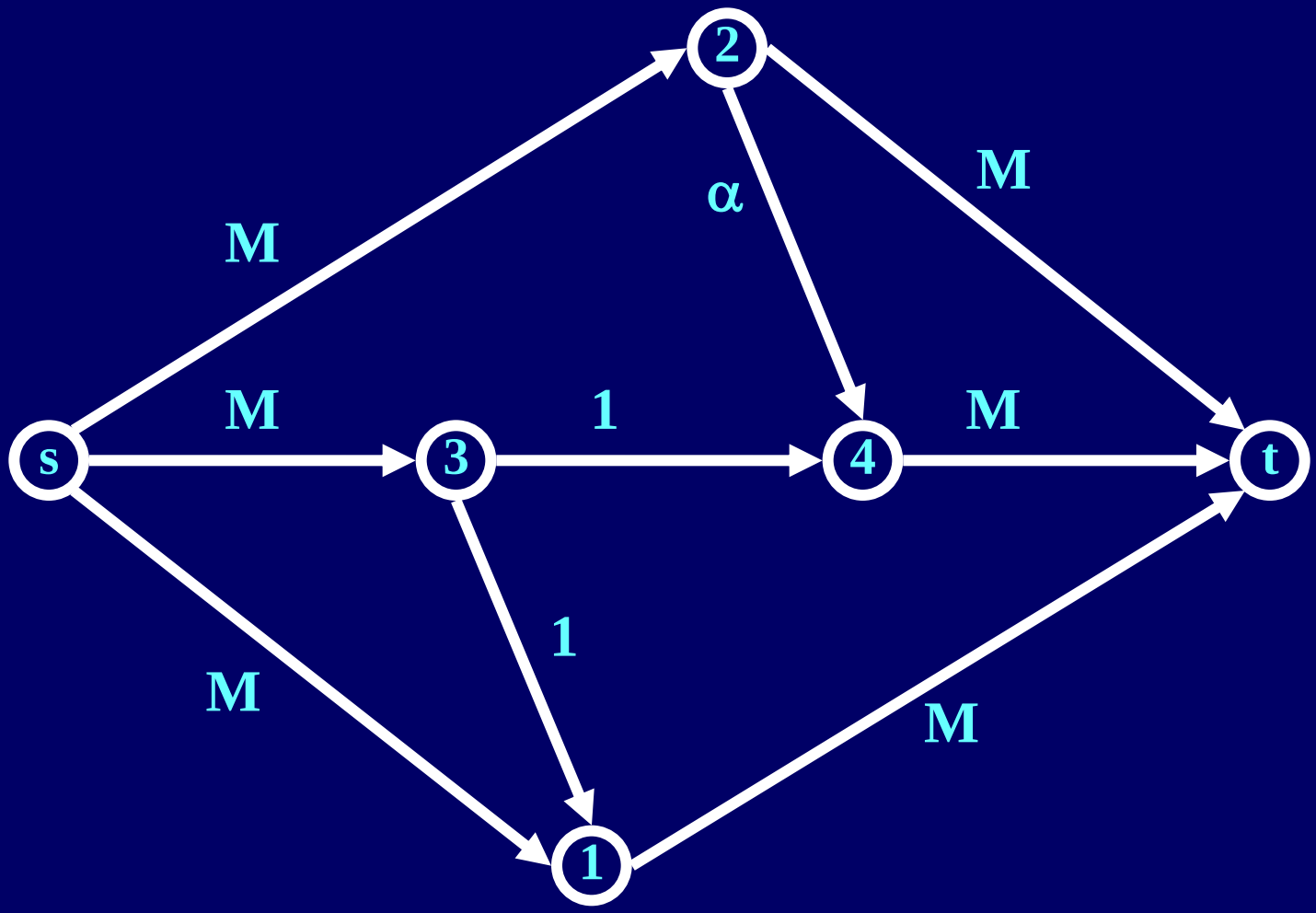
la fonction $dist_f(s,t)$ est croissante (non décroissante) en f.

Corollaire :

On a au plus $O(n\,m)$ augmentations

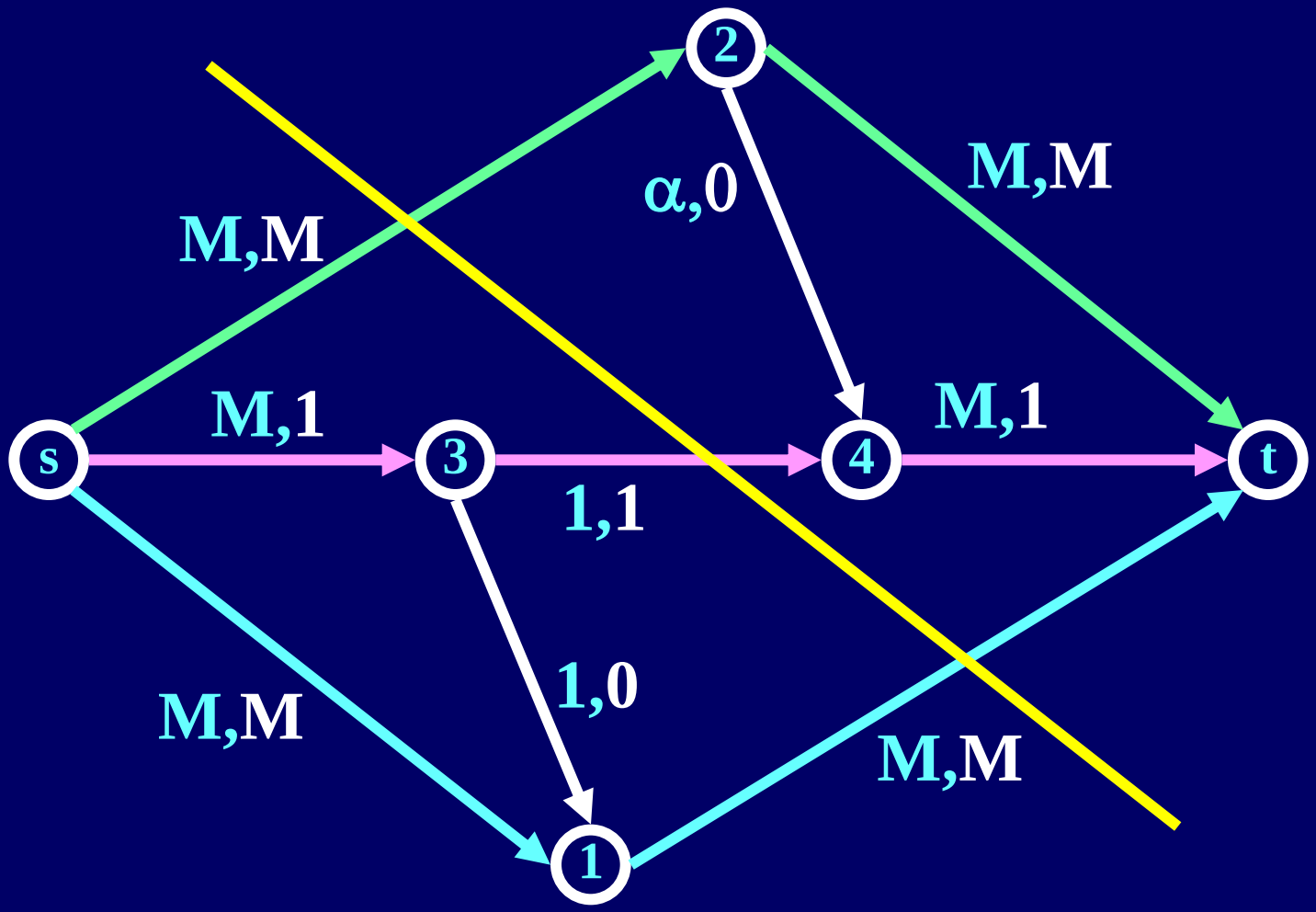
La méthode Ford & Fulkerson avec capacités réelles

Exemple de pb. avec Ford & Fulkerson



$$\alpha = \frac{\sqrt{5}-1}{2} \approx 0{,}618033988749895...$$

Le flot maximum (trivial …)



$$\alpha = \frac{\sqrt{5}-1}{2} \approx 0{,}618033988749895...$$

Une propriété

Soit α = (sqrt(5)-1)/2 ≈ 0,618

On définit

$a_n = \alpha^n$

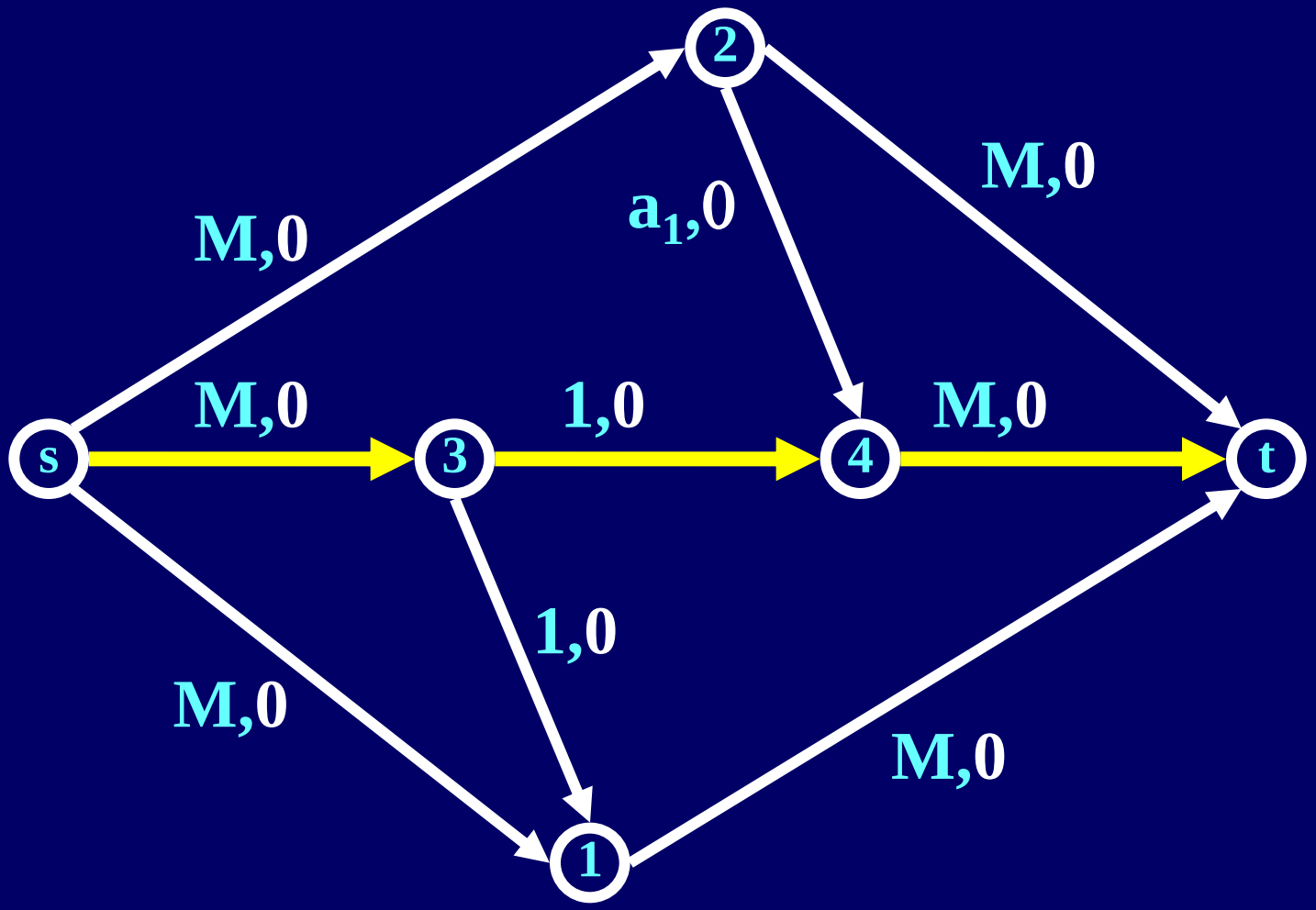
Et on obtient

$a_0 = 1$

$a_1 = \alpha$

$a_{n+2} = a_n - a_{n+1}$

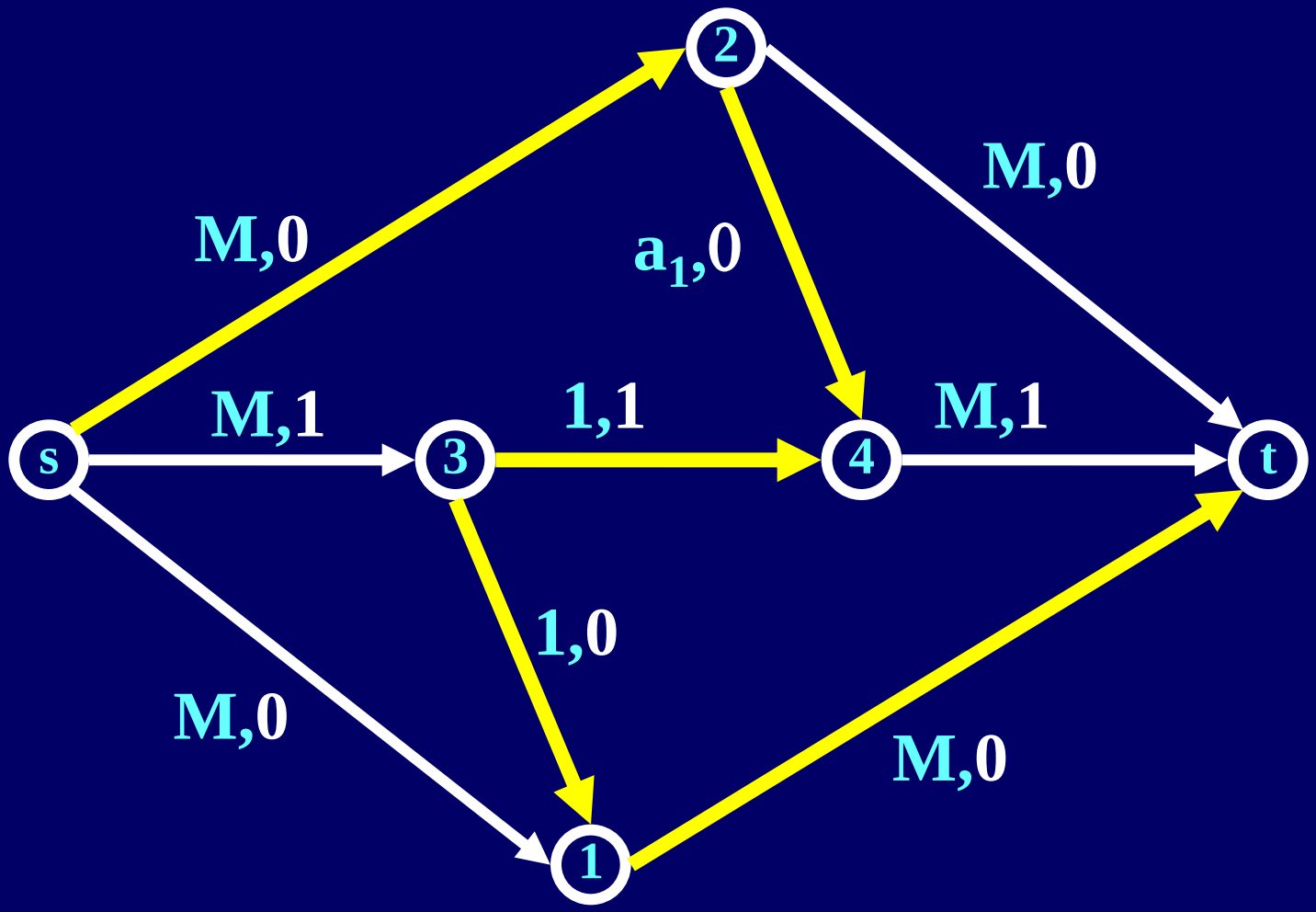
Les augmentations (1)



augmentation : 1

chaîne : s,3,4,t

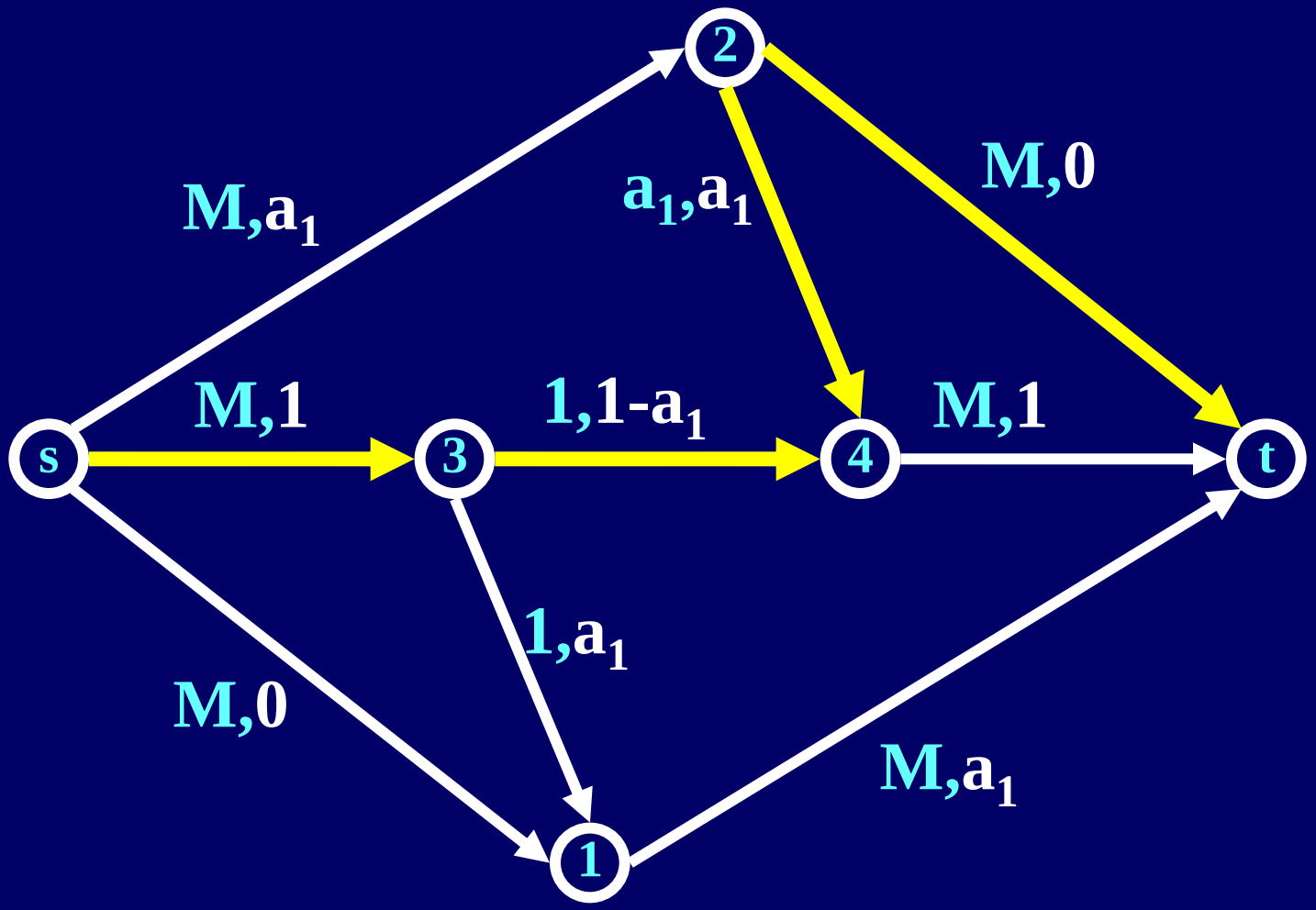
Les augmentations (2)



augmentation : a_1

chaîne : s,2,4,3,1,t

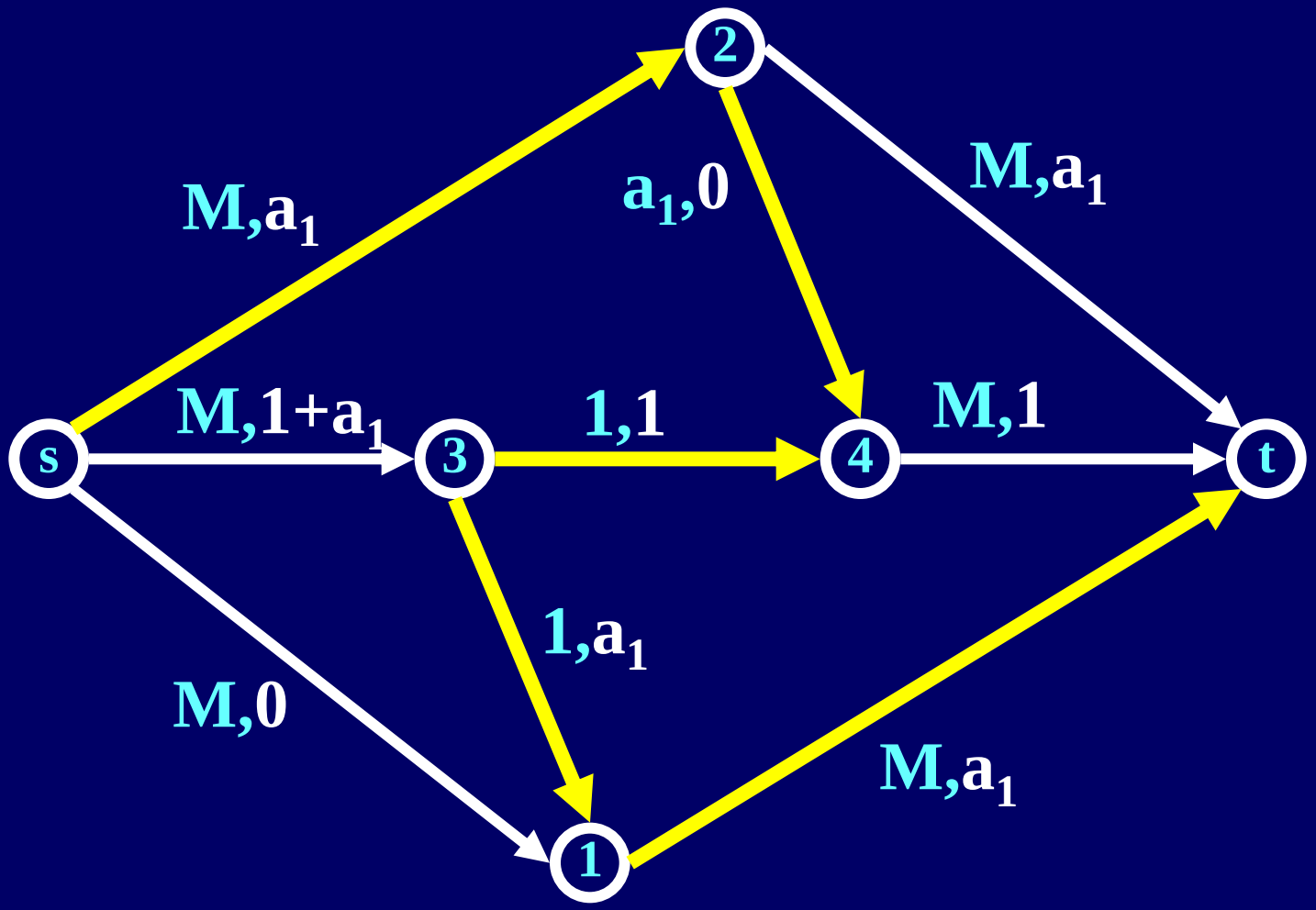
Les augmentations (3)



augmentation : a_1

chaîne : s,3,4,2,t

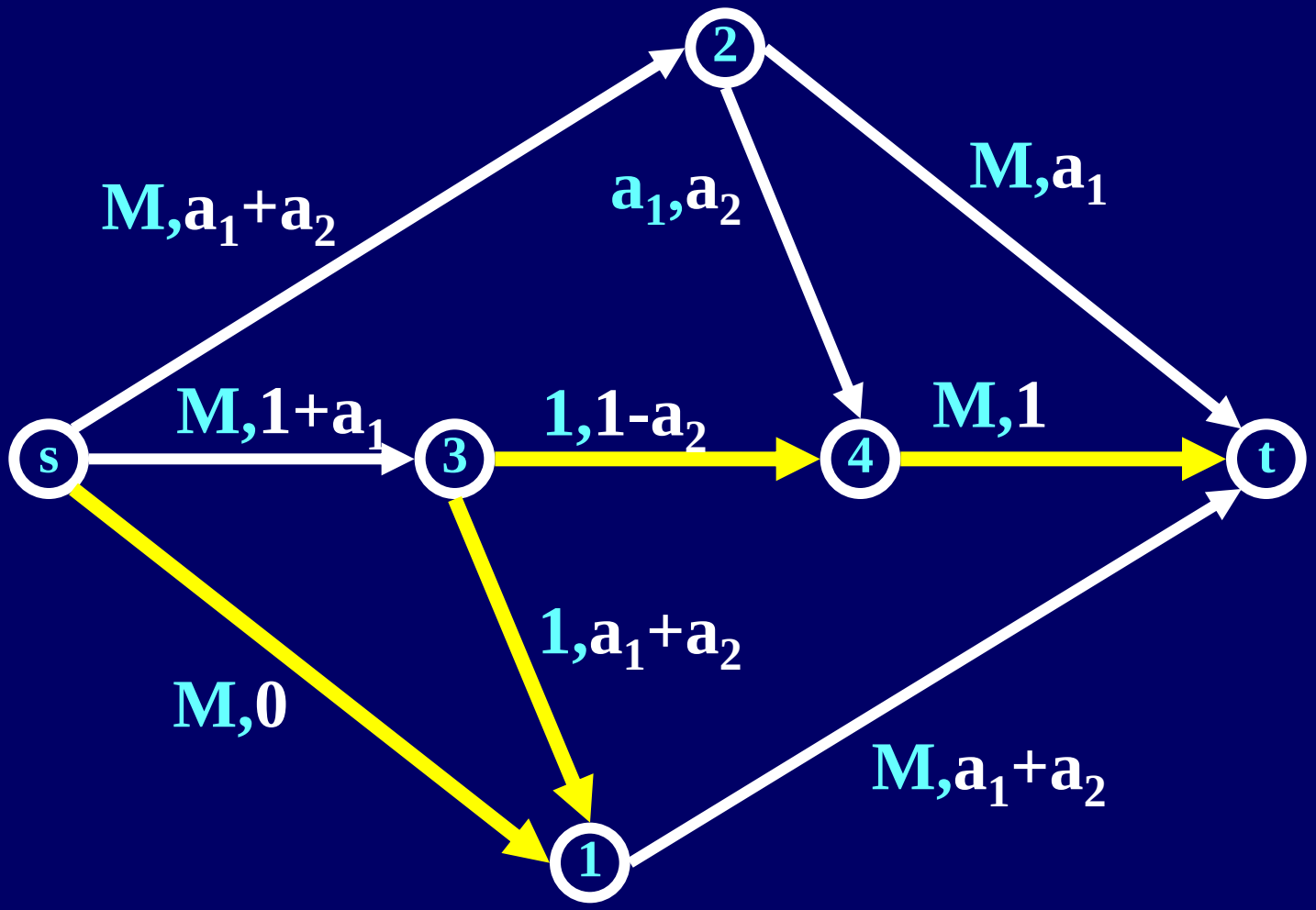
Les augmentations (4)



augmentation : a_2

chaîne : s,2,4,3,1,t

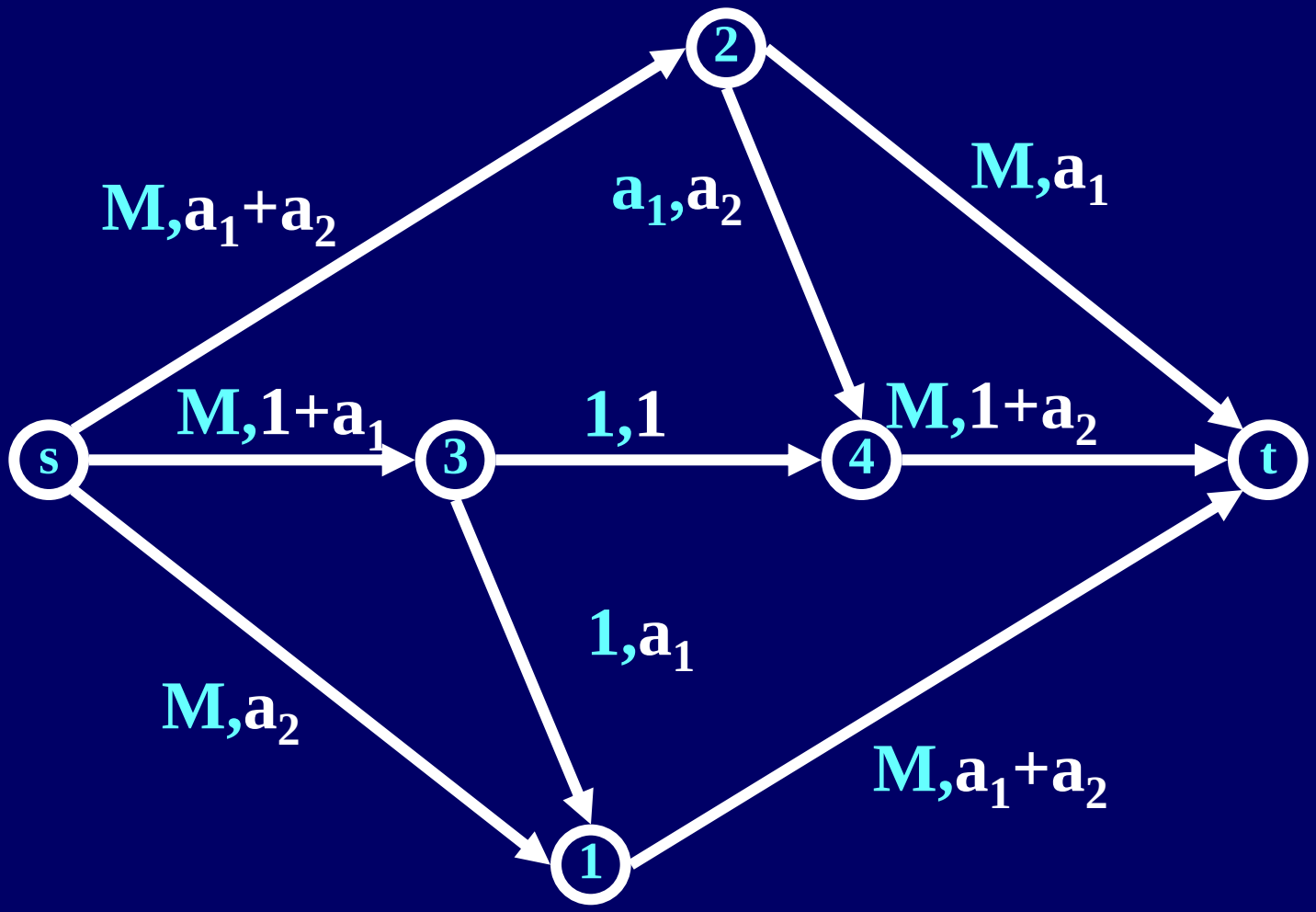
Les augmentations (5)



augmentation : a_2

chaîne : s,1,3,4,t

Les augmentations (6)



augmentation : a_3

	s,1	s,2	s,3	3,1	2,4	3,4	2,t	4,t	1,t
cap	M	M	M	1	α	1	M	M	M
1			1			1		1	
2		a_1	1	a_1	a_1	$1-a_1$		1	a_1
3		a_1	$1+a_1$	a_1	0	1	a_1	1	a_1
4		a_1+a_2	$1+a_1$	a_1+a_2	a_2	$1-a_2$	a_1	1	a_1+a_2
5	a_2	a_1+a_2	$1+a_1$	a_1	a_2	1	a_1	$1+a_2$	a_1+a_2
•									
•									
•									

	flot 3→1	flot 2→4	flot 3→4
4k+1	a_{2k}	a_{2k+1}	0
4k+2	a_{2k+2}	0	a_{2k+1}
4k+3	a_{2k+2}	a_{2k+1}	0
4k+4	0	a_{2k+3}	a_{2k+2}
4k+5	a_{2k+2}	a_{2k+3}	0

Le problème …

Le problème est que la suite des augmentations est 1, a_1, a_1, a_2, a_2, a_3, a_3, a_4, a_4, …

Ce qui nous fait converger vers

$$1+2\sum_{i=1}^{\infty}\alpha^i = 2\sum_{i=0}^{\infty}\alpha^i - 1 = 2\frac{1}{1-\alpha} - 1 = \frac{2-1+\alpha}{1-\alpha} = \frac{1+\alpha}{1-\alpha}$$

Ce qui est inférieur à 5, donc on est bien loin de 2M+1 !

Réseaux avec bornes inf et sup

Dans les réseaux que nous avons vu précédemment,

$$f(e) \geq 0.$$

Dans un cas plus général : toute arête e dispose de deux bornes, $b(e)$ et $c(e)$ et on exige

$$b(e) \leq f(e) \leq c(e)$$

Ford & Fulkerson ?

Peut-on adapter

l'algorithme de Ford & Fulkerson ?

OUI …. mais …..

Réseau résiduel

Si on a un arc de *u* vers *v*, alors

on met dans G_f un arc de *u* vers *v* de capacité

$$c_f(u,v) = c(u,v) - f(u,v)$$

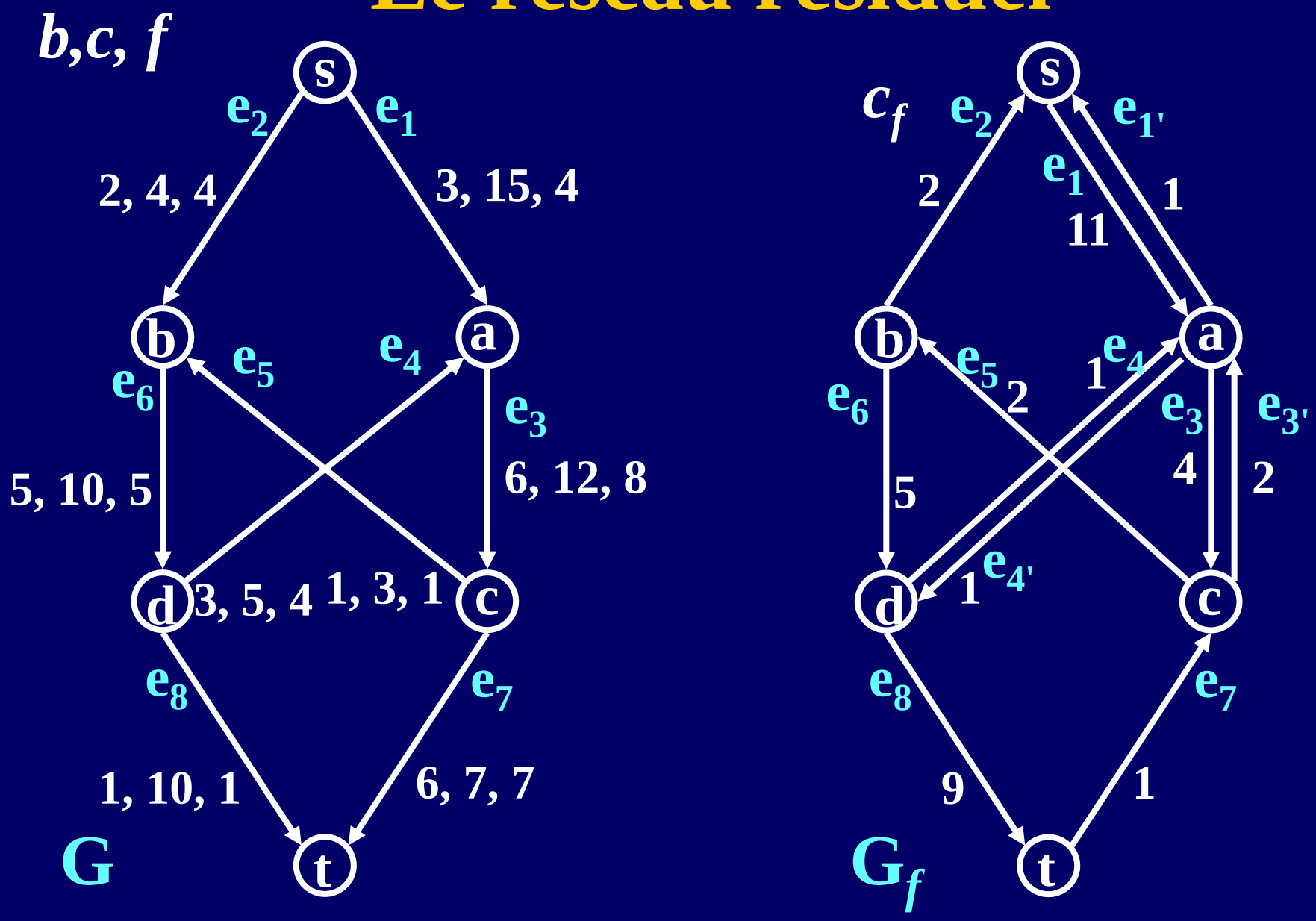
(si $c(u,v) > f(u,v)$)

on met dans G_f un arc de *v* vers *u* de capacité

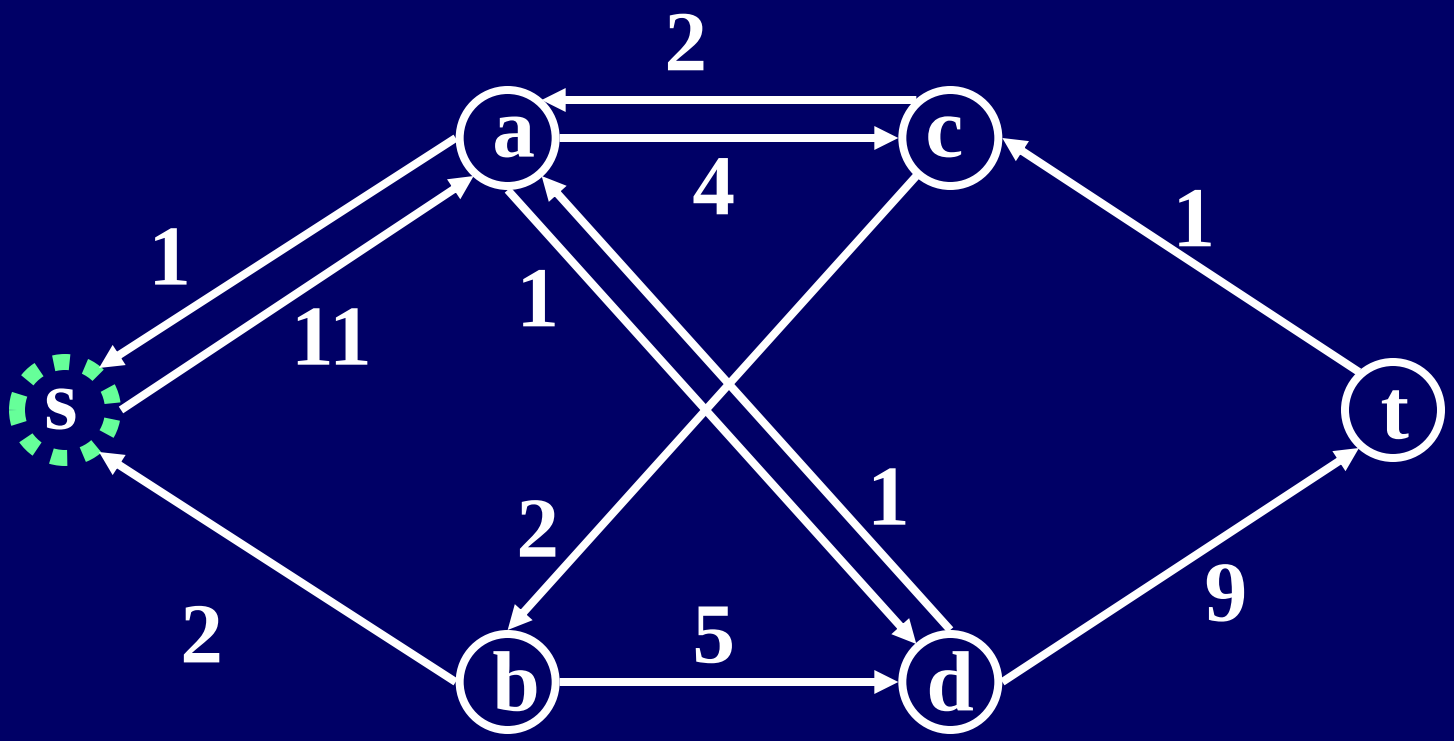
$$c_f(v,u) = f(u,v) - b(u,v)$$

($f(u,v) > b(u,v)$)

Le réseau résiduel

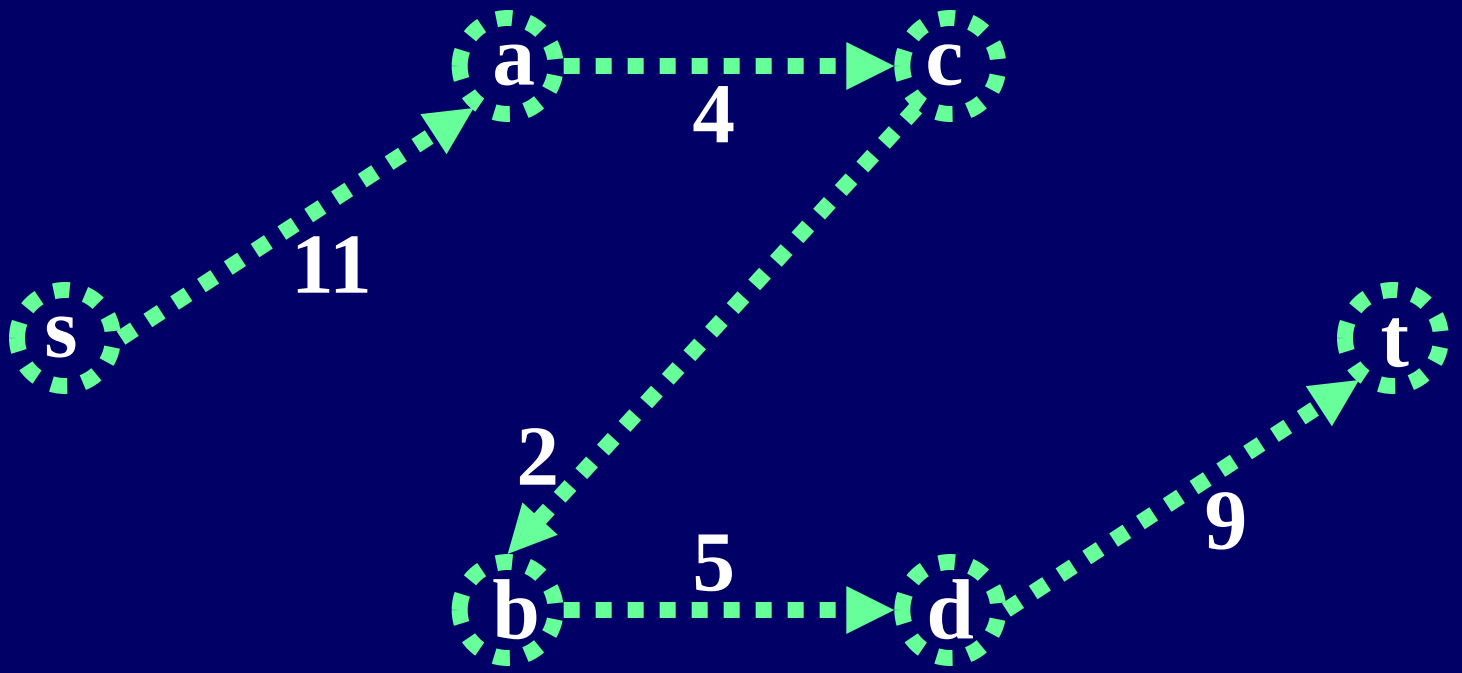


Recherche de chaîne augmentante



c_f

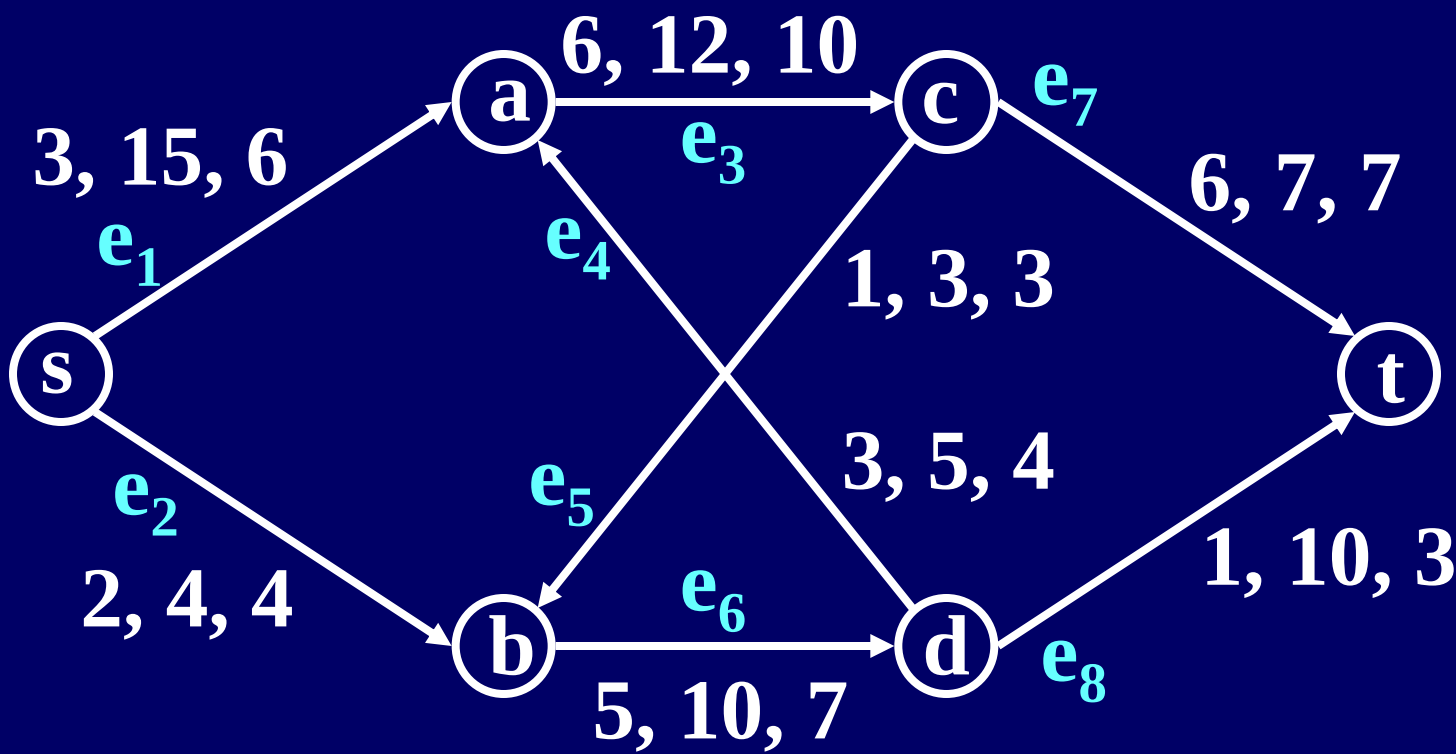
La chaîne augmentante



c_f

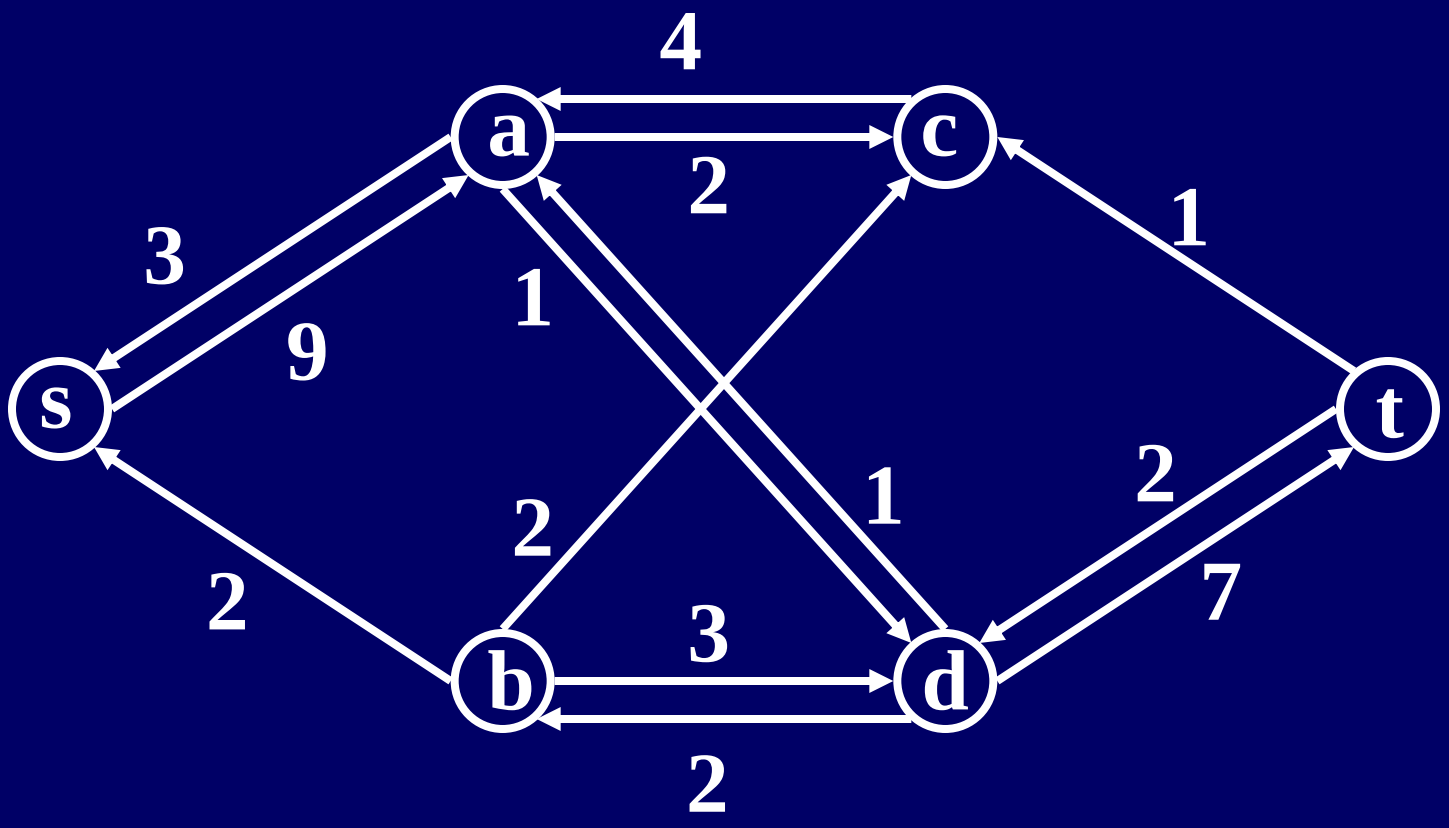
On peut faire une augmentation de min{11,4,2,5,9}=2.

Le flot résultant



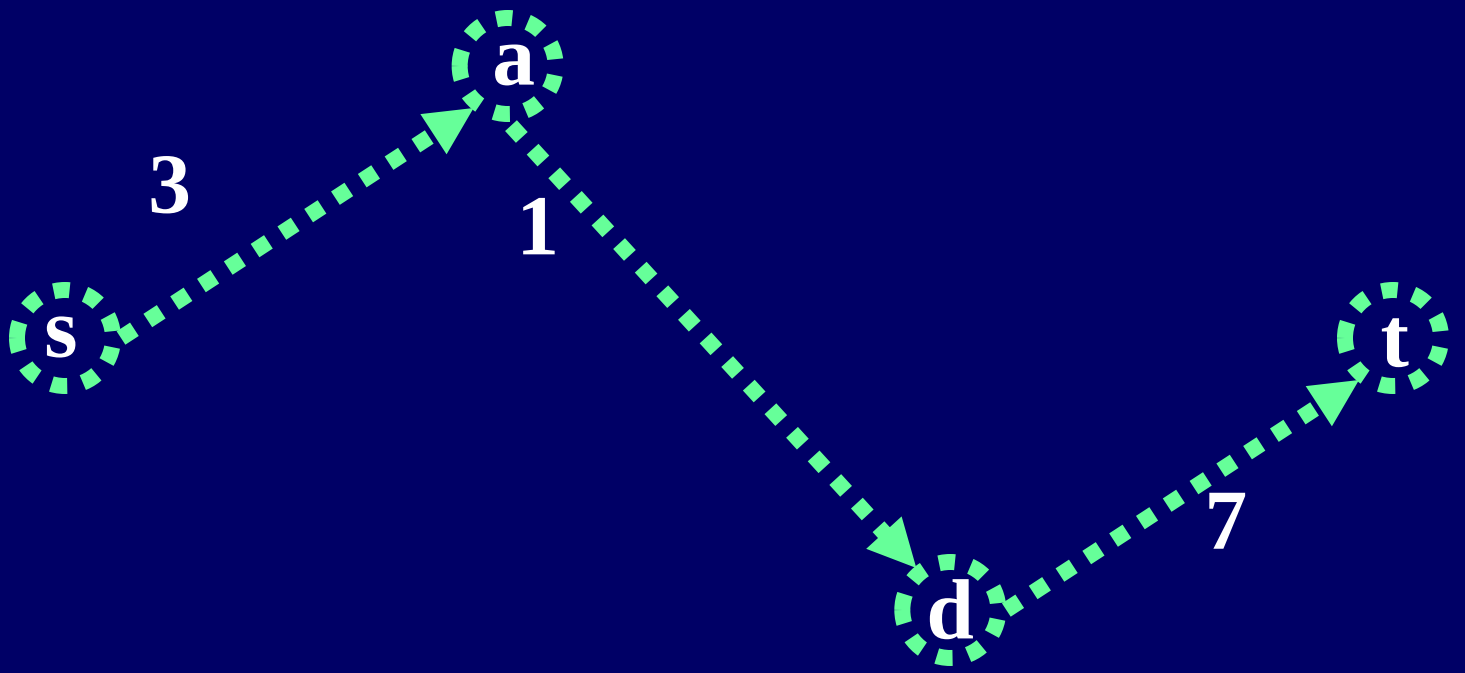
b, c, f

Le réseau résiduel



c_f

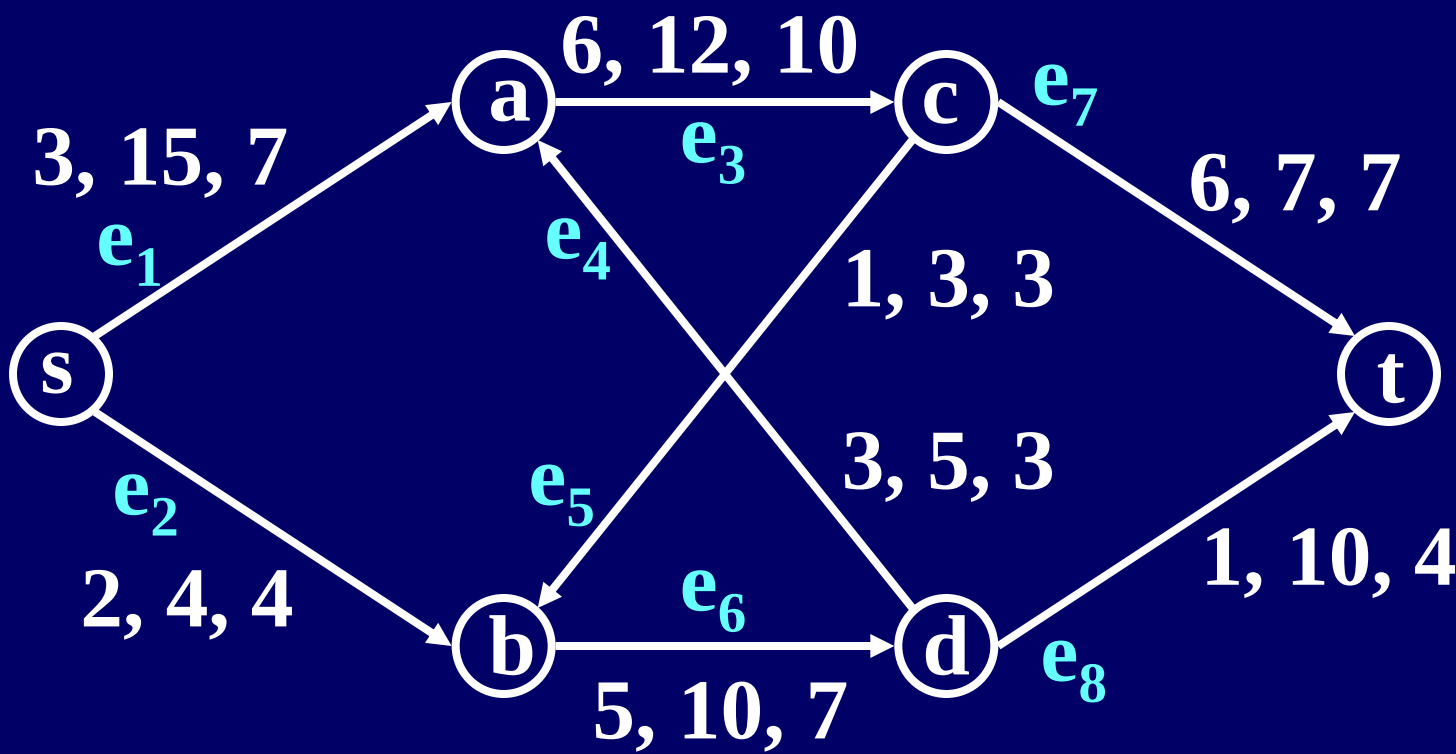
La chaîne augmentante



c_f

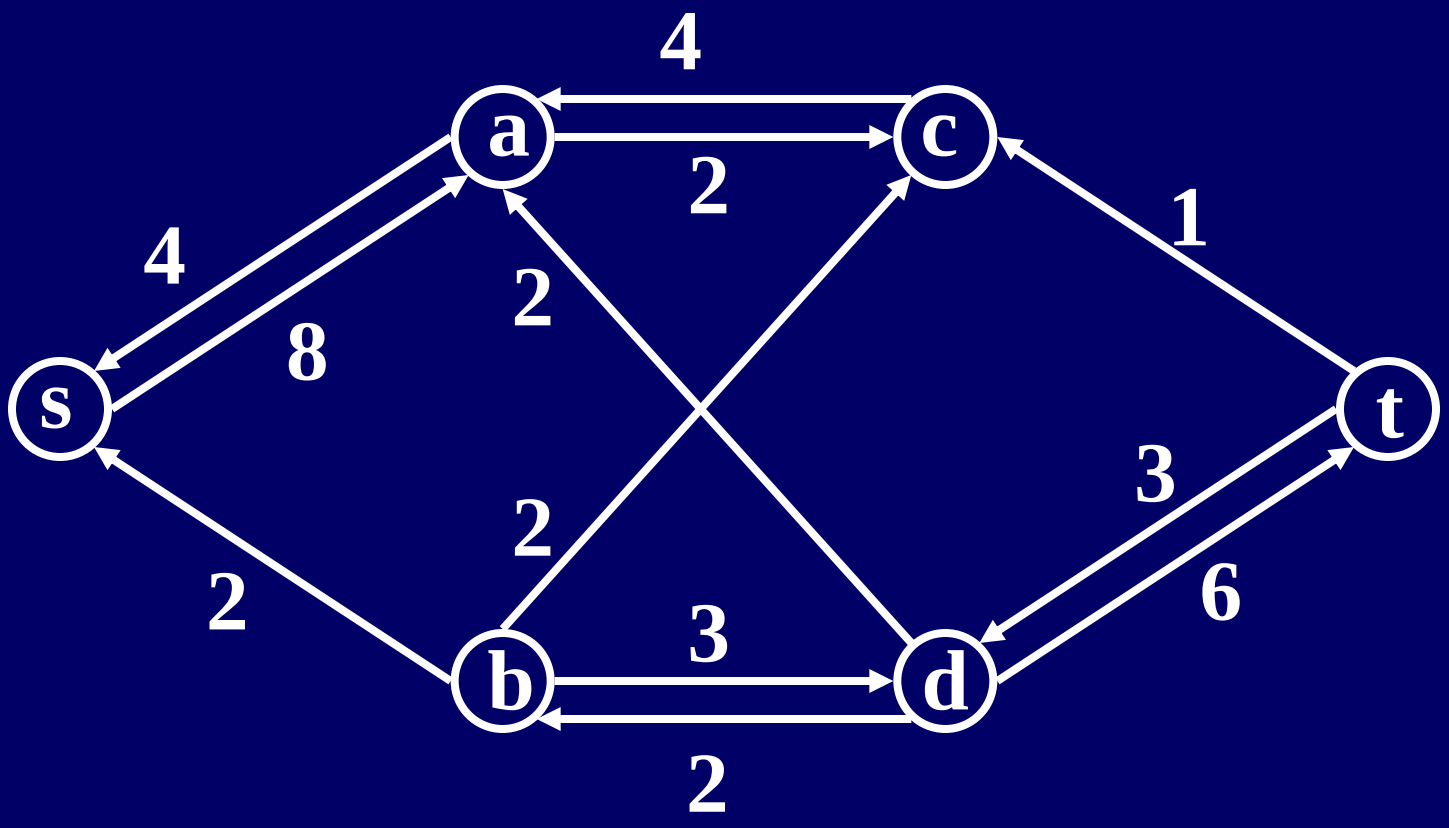
On peut faire une augmentation de min{3,1,7}=1.

Le flot résultant



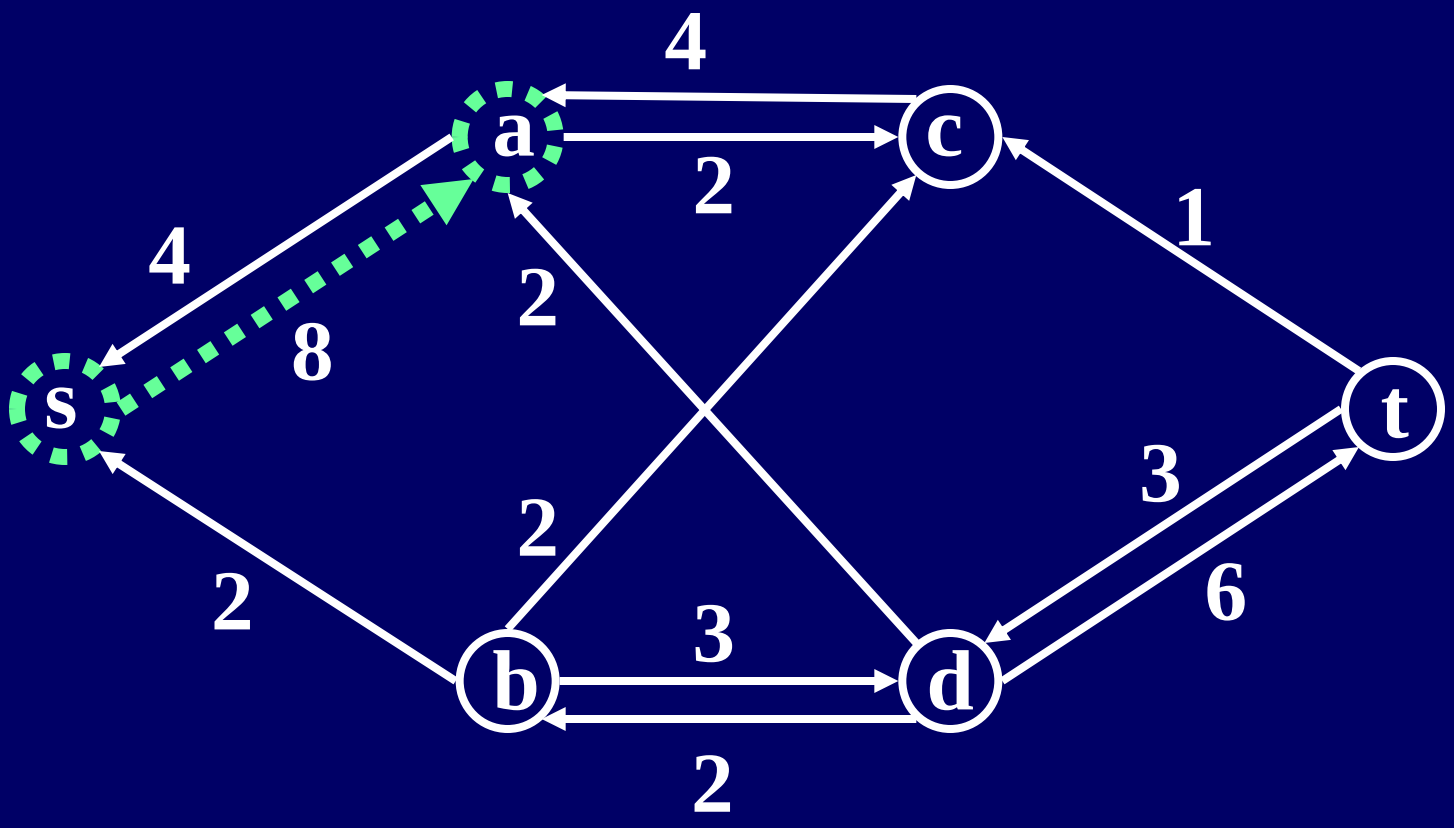
b, c, f

Le réseau résiduel



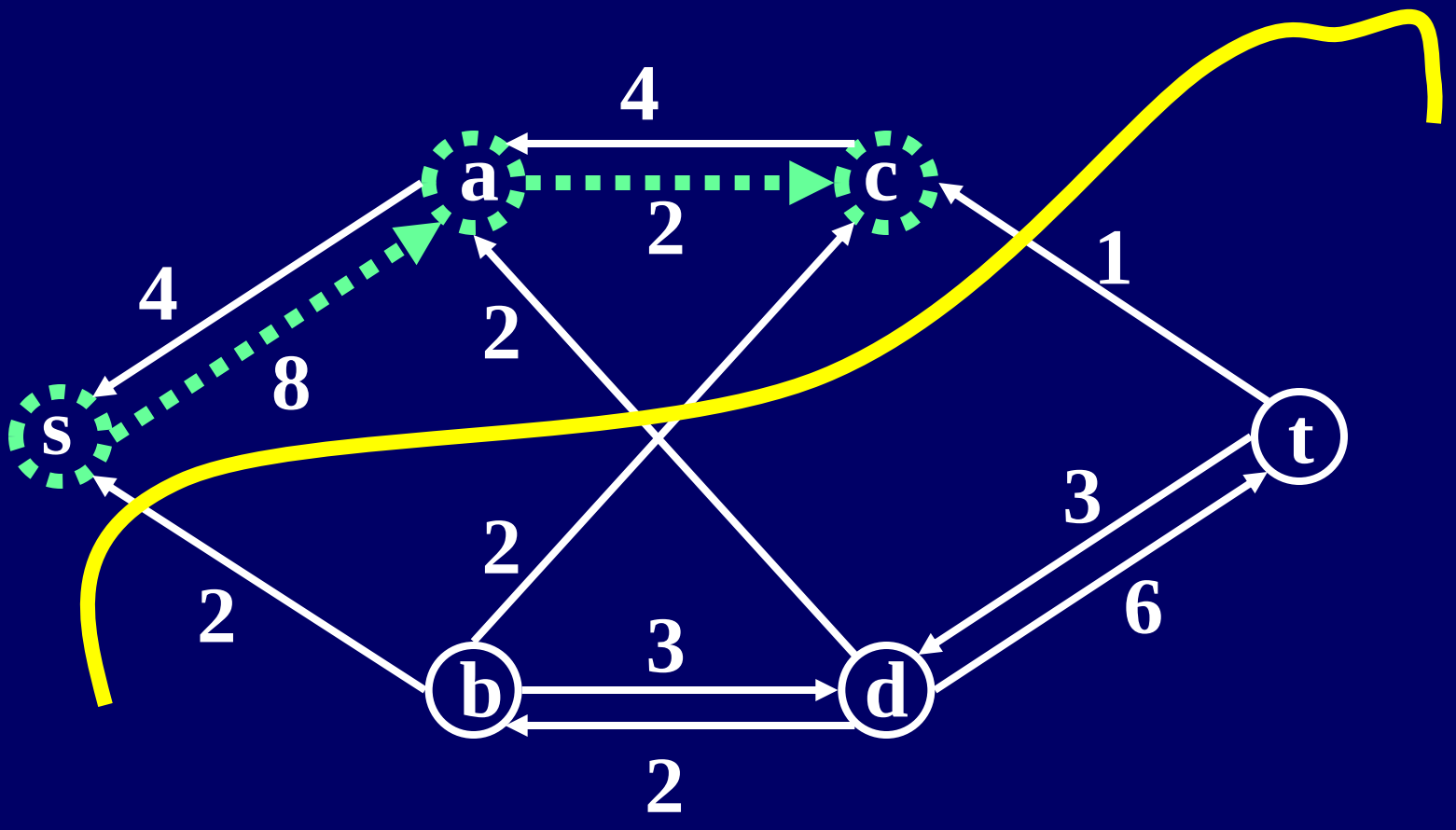
c_f

Recherche de chaîne augmentante



c_f

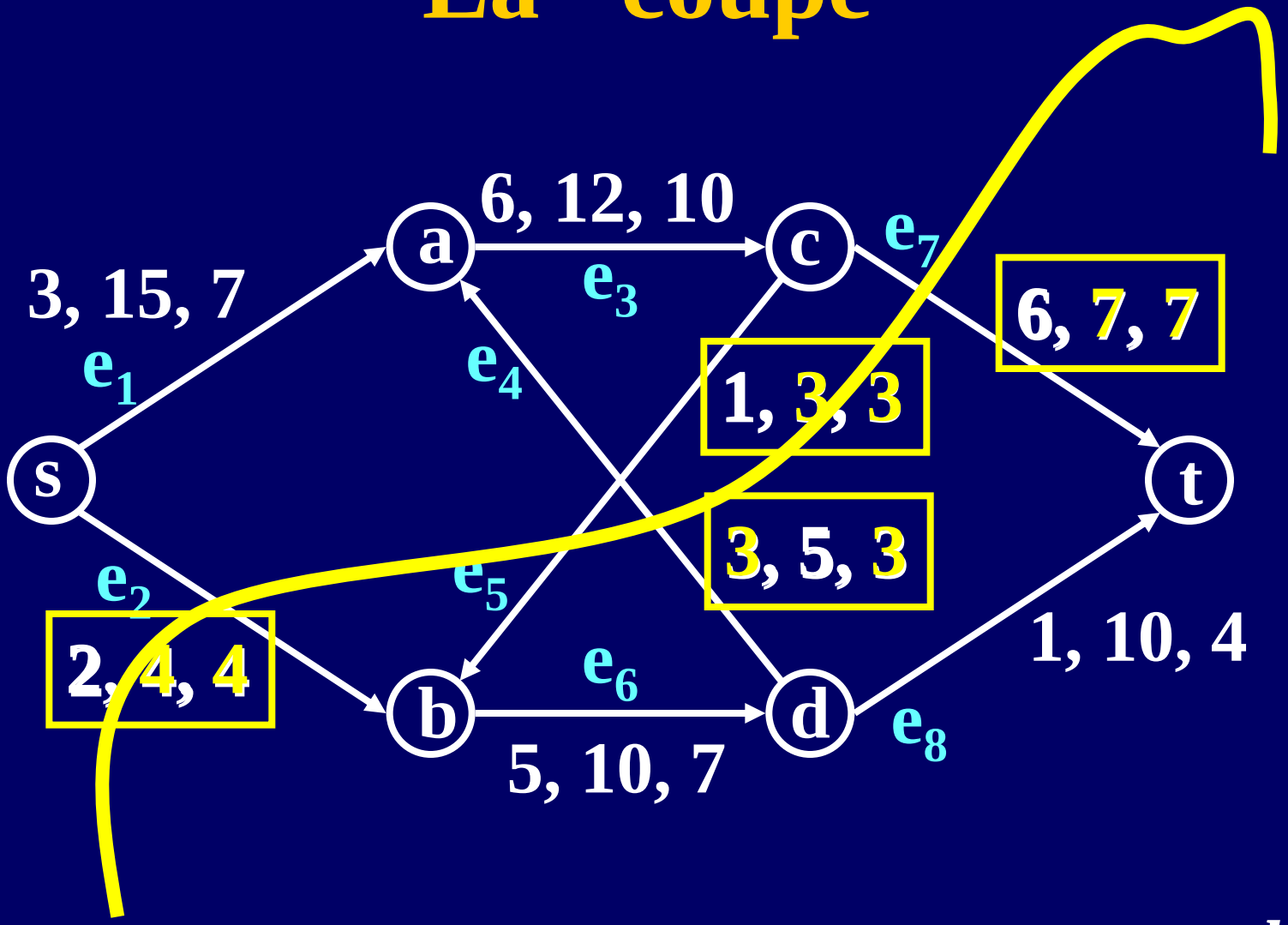
Recherche de chaîne augmentante



Et on ne peut plus continuer !

c_f

La "coupe"


6, 12, 10
a
c
e7
3, 15, 7
e3
6, 7, 7
e1
e4
1, 3, 3
s
t
3, 5, 3
e2
e5
2, 4, 4
1, 10, 4
e6
b
d
e8
5, 10, 7

b, c, f

La coupe

Définition : $c(S) = \Sigma_{e \in (S;S')} c(e) - \Sigma_{e \in (S';S)} b(e)$

Propriété : **pour tout flot f et pour toute coupe S**

$$|f| \leq c(S)$$

Corollaire : Si $|f| = c(S)$ alors le flot est maximum et la coupe S est de capacité minimum.

La coupe (suite)

Le théorème du flot max – coupe min se déduit exactement comme dans le cas précédent;

Mais …

Il reste un seul problème …

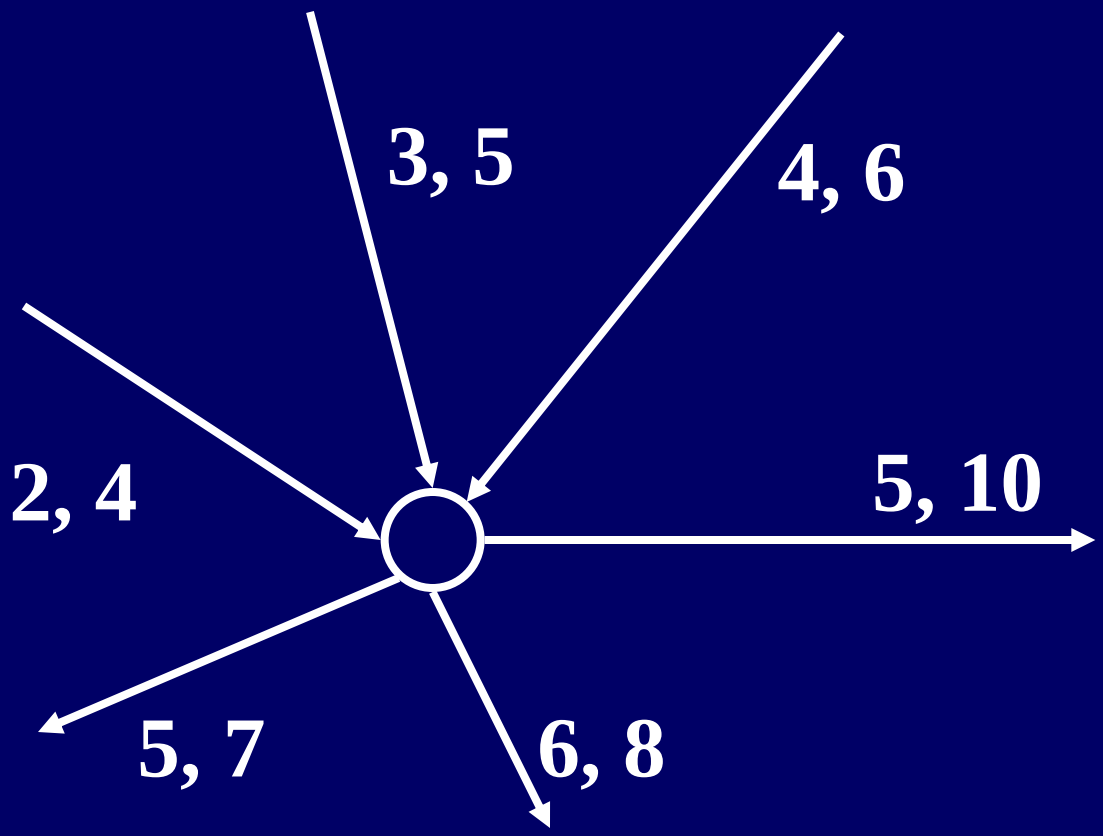
Dans le cas classique, nous avons commencé avec un flot nul, puis augmenté

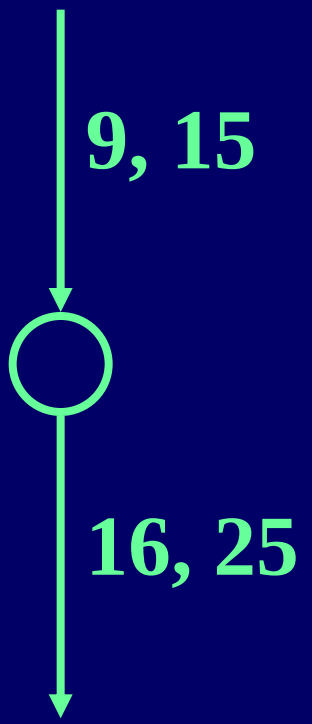
Mais ici un flot nul n'est pas un flot "correcte" (ne respecte pas $b(e) \leq f(e)$)

Un flot "correcte" on appelle un flot admissible.

Recherche de flot admissible

Cas simple où il n'en existe pas :

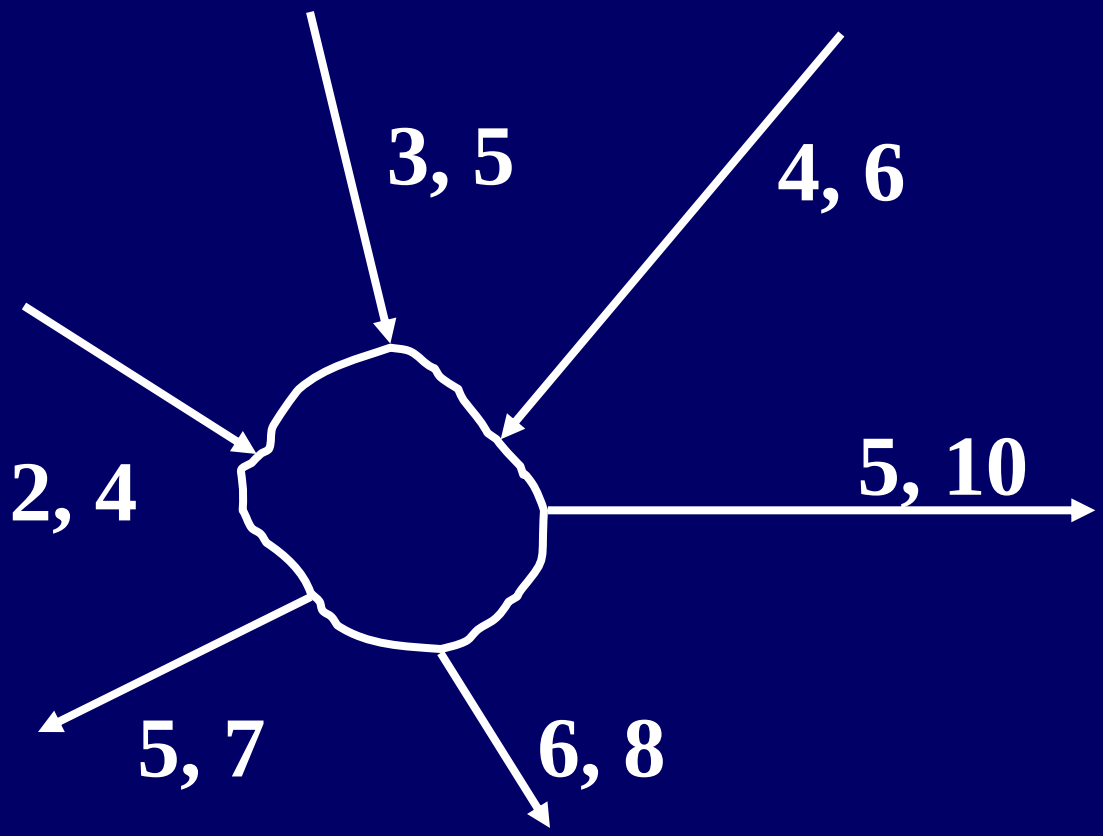


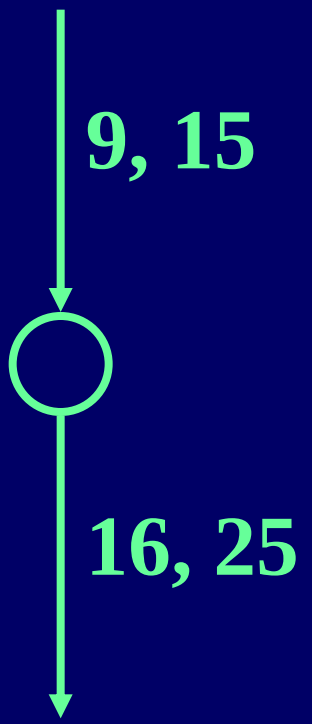


b, c

Recherche de flot admissible

Cas plus compliqué où il n'en existe pas :





b, c

Recherche de flot admissible

On rajoute une "source" $\underline{s}$, et un "puits" $\underline{t}$.

On construit un réseau "classique" avec borne inférieure 0 et borne supérieure $\underline{c}$.

Pour chaque nœud v on rajoute

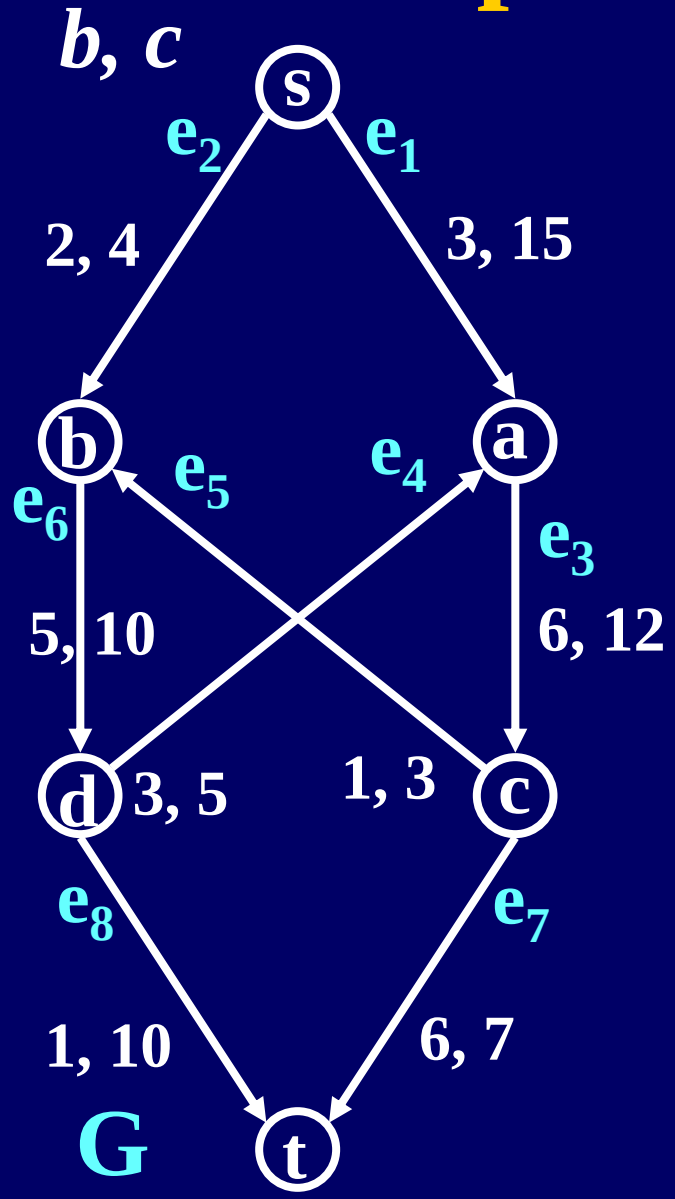
- un arc $(v, \underline{t})$ avec $\underline{c}(v, \underline{t}) = \Sigma_{e \in \mathrm{Out}(v)} b(e)$
- un arc $(\underline{s}, v)$ avec $\underline{c}(\underline{s}, v) = \Sigma_{e \in \mathrm{In}(v)} b(e)$

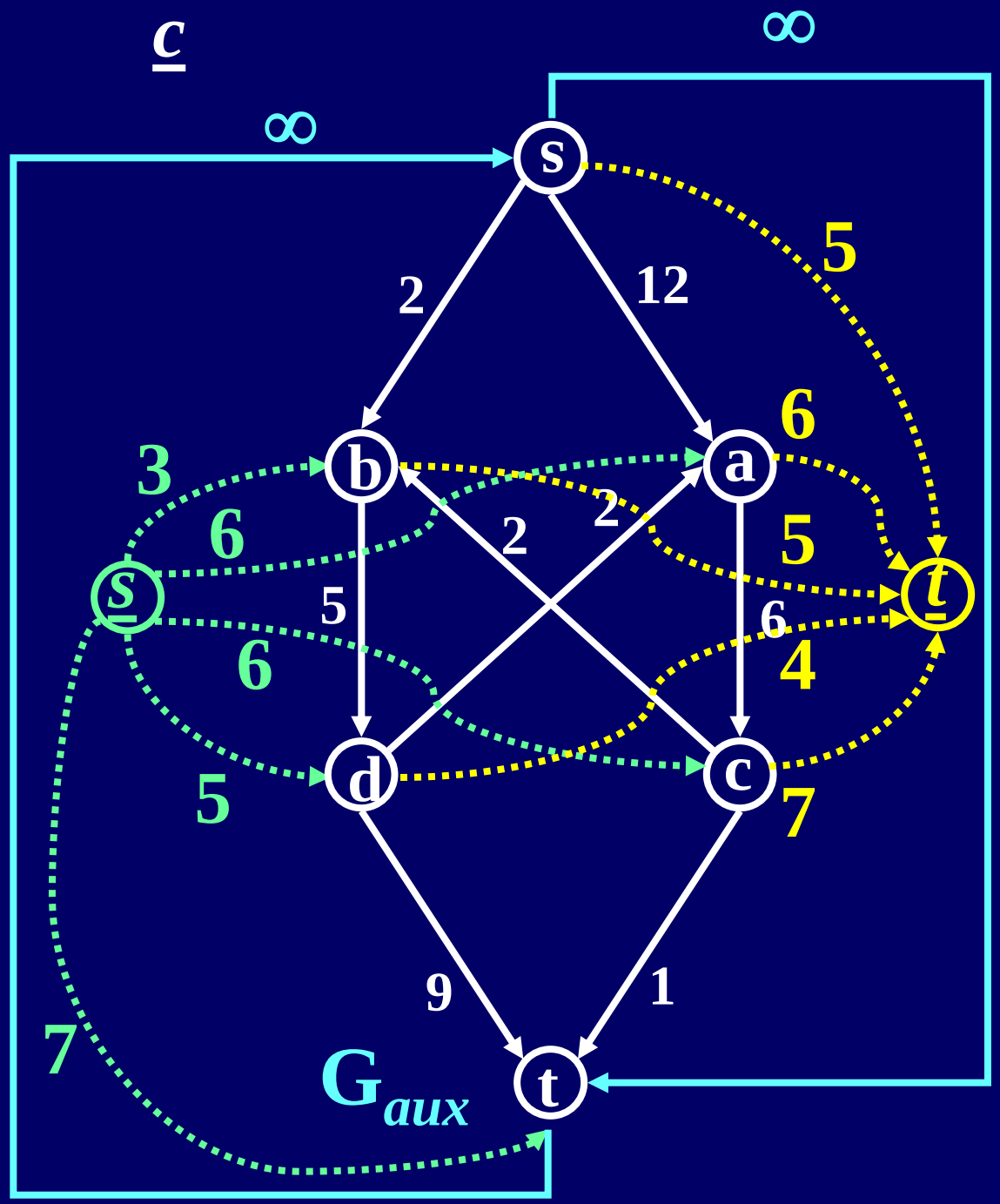
Pour toute les autres arcs on définit

$$\underline{c}(e) = c(e) - b(e)$$

On rajoute deux arcs : $\underline{c}(s, t) = \infty$ et $\underline{c}(t, s) = \infty$.

Exemple


b, c
s
e2
e1
2, 4
3, 15
b
a
e5
e4
e6
e3
5, 10
6, 12
d
3, 5
1, 3
c
e8
e7
1, 10
6, 7
G
t


c
∞
∞
s
2
12
5
6
3
b
a
6
2
2
5
s
5
t
6
6
4
5
d
c
7
9
1
7
t
G aux

La caractérisation

Théorème : Le réseau original admet un flot admissible si et seulement si le flot maximum du réseau modifié sature toutes les arêtes sortants de s.

Remarque : dans ce cas les arcs entrants en *t* sont aussi saturés.

La preuve

SI :

Supposons avoir un flot maximum $\underline{f}$ dans le réseau modifié.

Pour le réseau original on défini pour toutes les arcs $f(e) = \underline{f}(e) + b(e)$.

Comme nous avons $0 \leq \underline{f}(e) \leq \underline{c}(e) = c(e) - b(e)$

on obtient $b(e) \leq f(e) \leq c(e)$

La preuve (suite)

La conservation des flots : soit v un sommet, $v \neq s, t$

nous avons dans G_{aux}

$$\Sigma_{e \in \mathrm{In}(v)} \underline{f}(e) + \underline{f}(\underline{s},v) = \Sigma_{e \in \mathrm{Out}(v)} \underline{f}(e) + \underline{f}(v,\underline{t})$$

mais comme

$$\underline{f}(\underline{s},v) = \underline{c}(\underline{s}, v) = \Sigma_{e \in \mathrm{In}(v)} b(e)$$

et

$$\underline{f}(v,\underline{t}) = \underline{c}(v, \underline{t}) = \Sigma_{e \in \mathrm{Out}(v)} b(e)$$

on a

$$\Sigma_{e \in \mathrm{In}(v)} f(e) = \Sigma_{e \in \mathrm{Out}(v)} f(e)$$

et ainsi le flot f est admissible.

La preuve (3)

Seulement si :

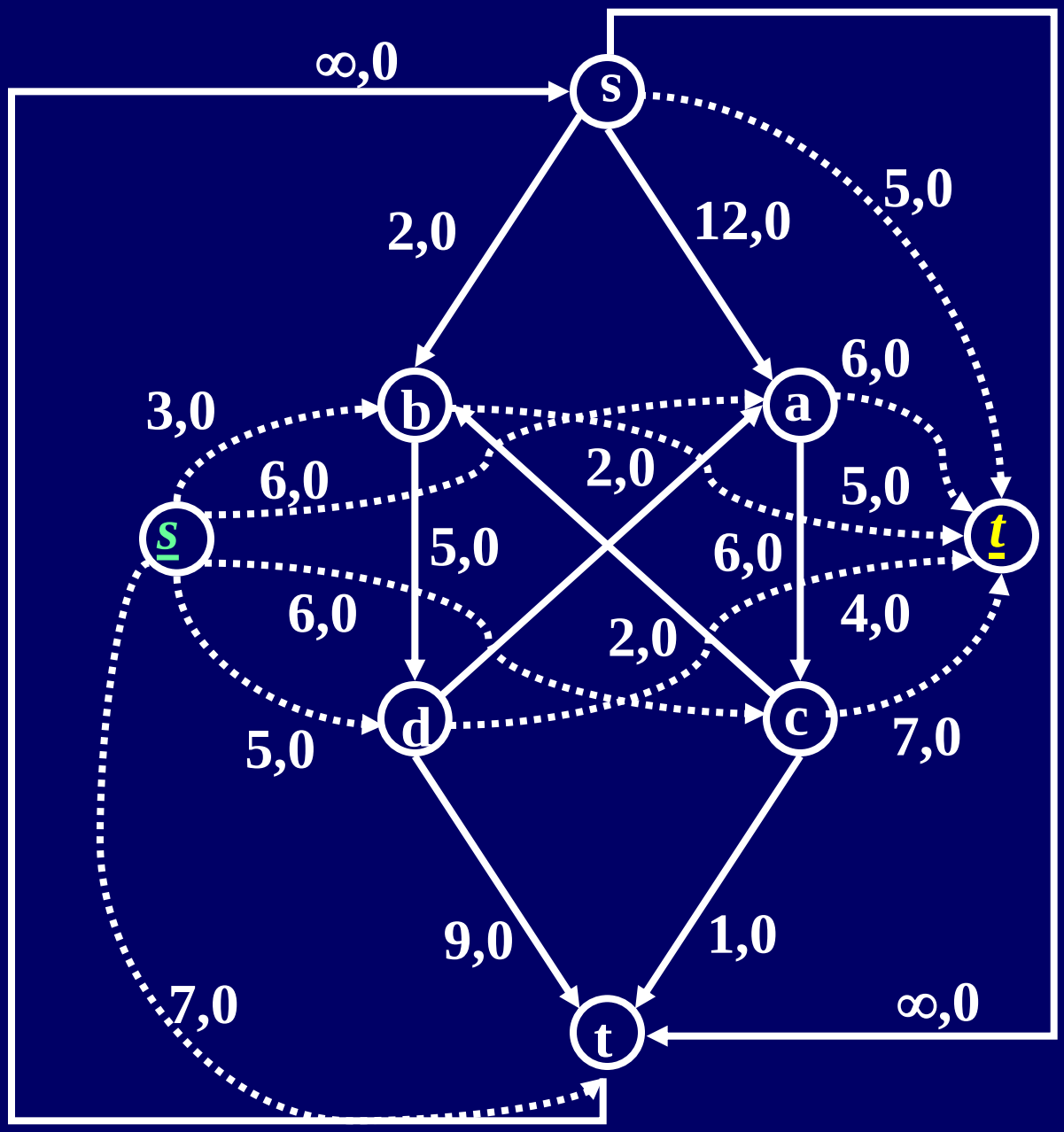
En fait la preuve est "réversible".

En effet, si on a un flot admissible, on peut en déduire un flot f dans G_{aux}.

Le résultat vérifie la conservation des flots dans tous les sommets, sauf éventuellement *s* et *t*.

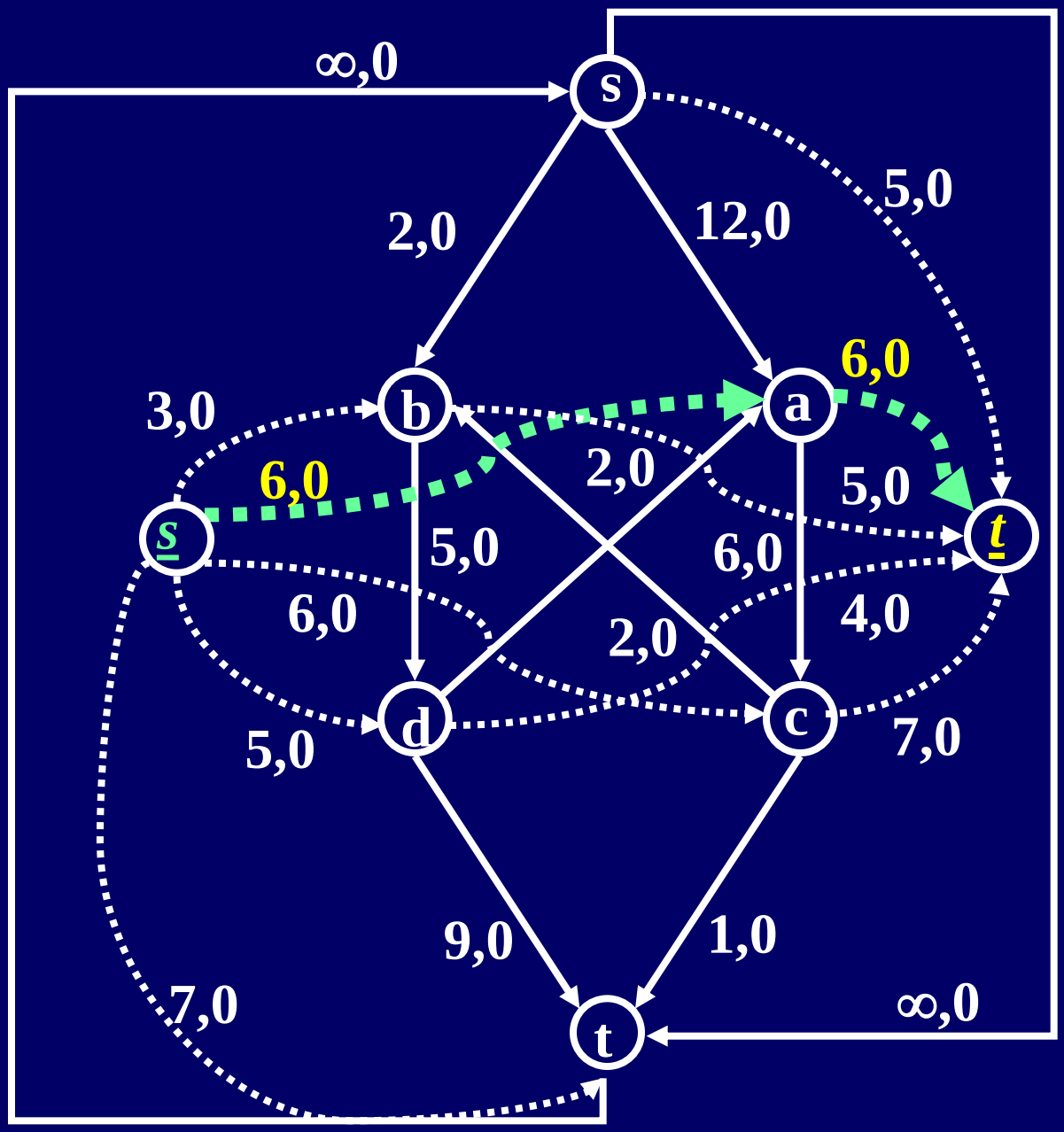
Mais ceci peut être assuré, en utilisant les arcs de capacité infinie entre elles.

L'exemple



G_{aux}

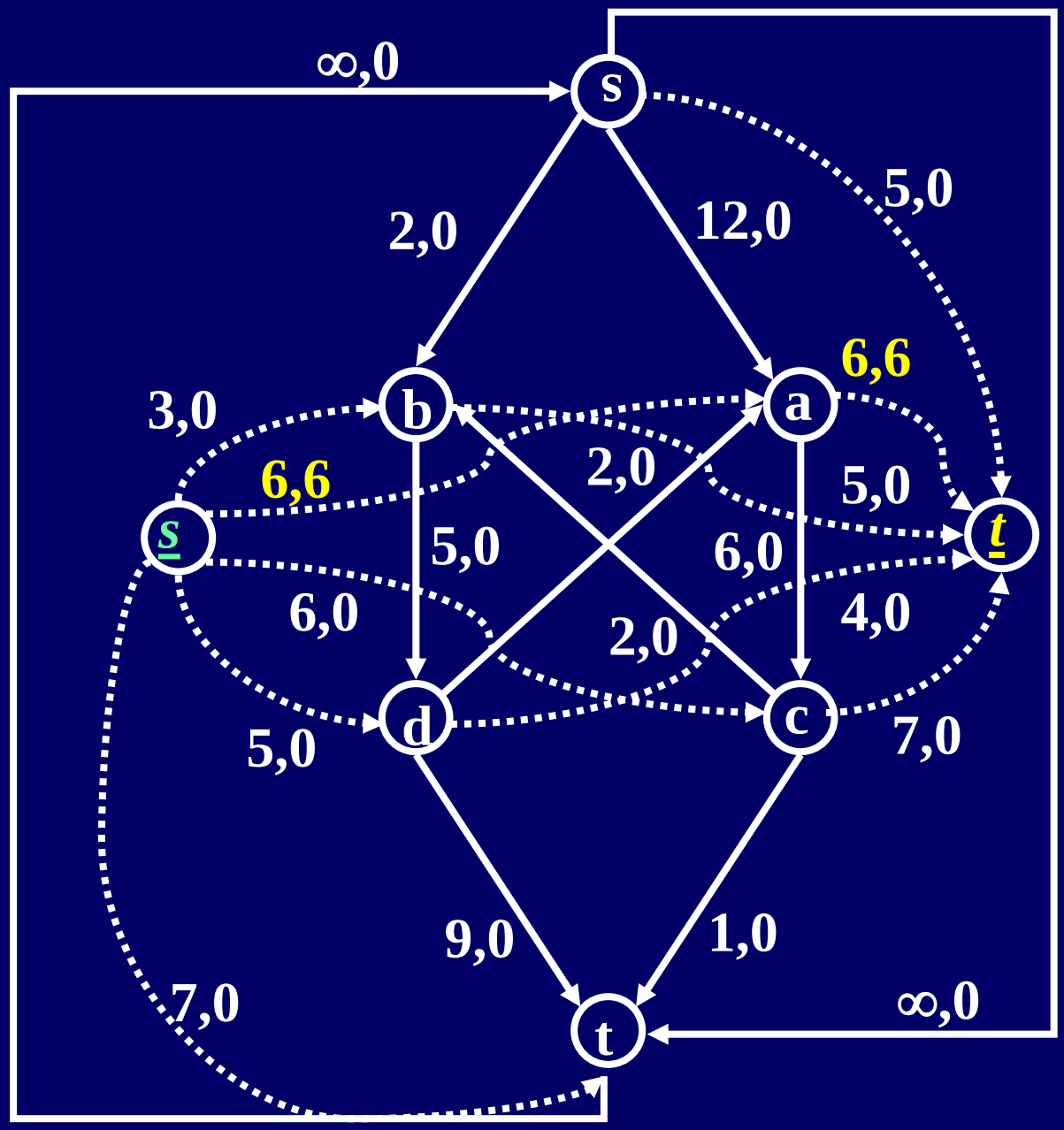
L'exemple
(2)



G_{aux}

L'exemple

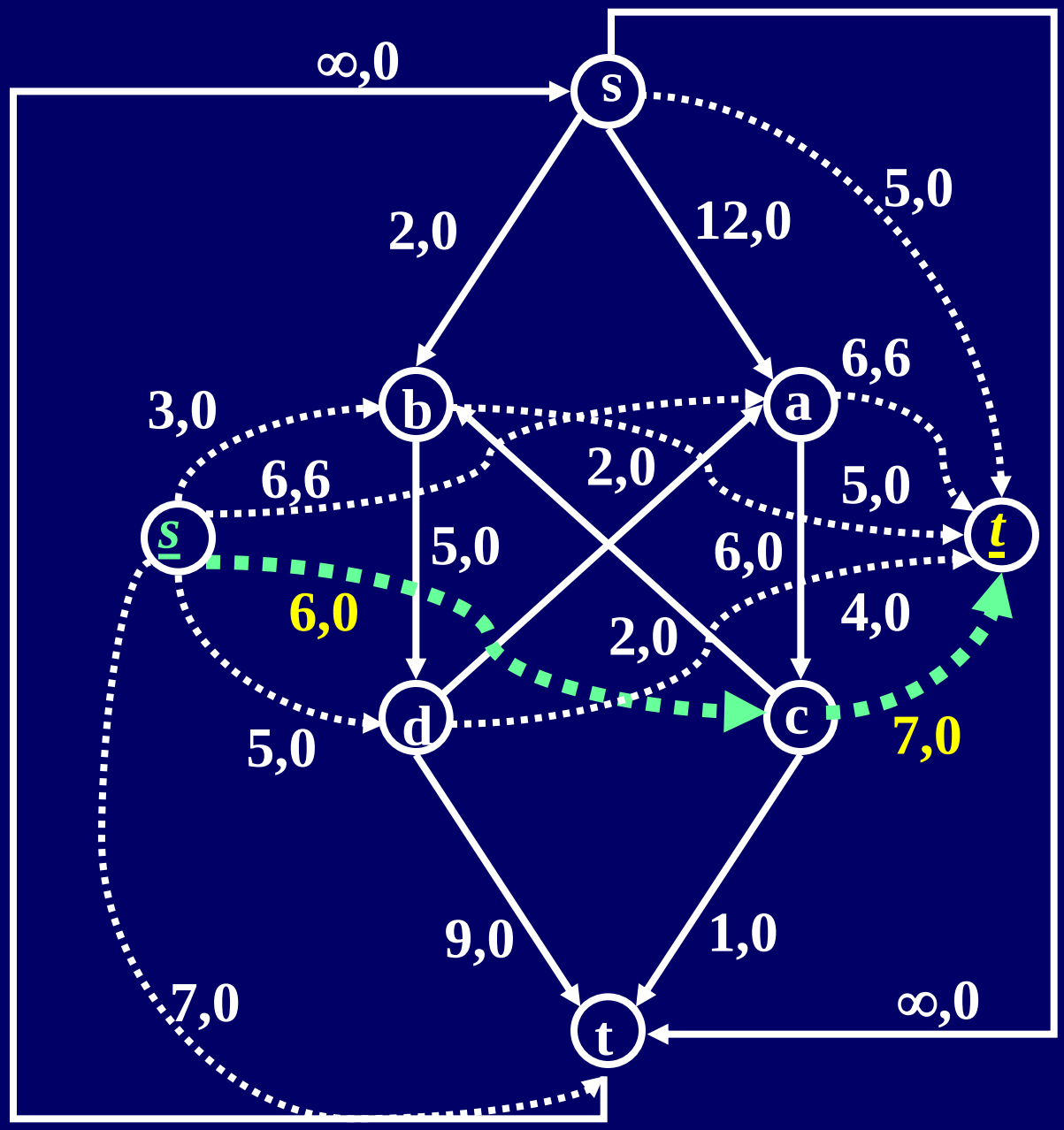
(3)



G_{aux}

L'exemple

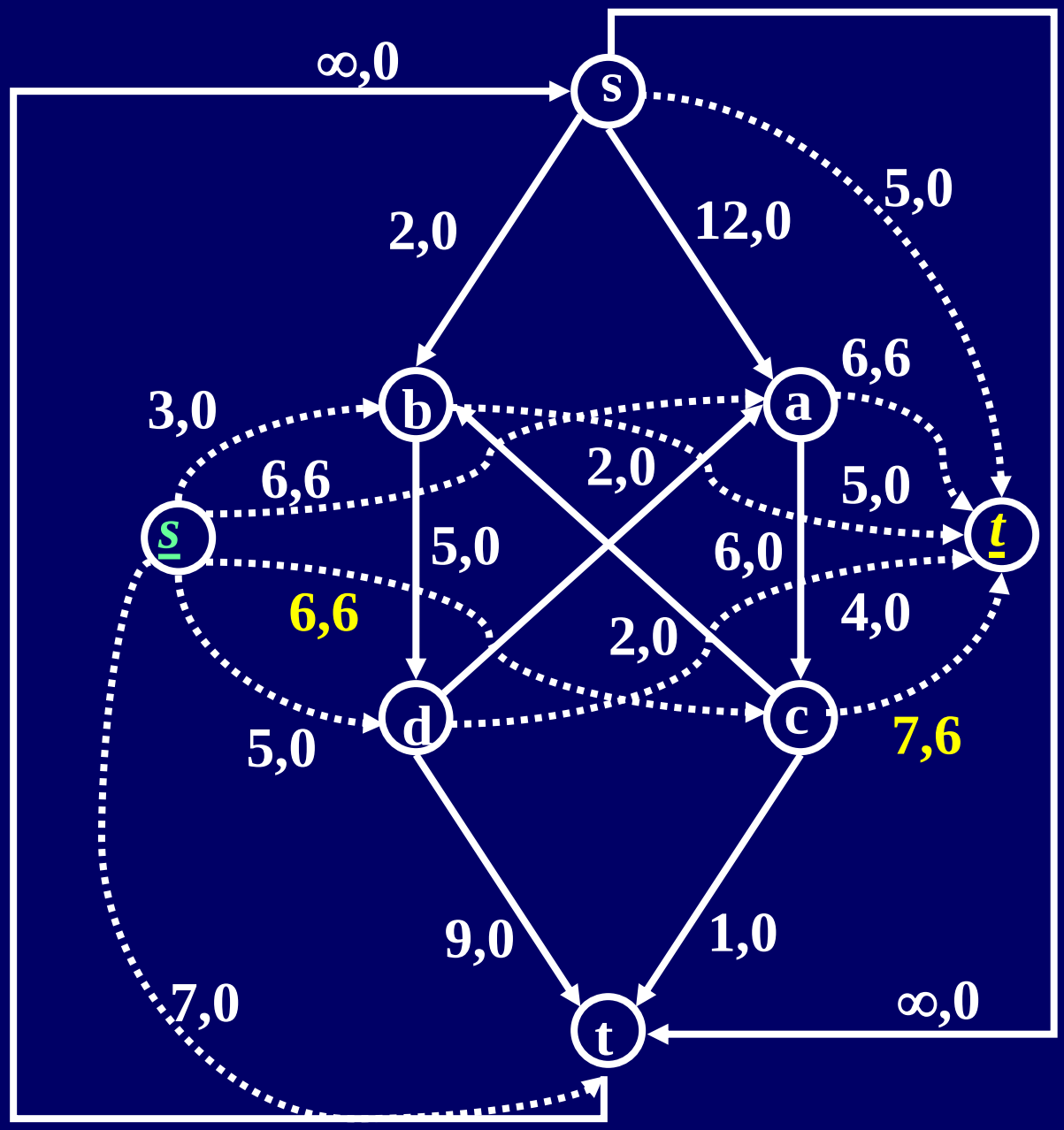
(4)



G_{aux}

L'exemple

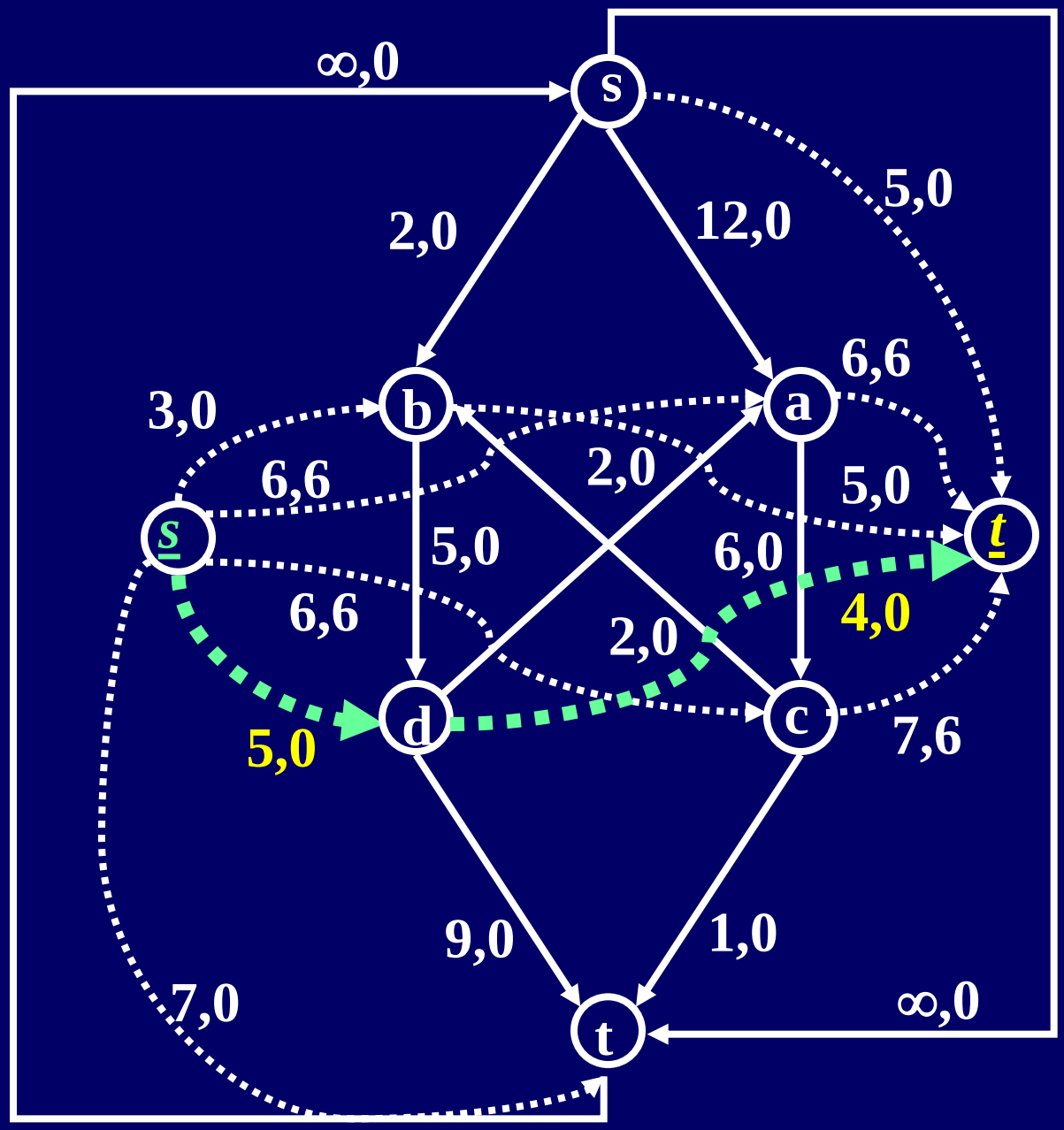
(5)



G_{aux}

L'exemple

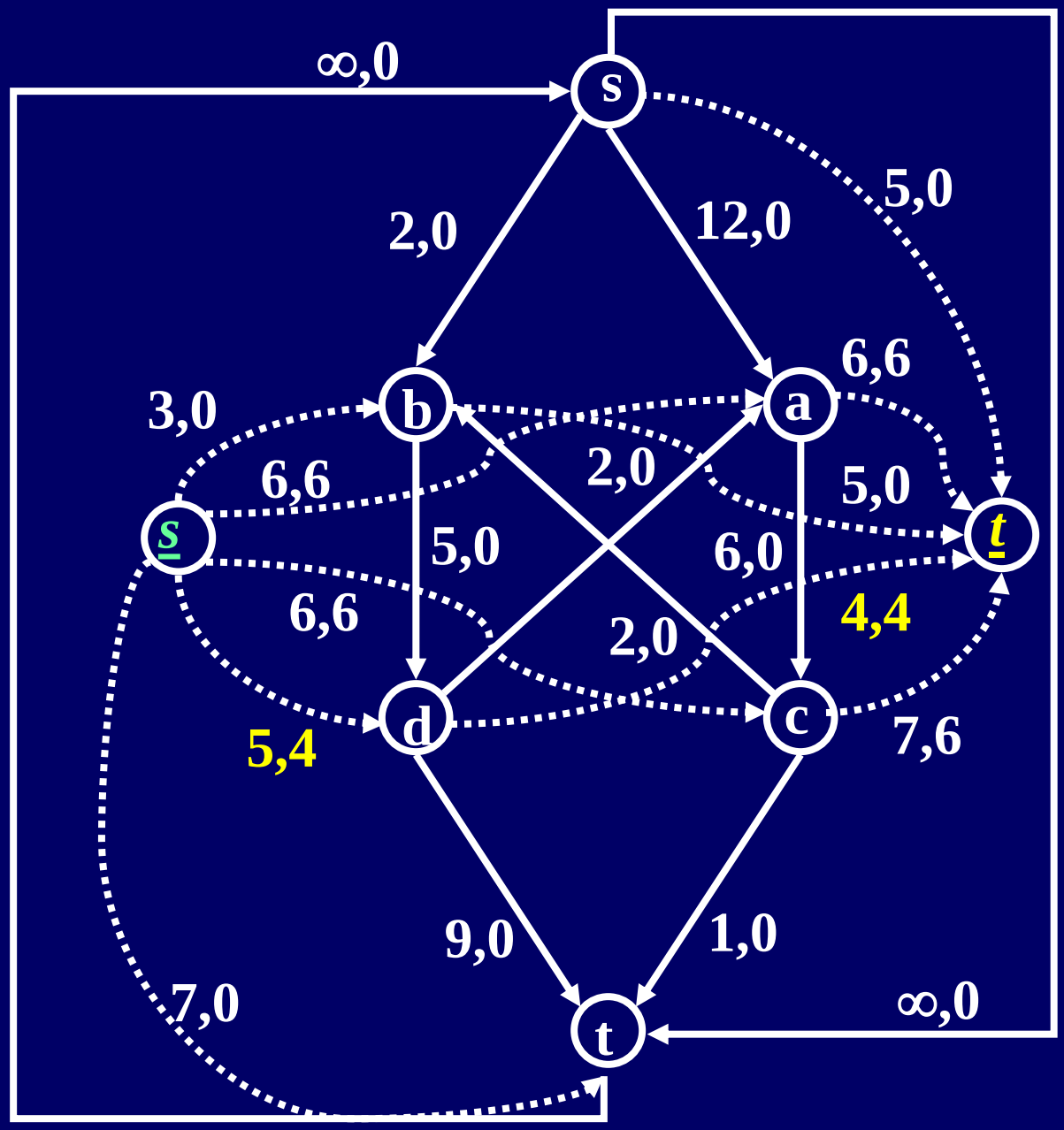
(6)



G_{aux}

L'exemple

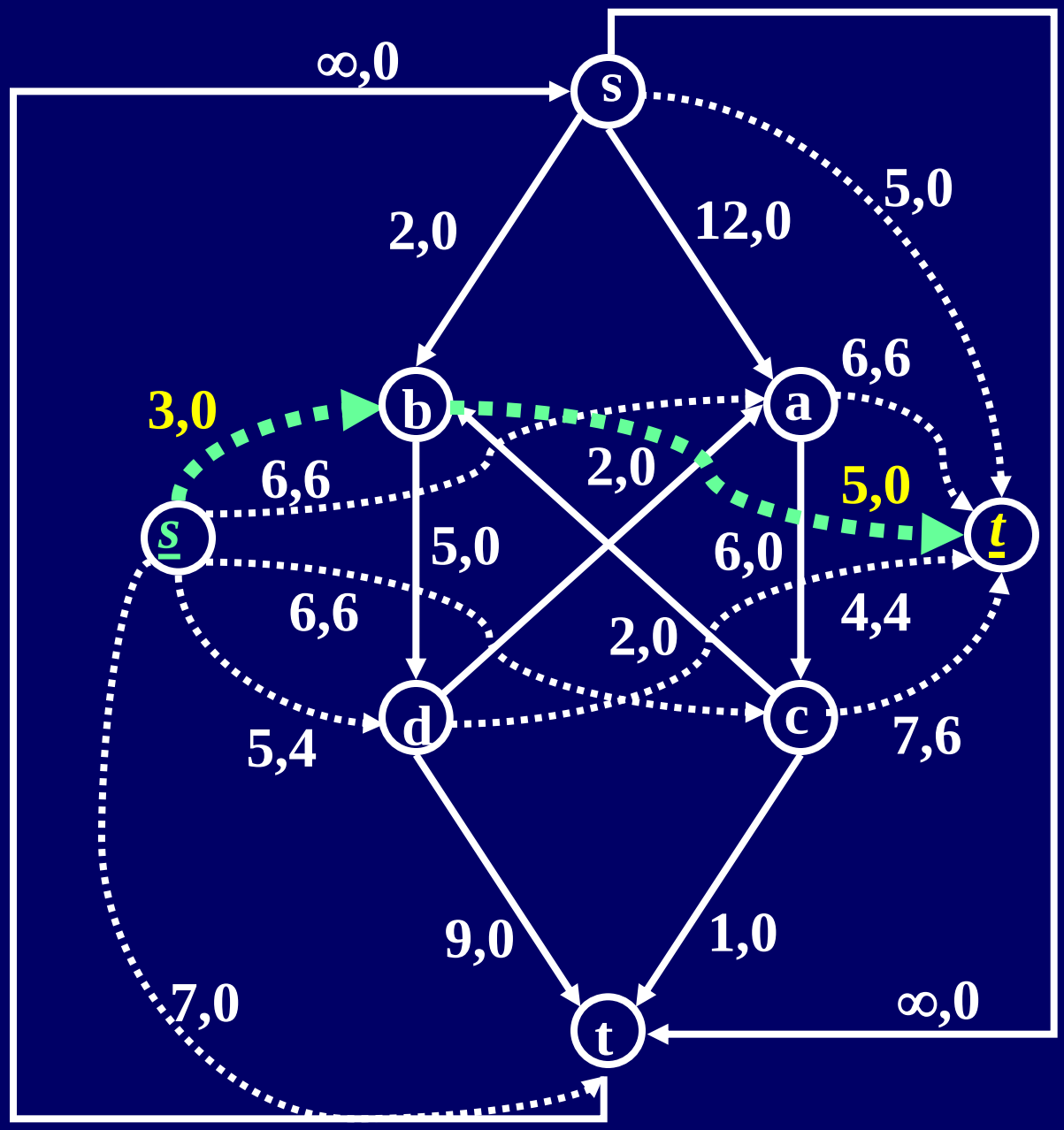
(7)



G_{aux}

L'exemple

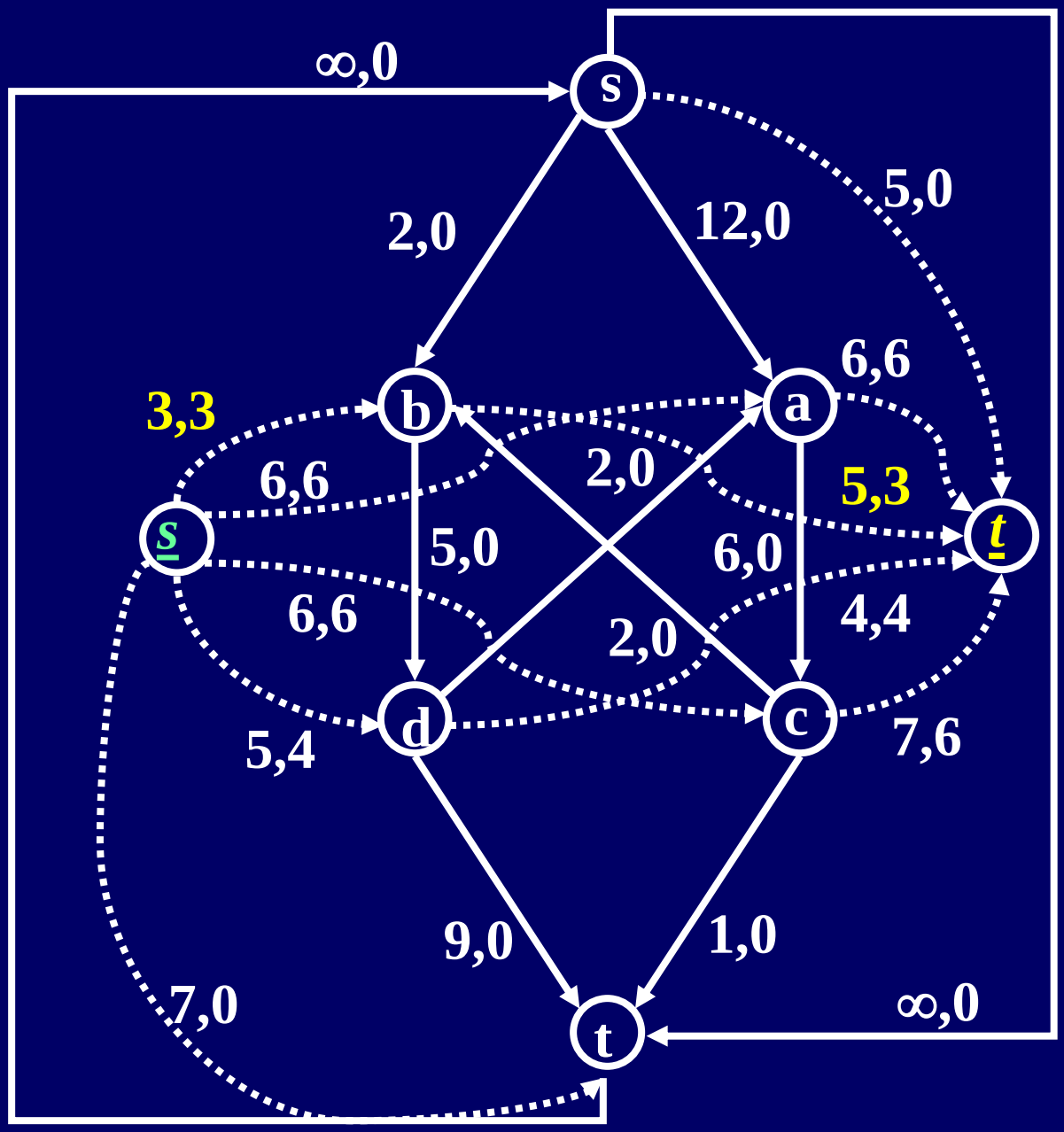
(8)



G_{aux}

L'exemple

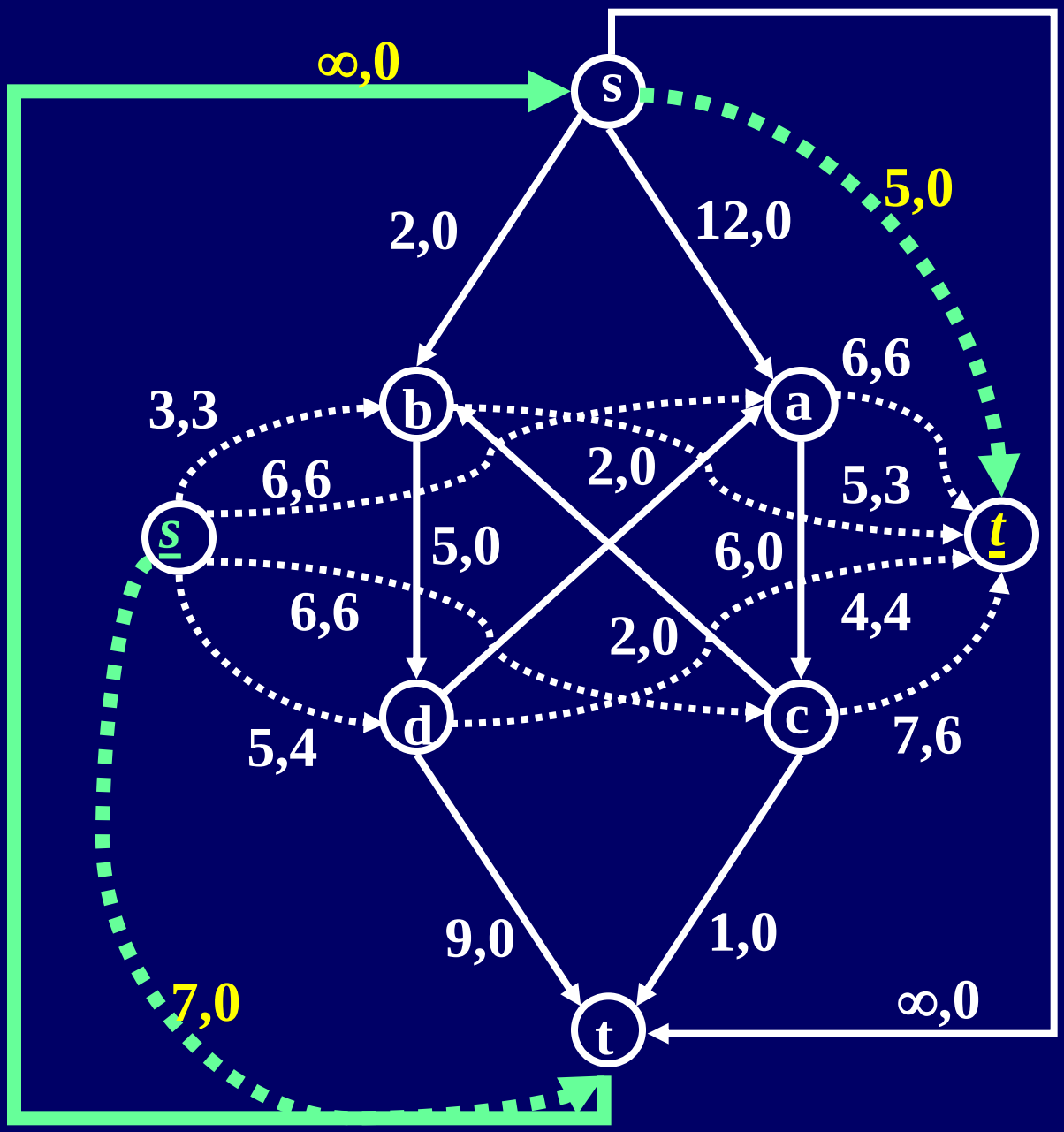
(8)



G_{aux}

L'exemple

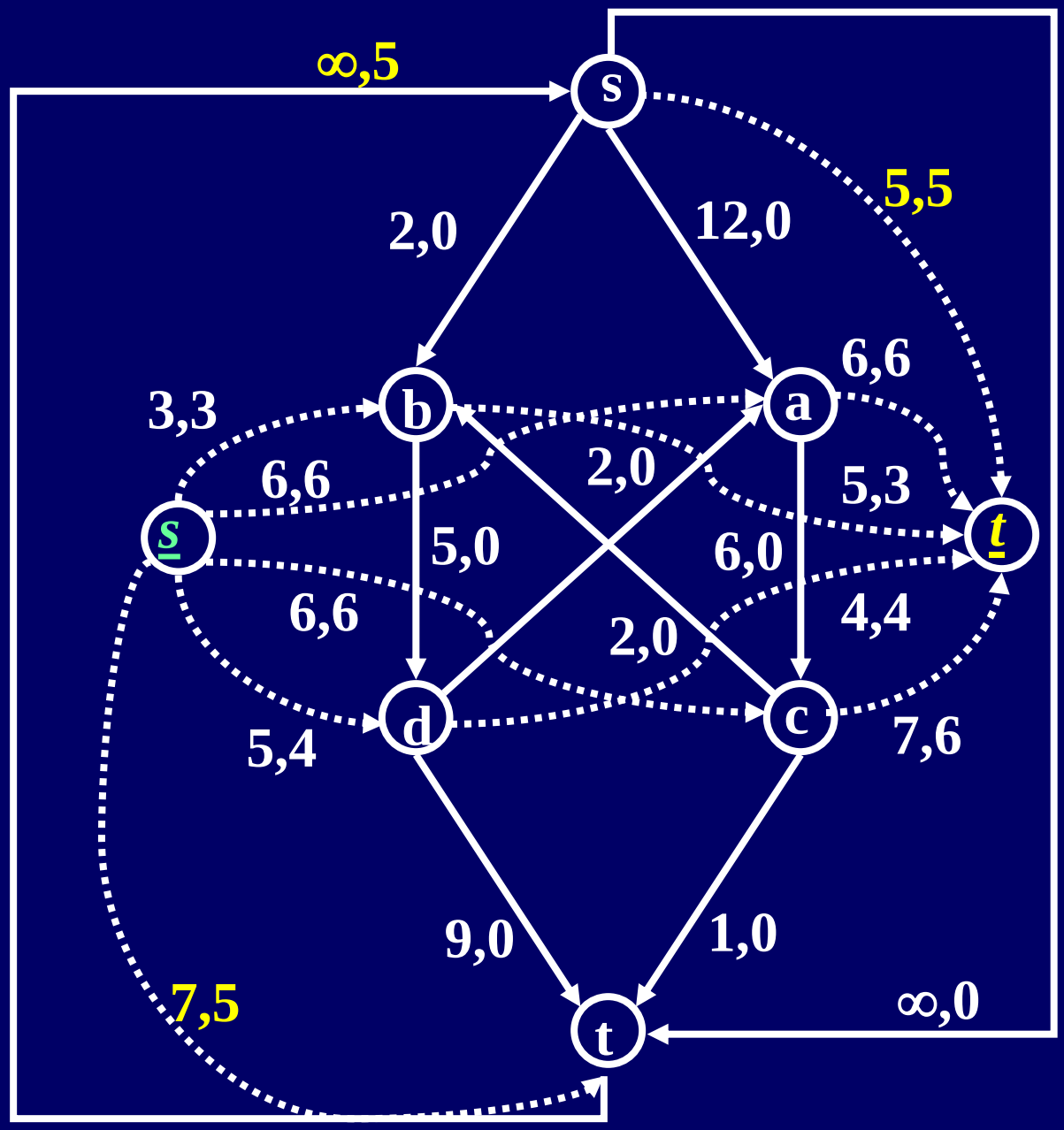
(9)



G_{aux}

L'exemple

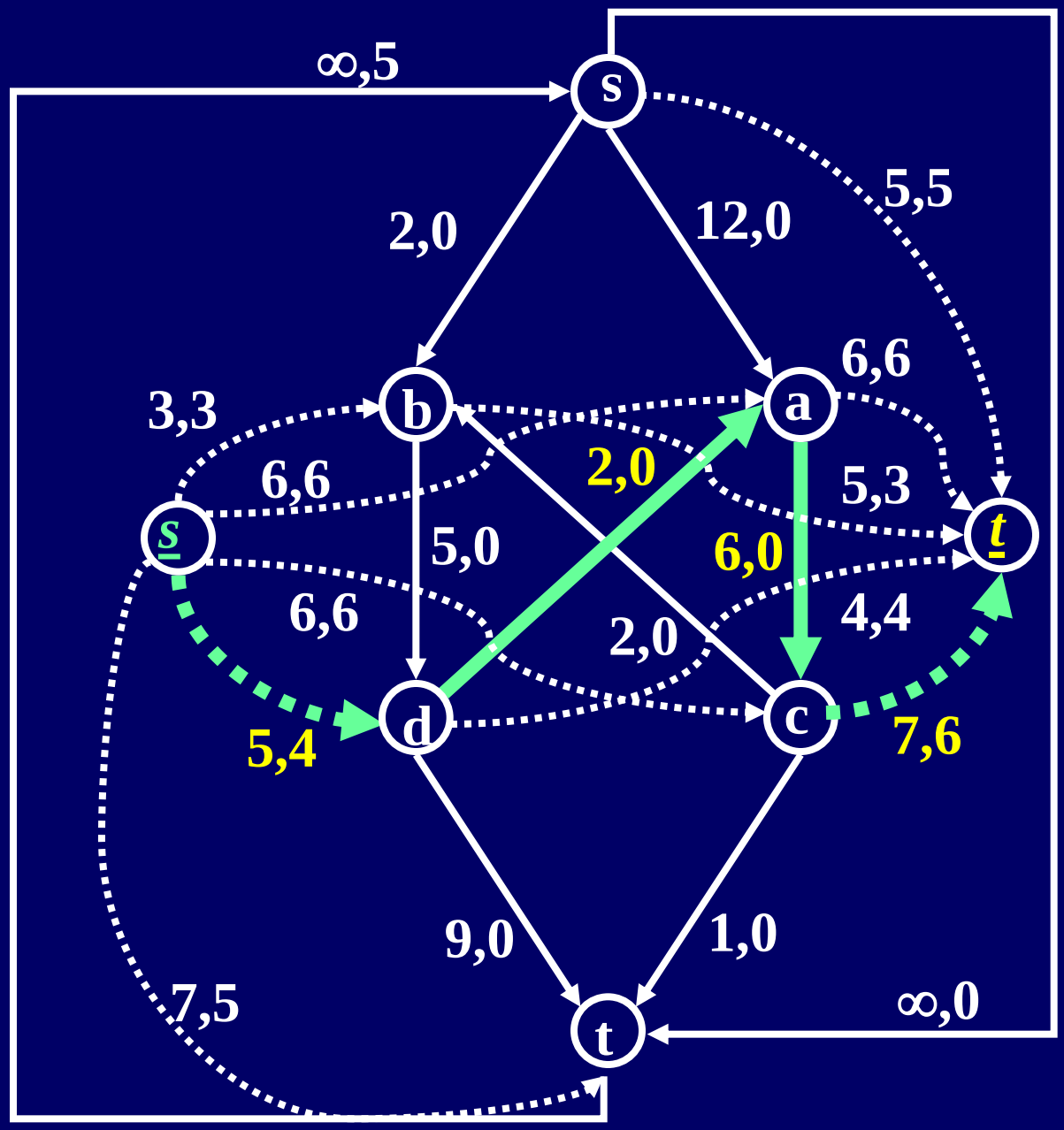
(10)



G_{aux}

L'exemple

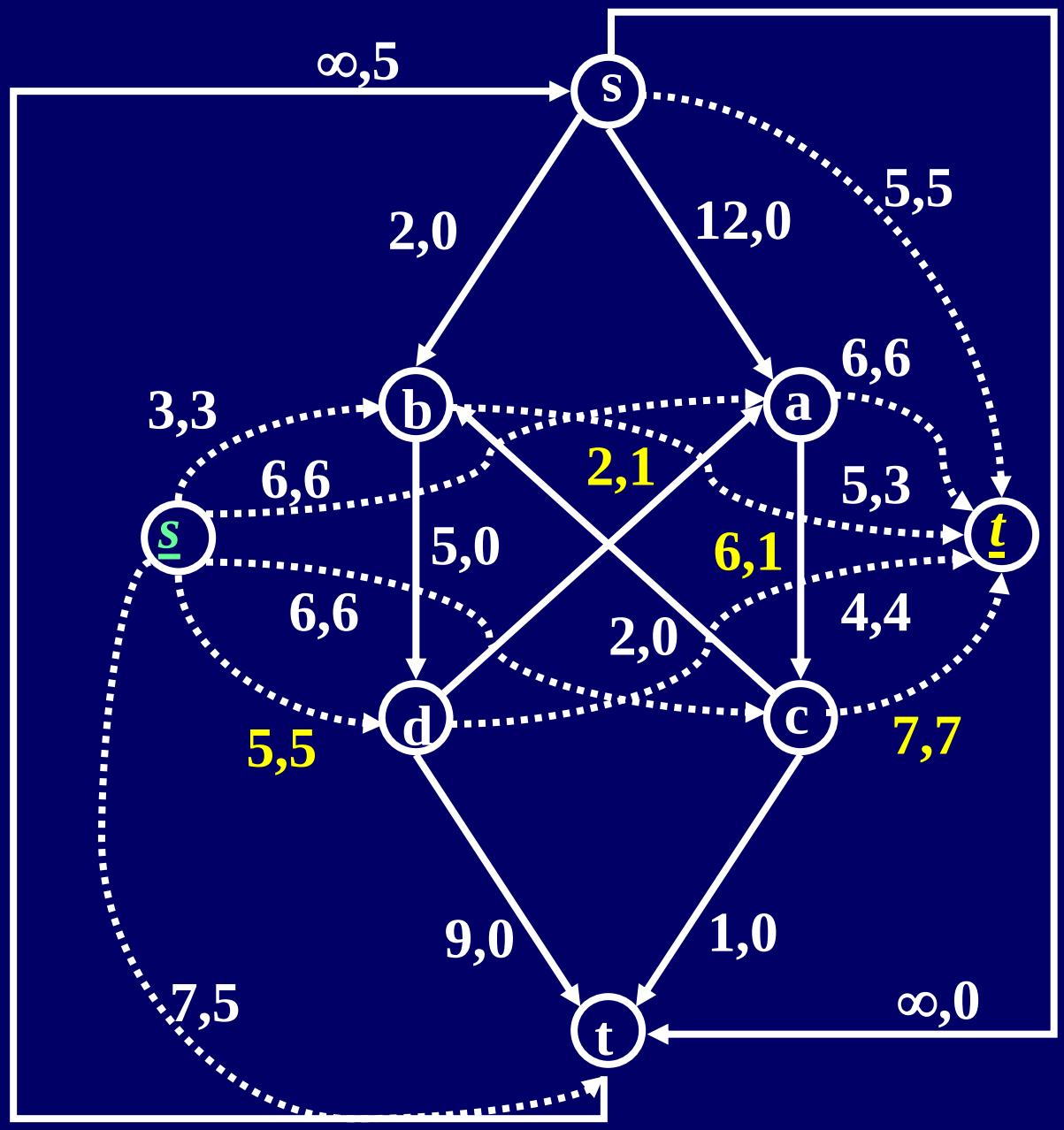
(11)



G_{aux}

L'exemple

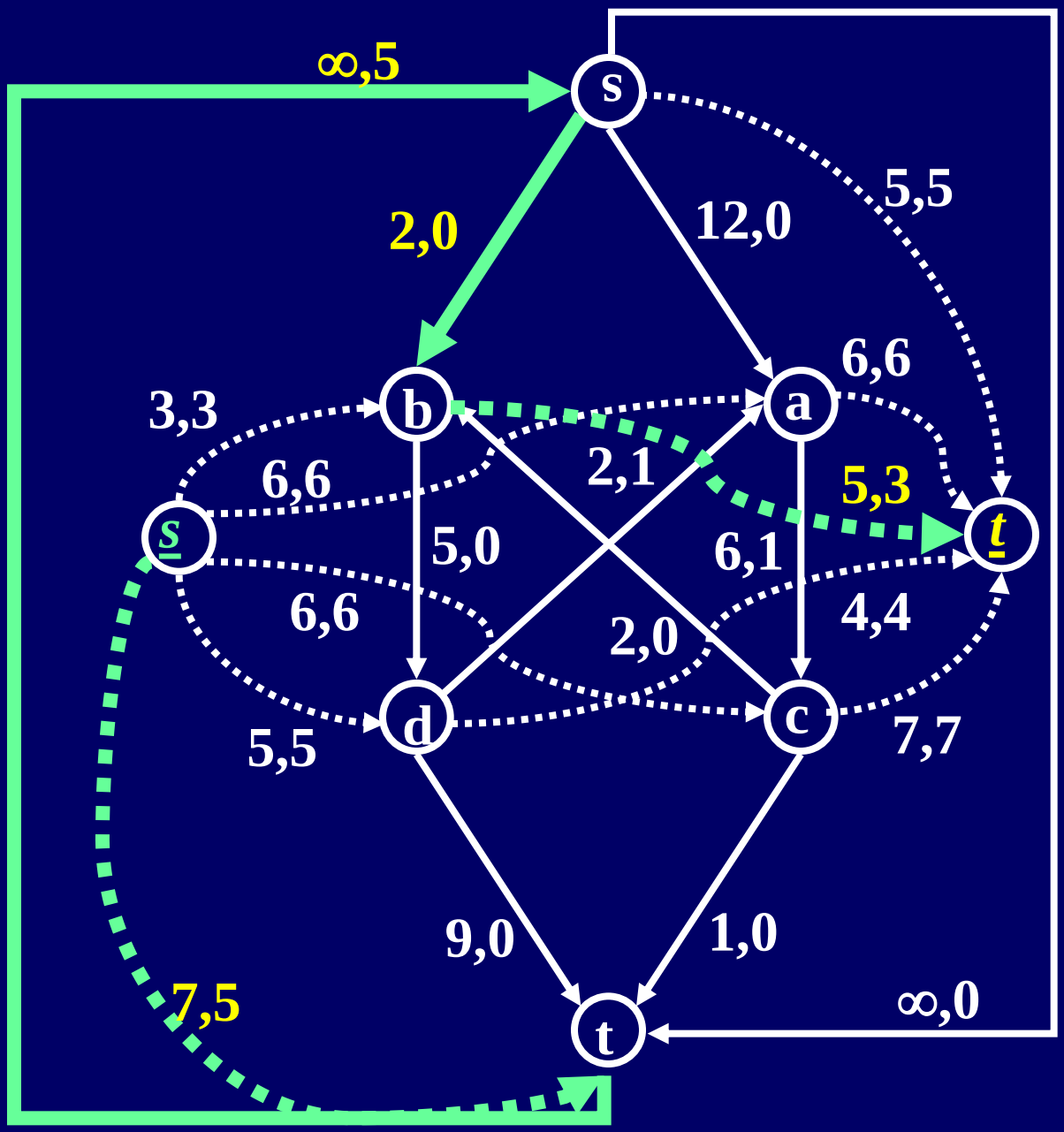
(12)



G_{aux}

L'exemple

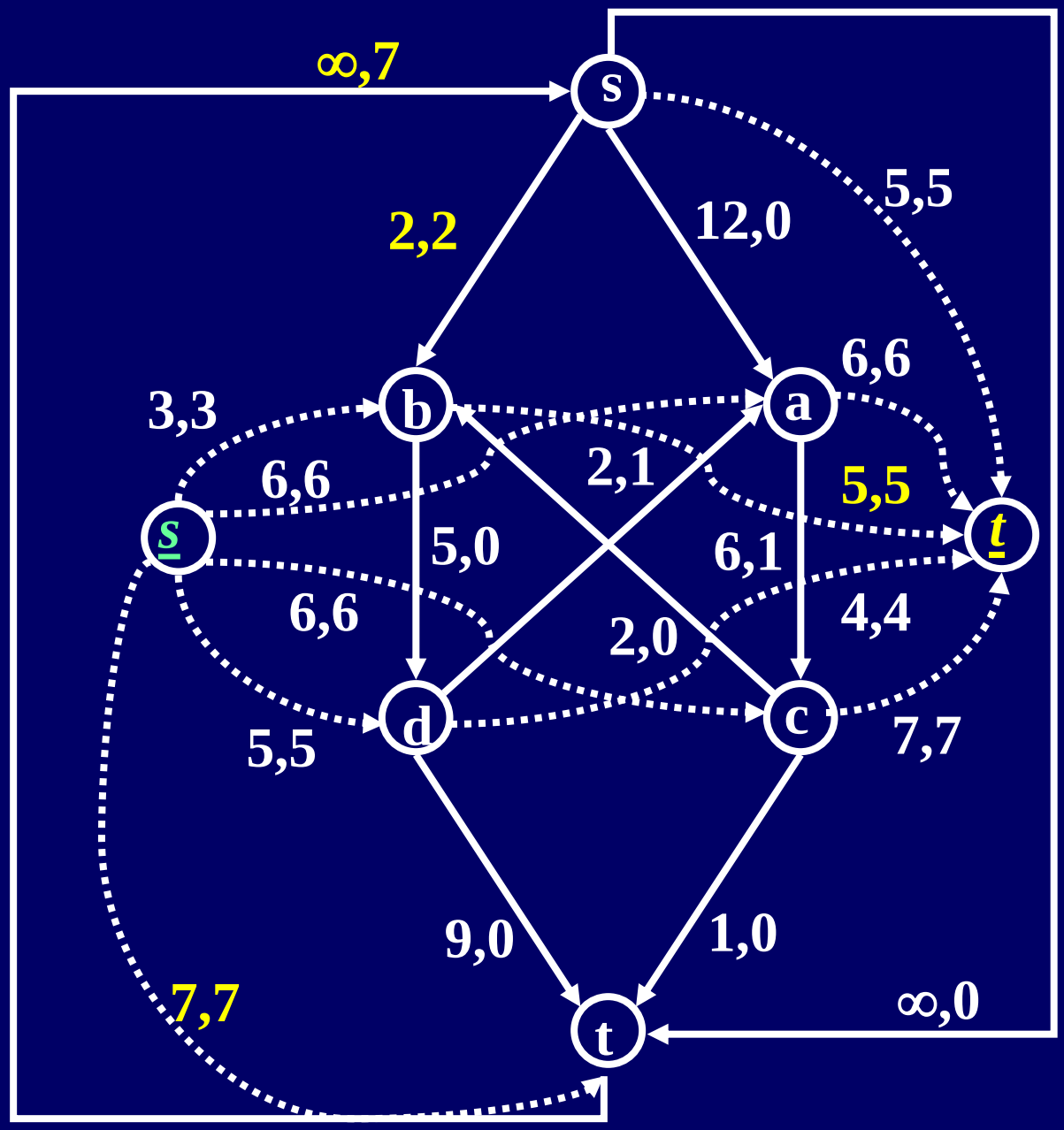
(13)



G_{aux}

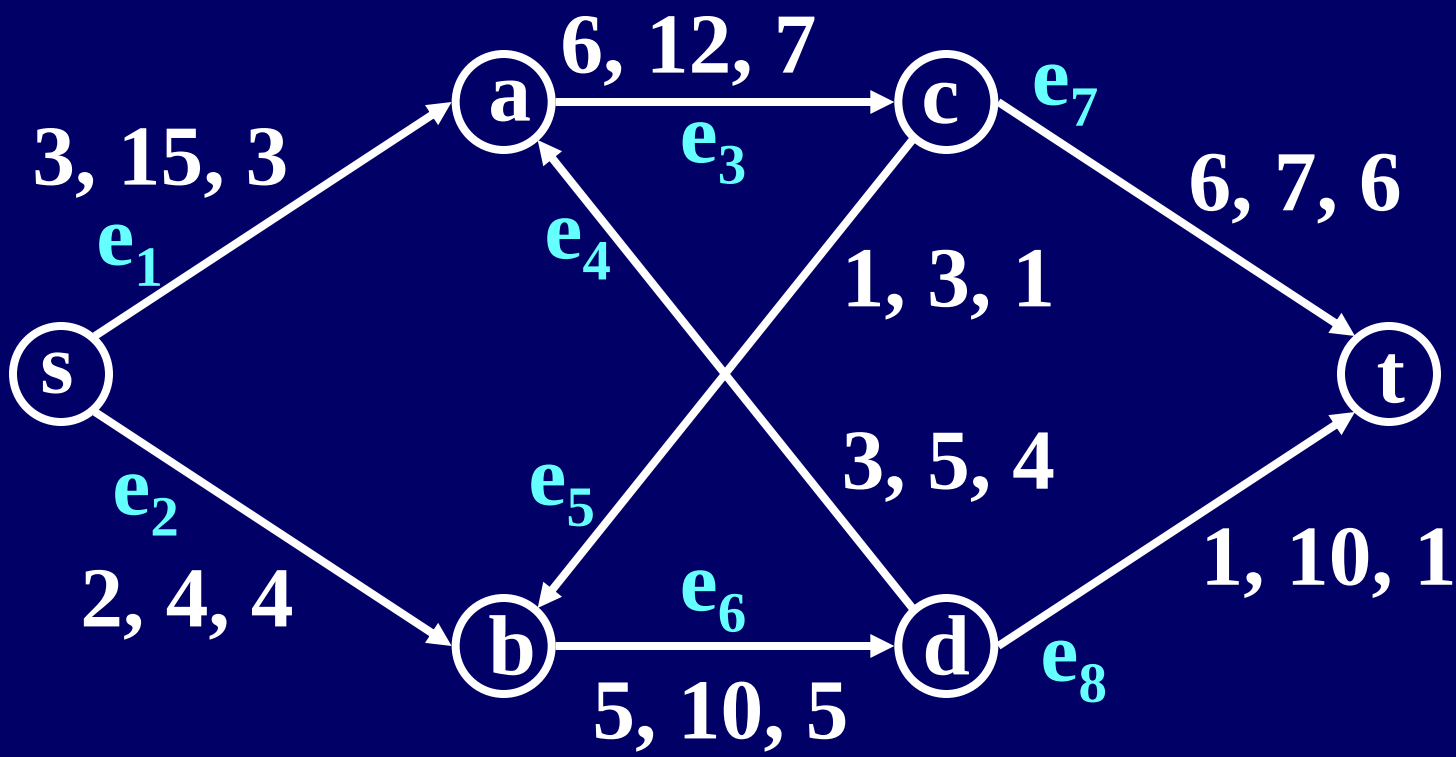
L'exemple

(14)



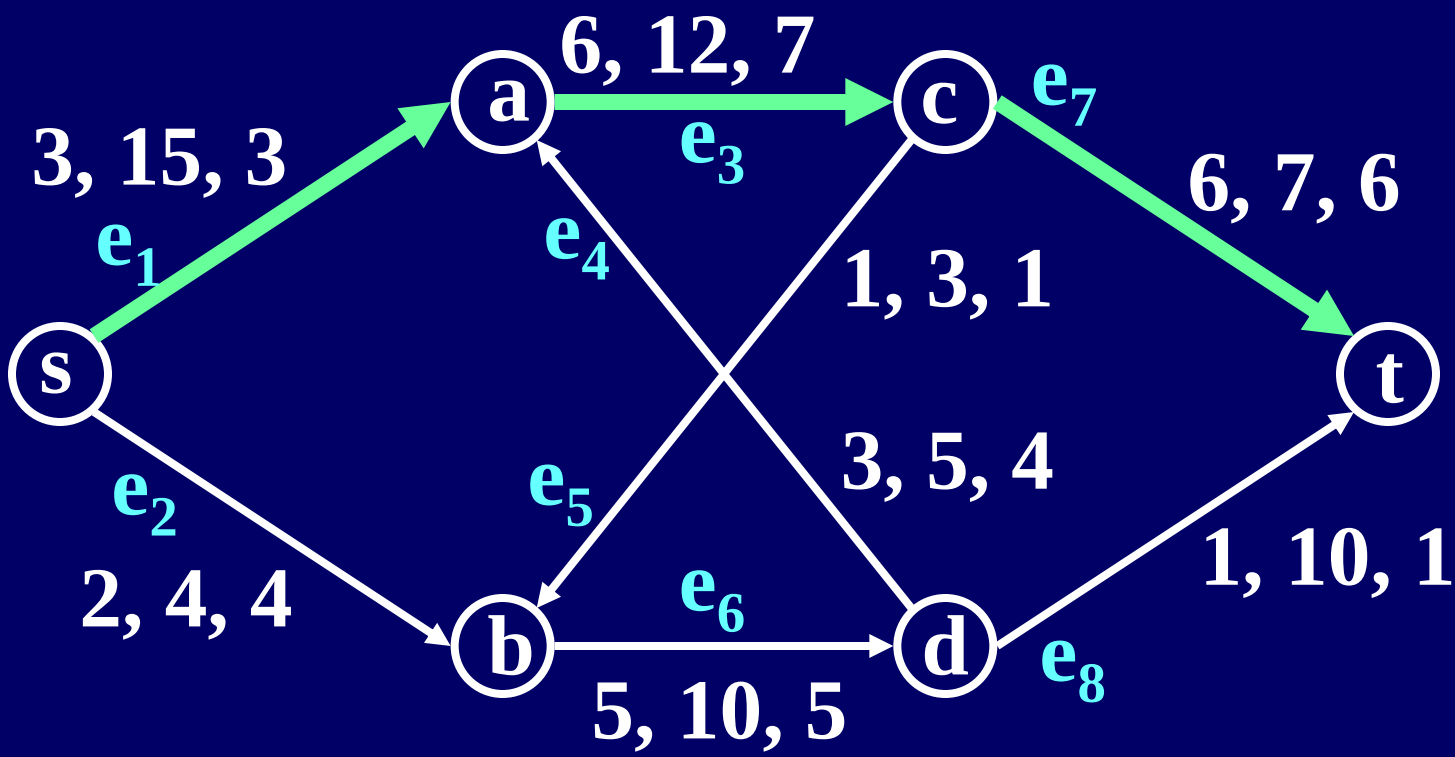
G_{aux}

Et le flot admissible



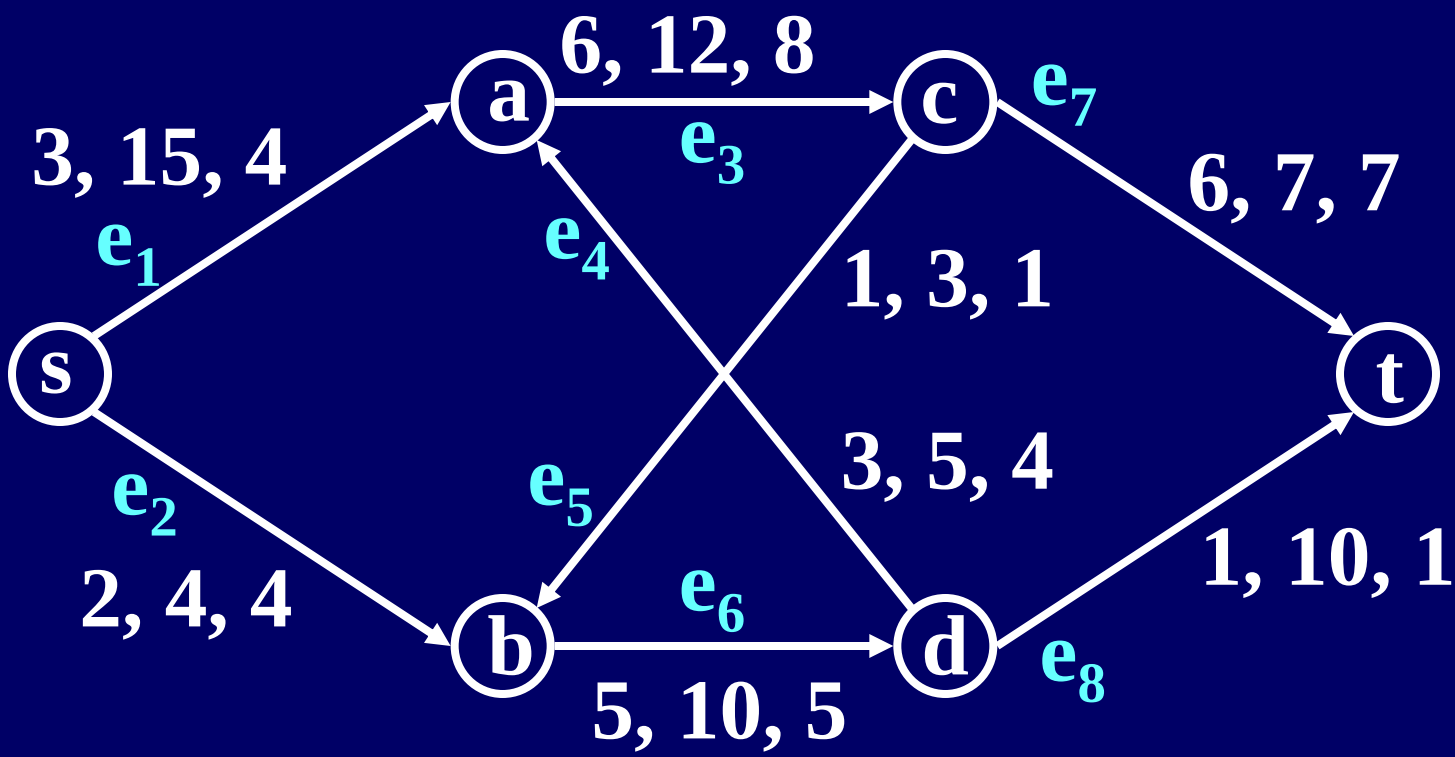
b, c, f

Une augmentation



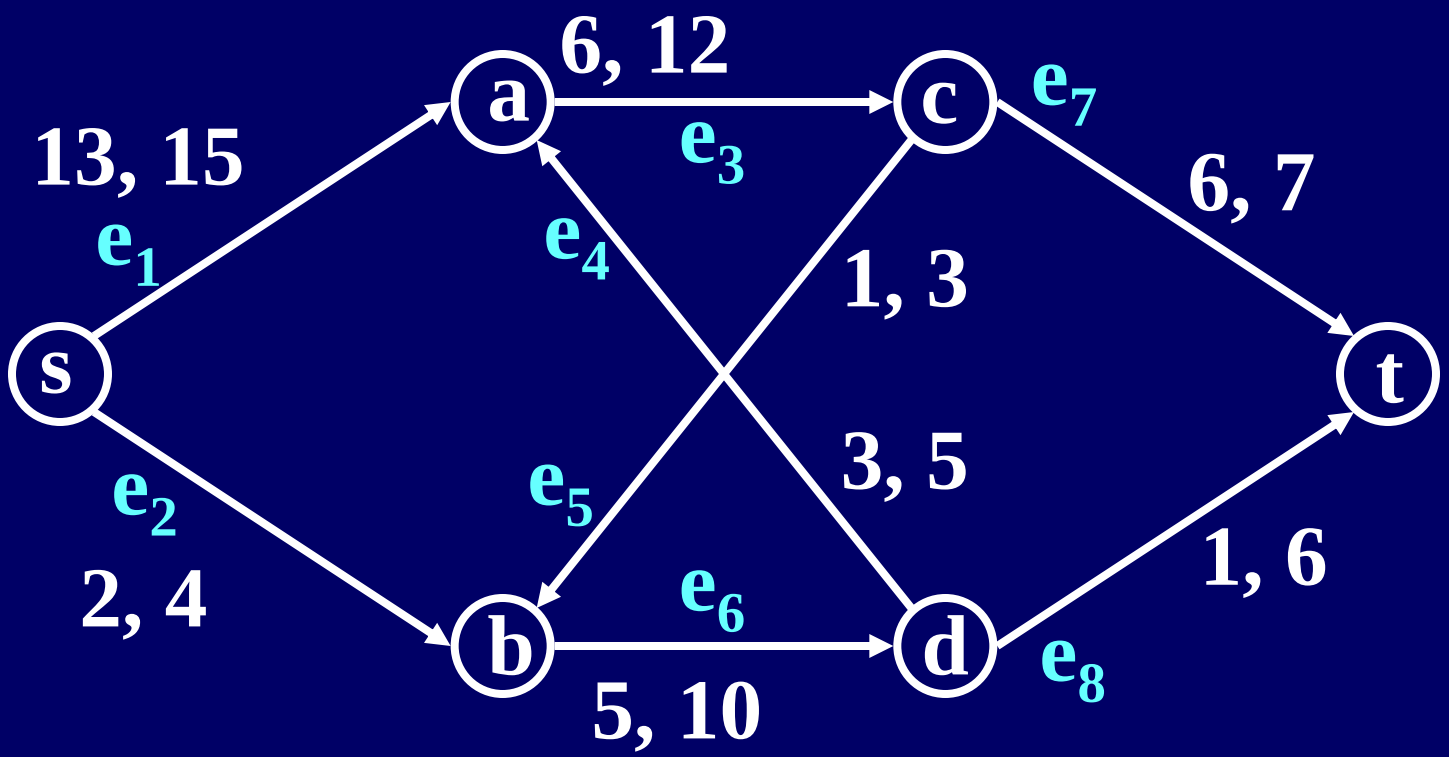
b, c, f

Et nous voici au point de départ



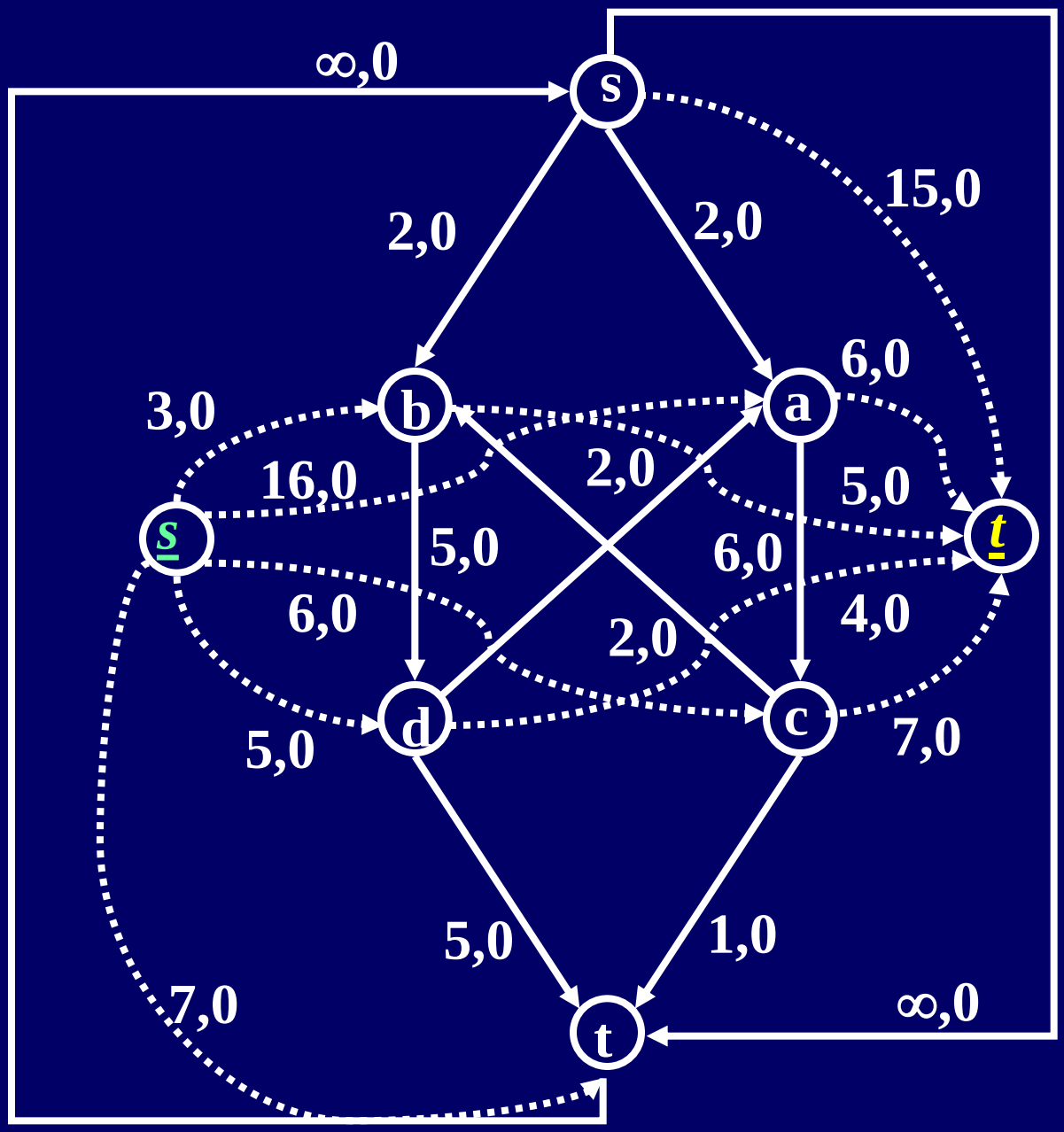
b, c, f

Un cas négative



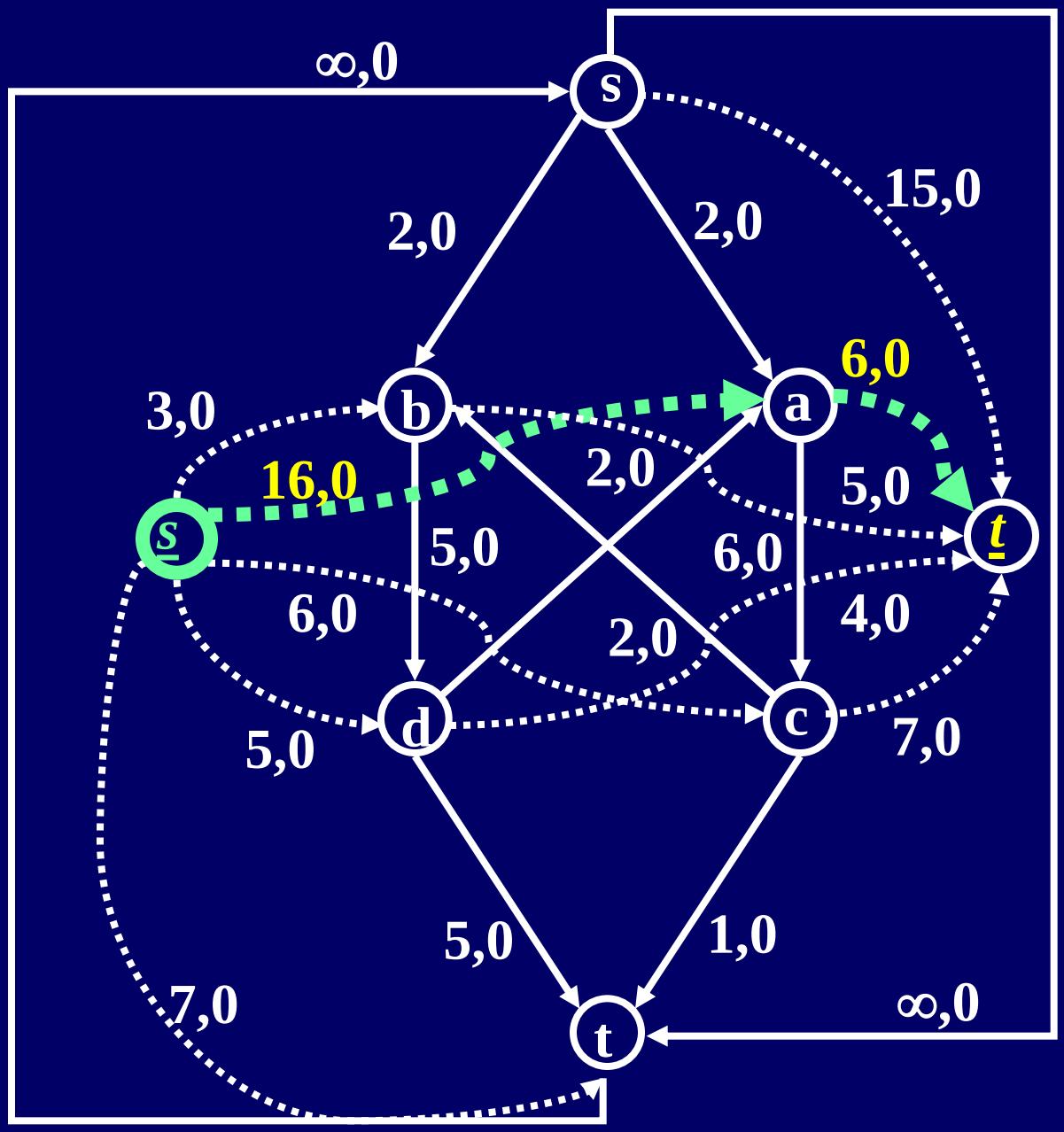
b, c

G_{aux}



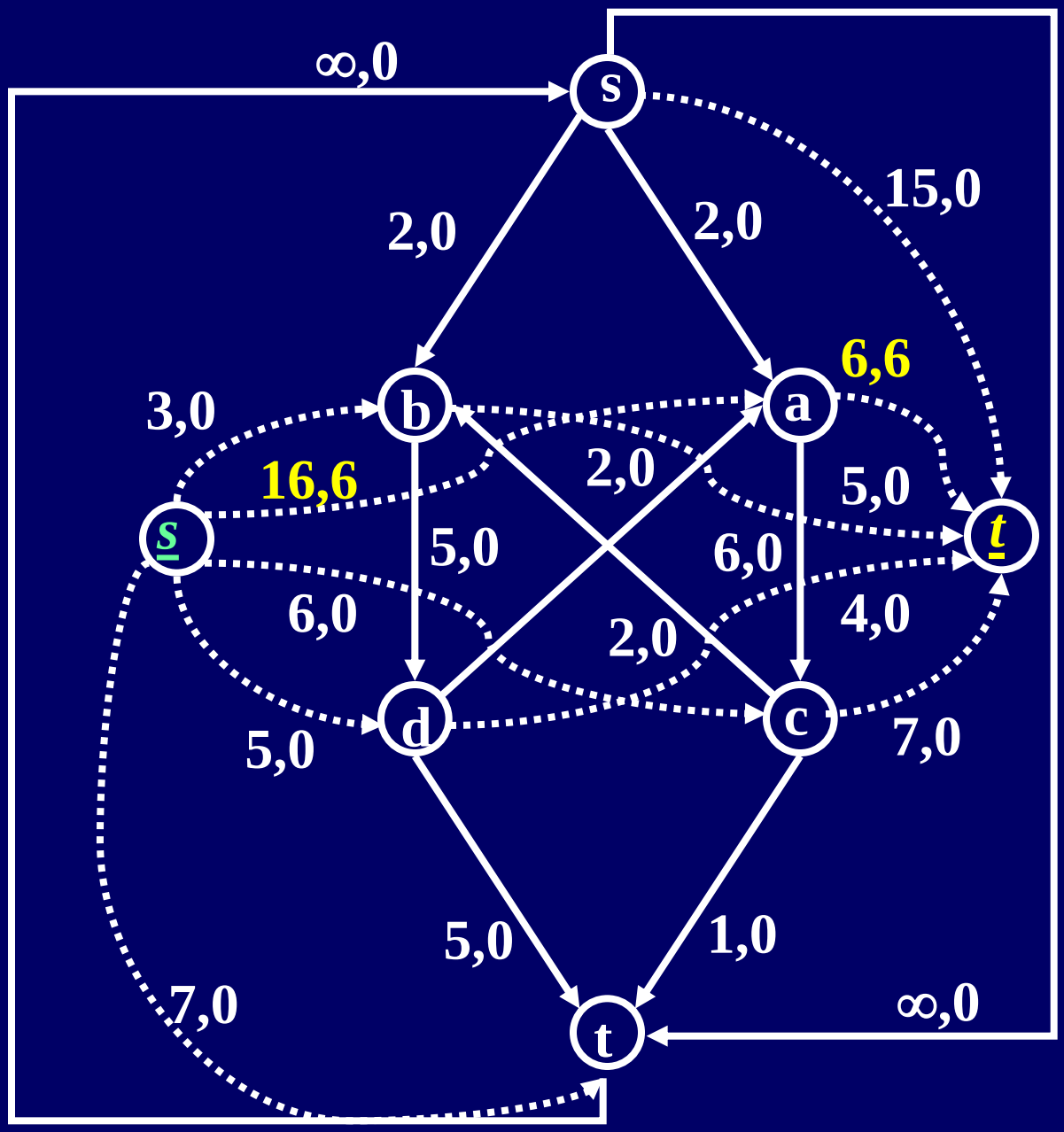
G_{aux}

G_{aux}



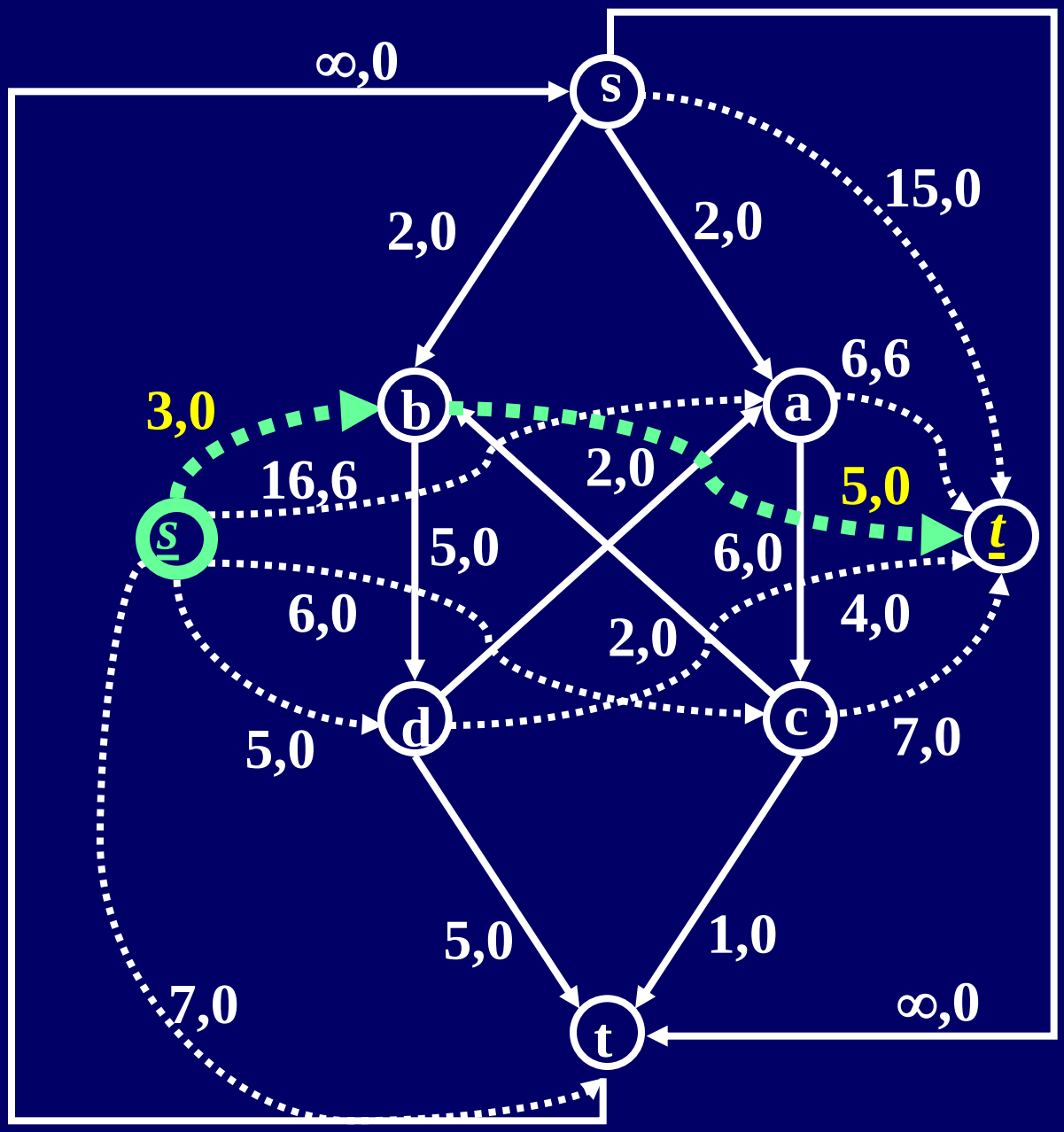
G_{aux}

G_{aux}



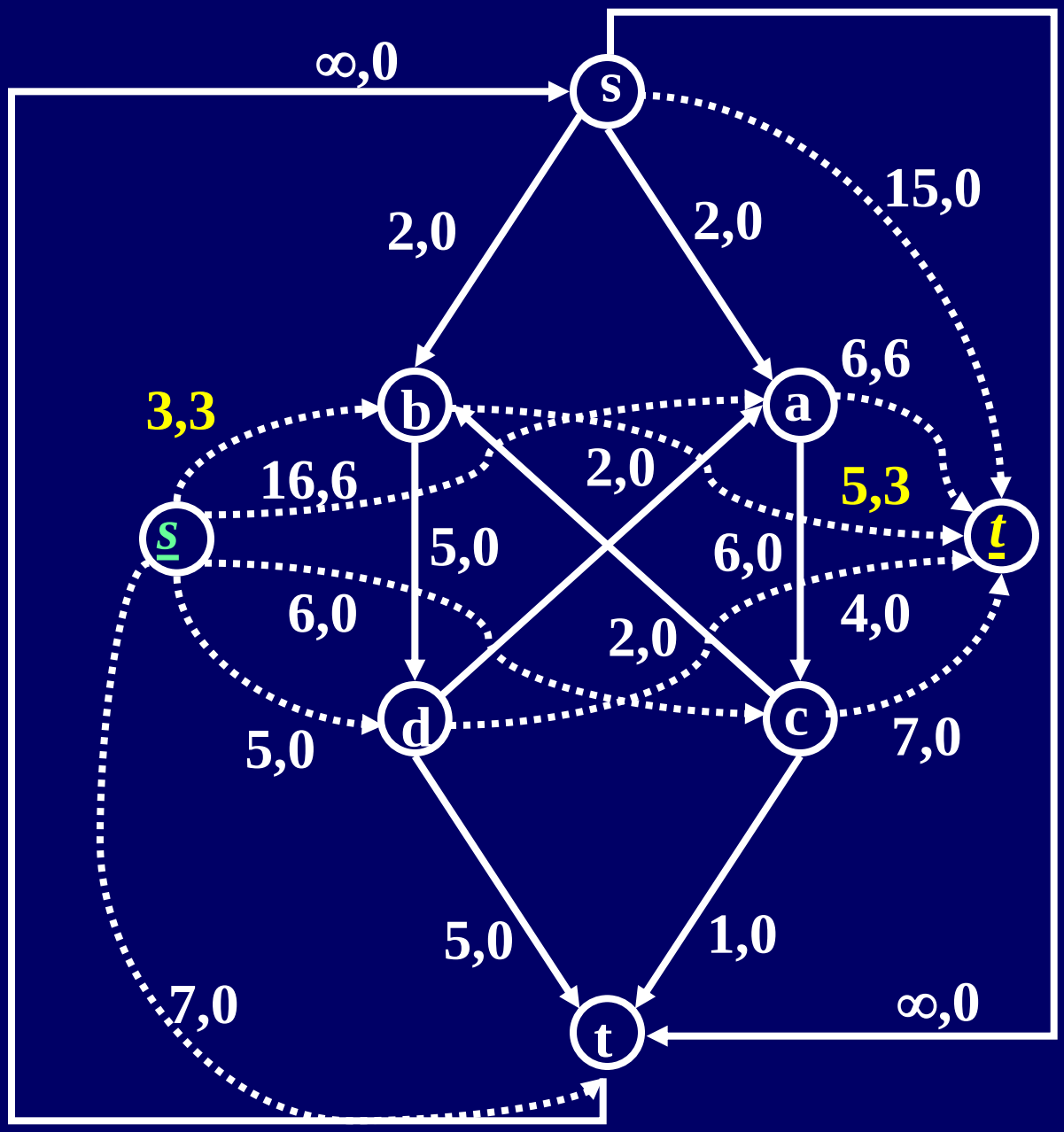
G_{aux}

G_{aux}



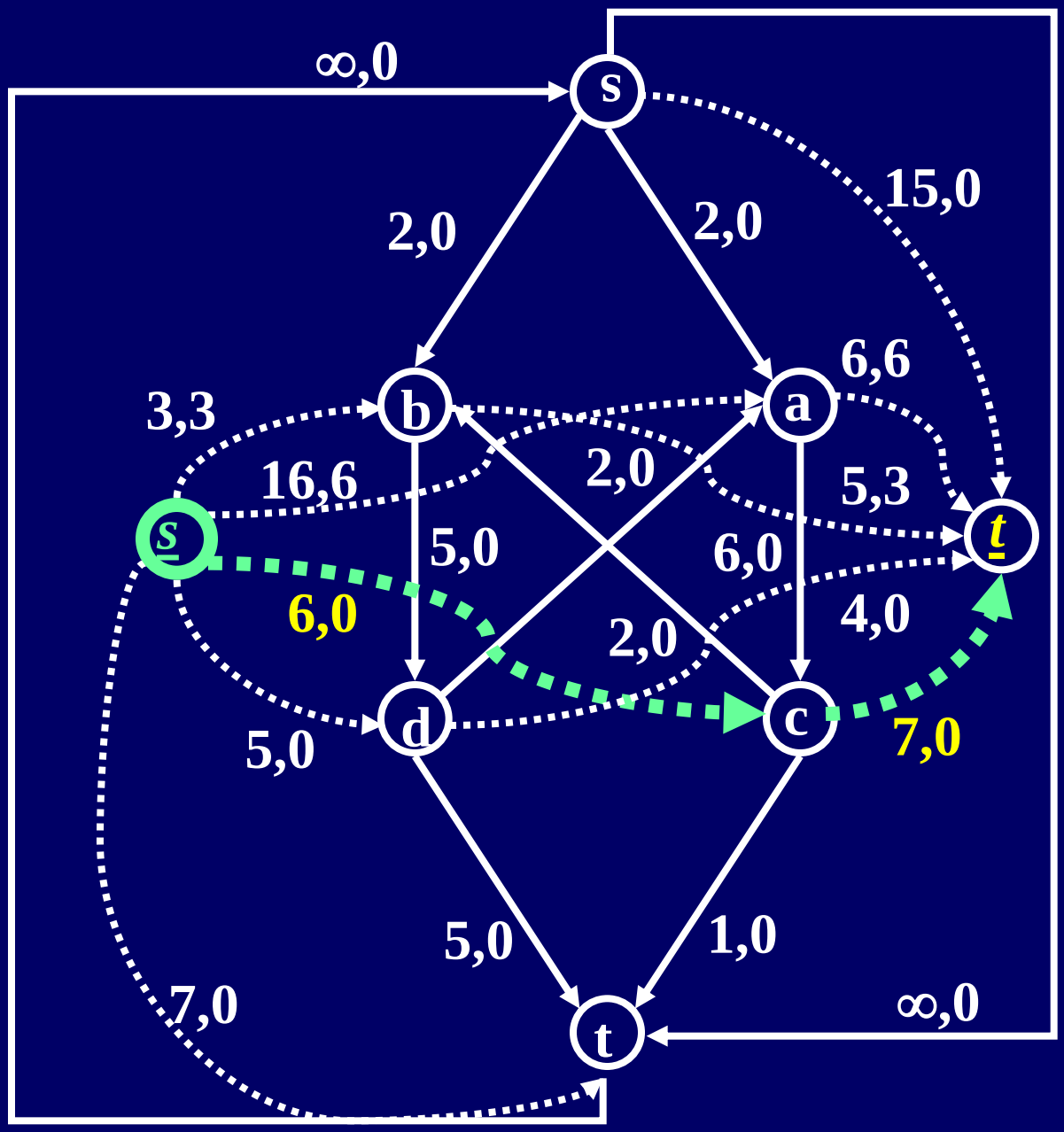
G_{aux}

G_{aux}



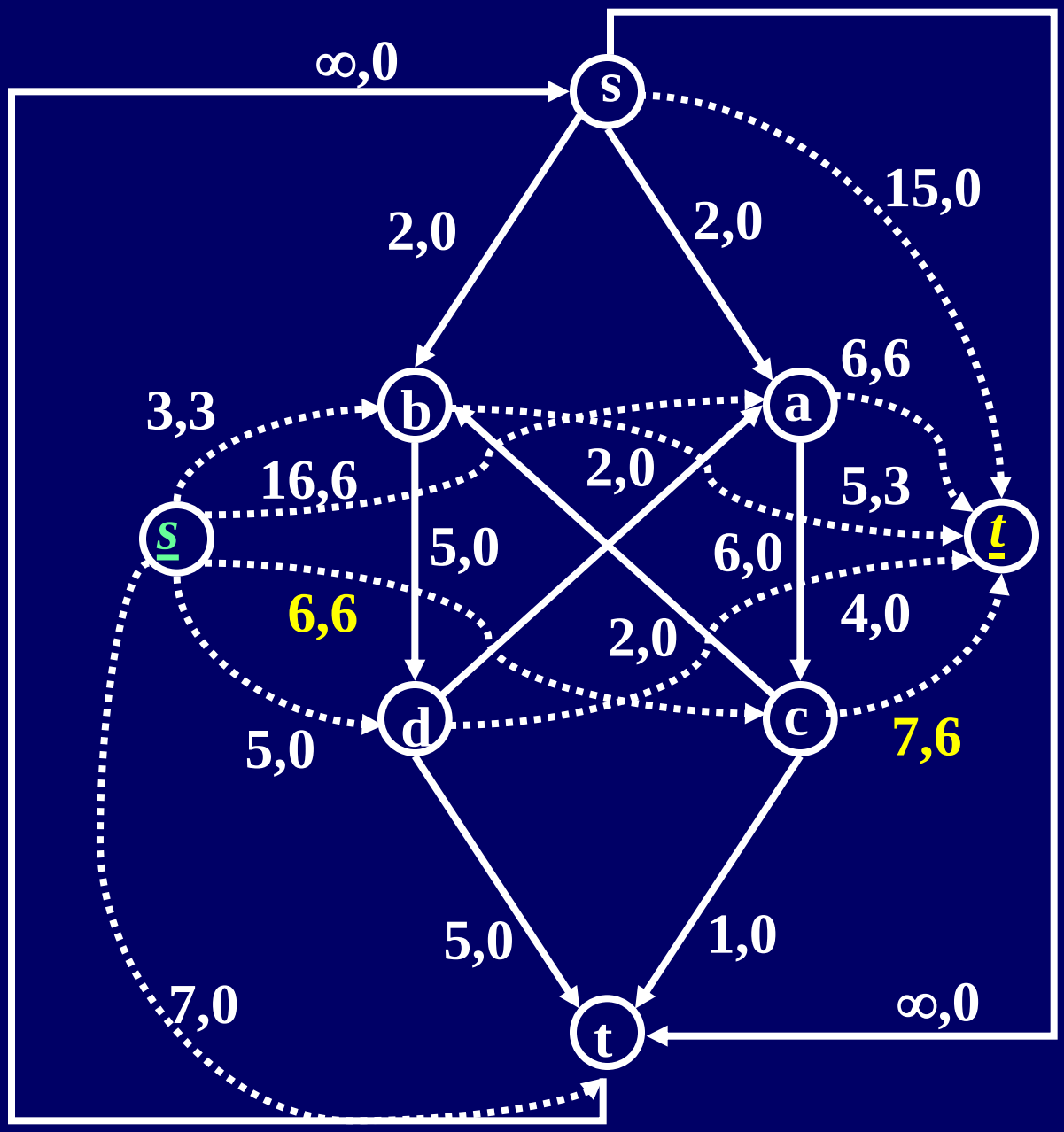
G_{aux}

G_{aux}



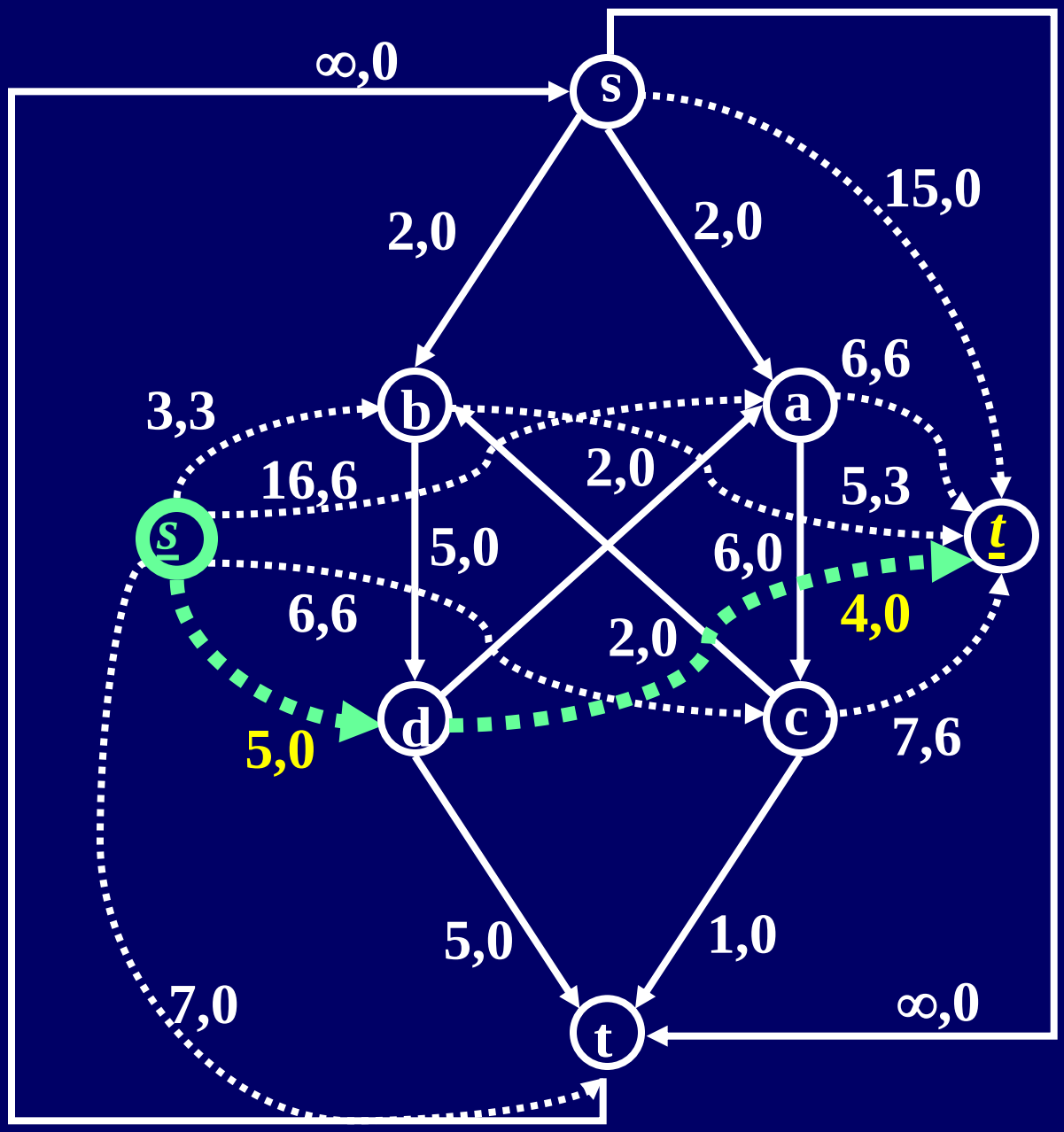
G_{aux}

G_{aux}



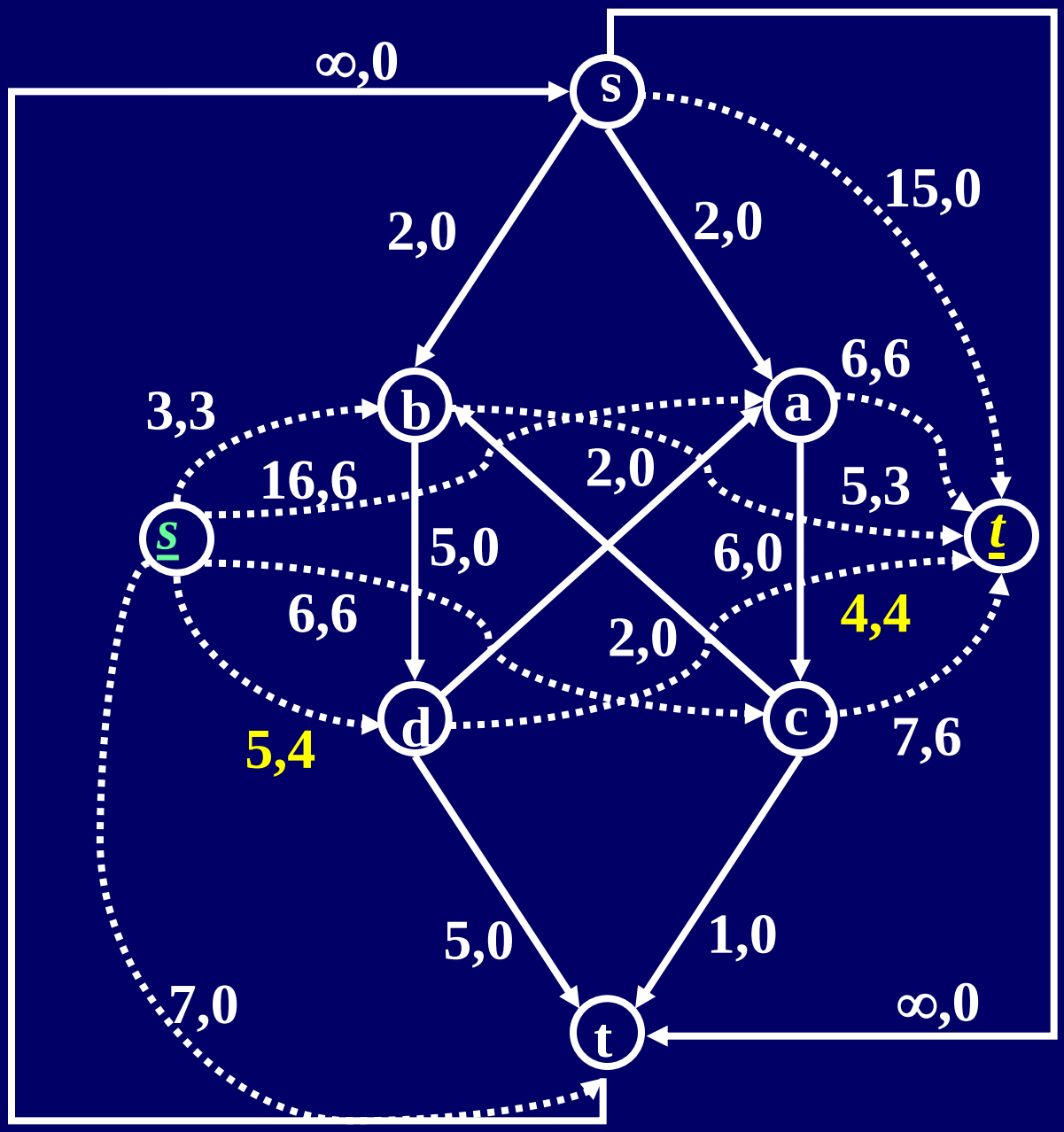
G_{aux}

G_{aux}



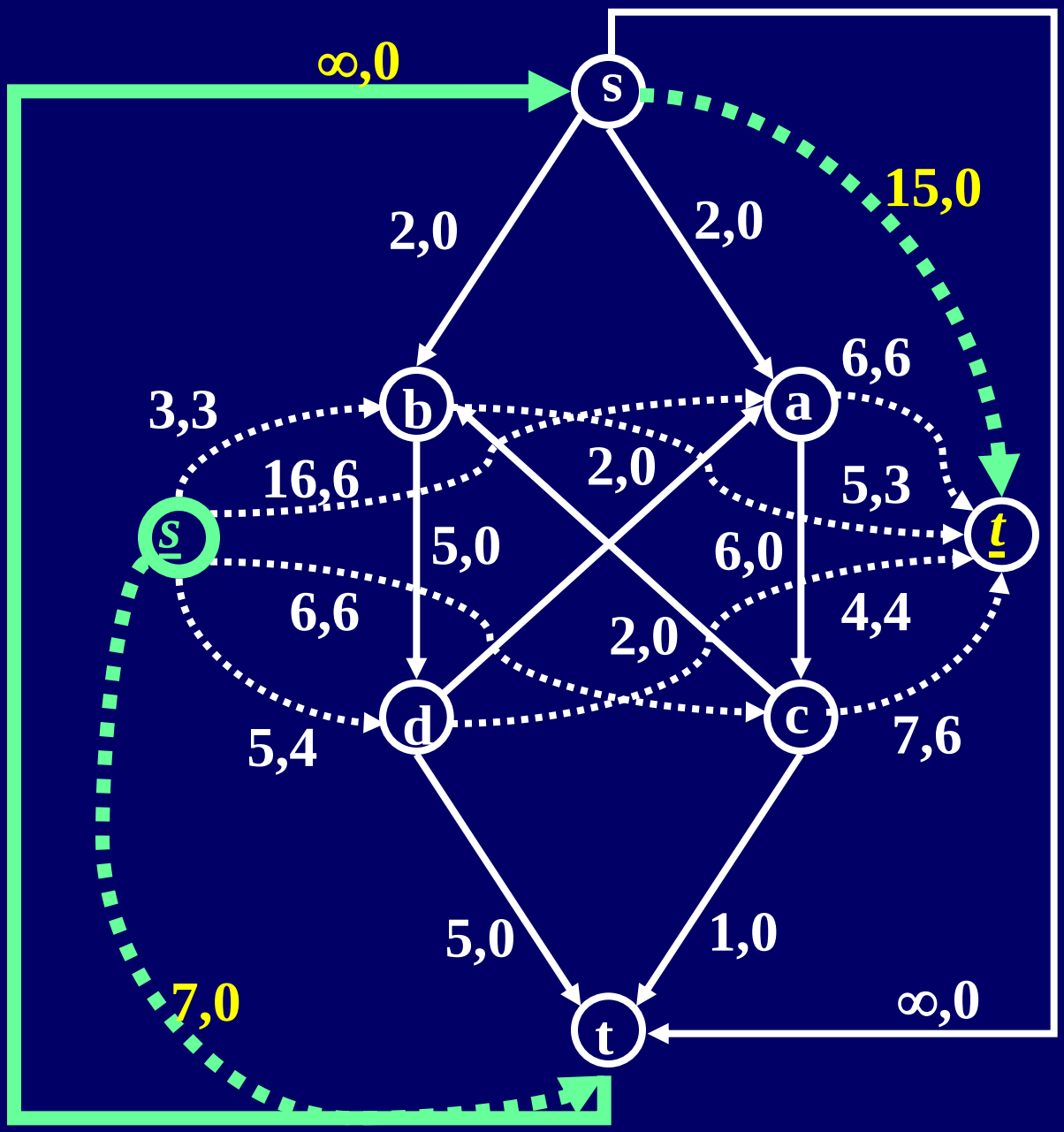
G_{aux}

G_{aux}



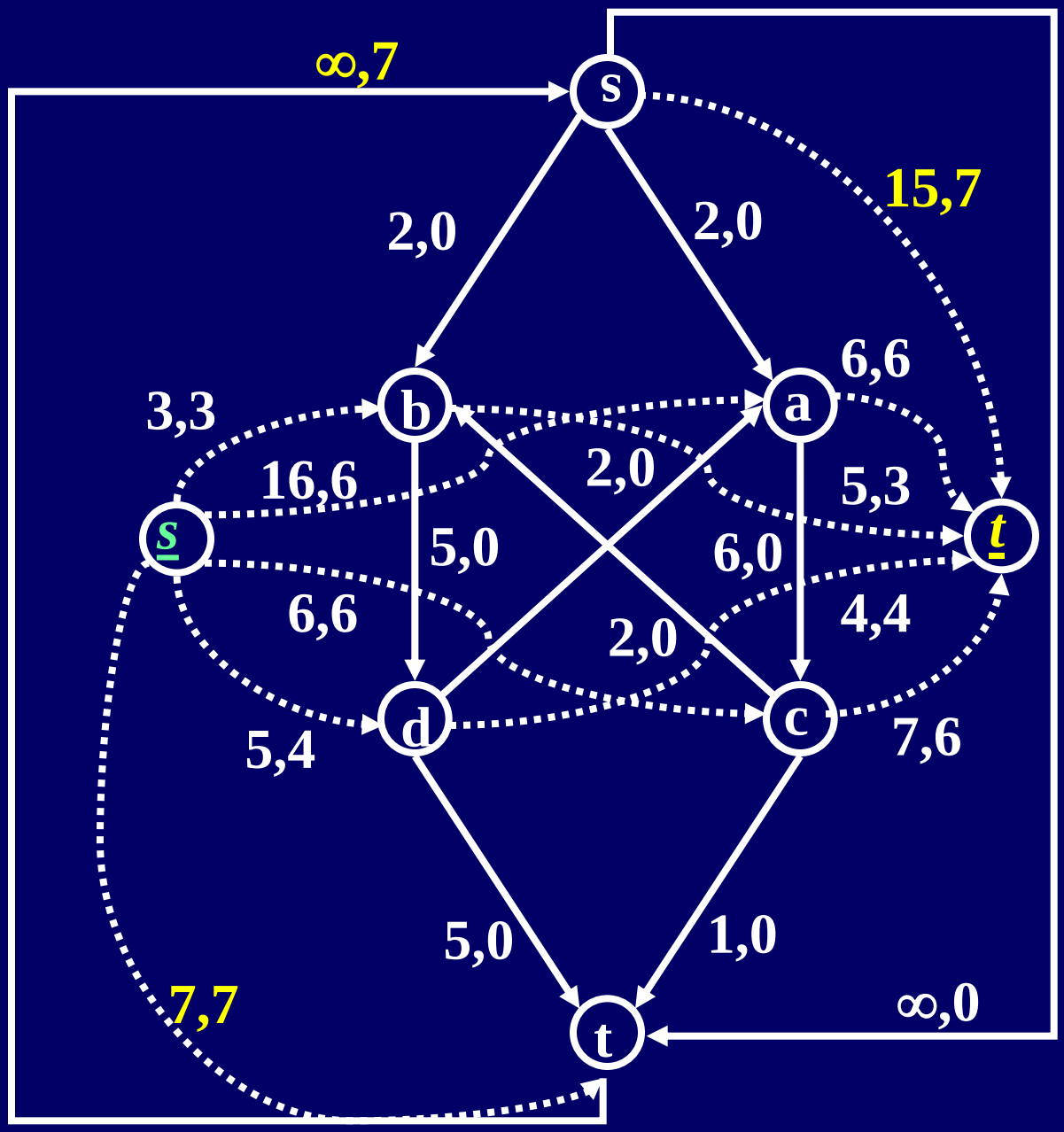
G_{aux}

G_{aux}



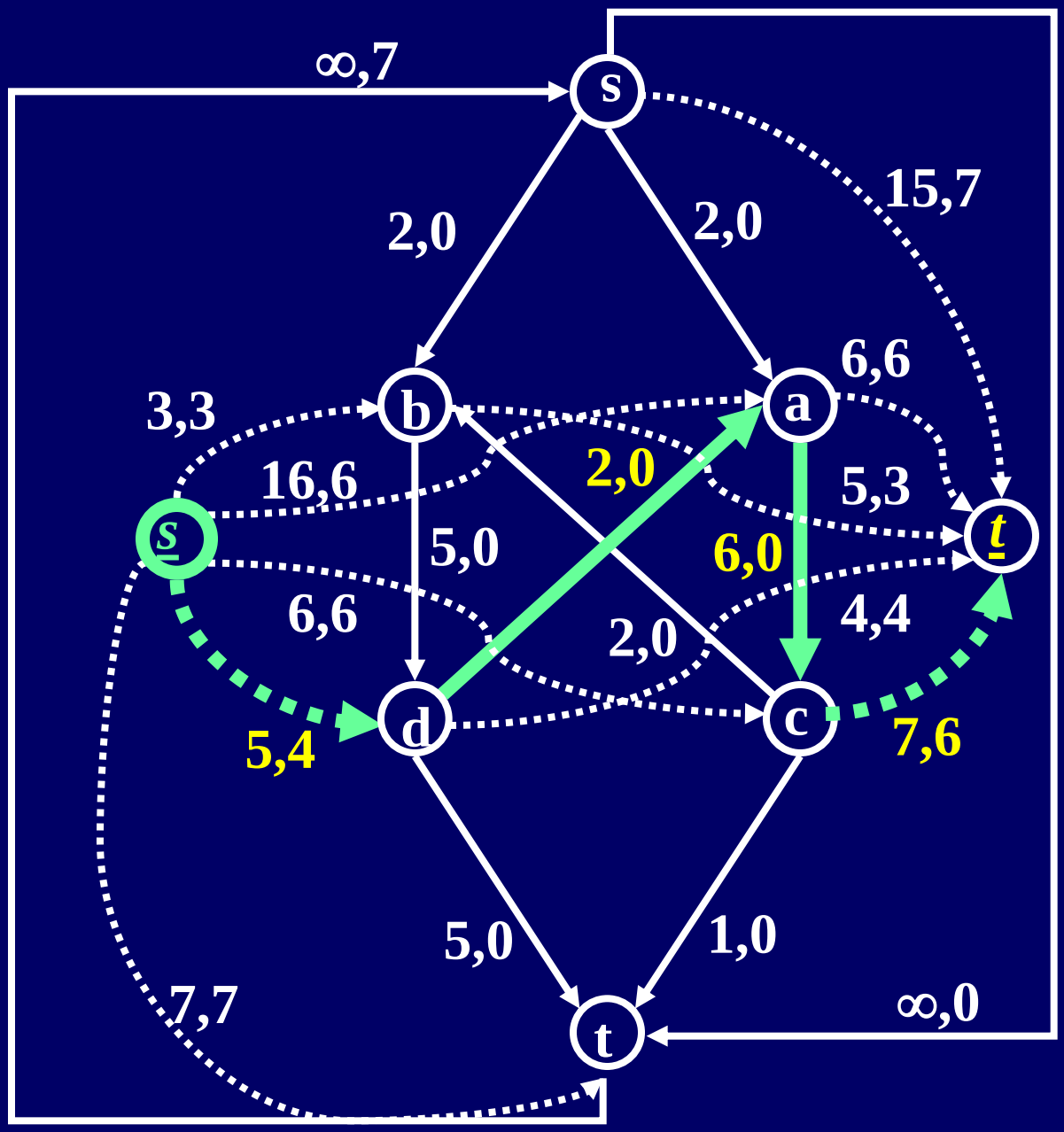
G_{aux}

G_{aux}



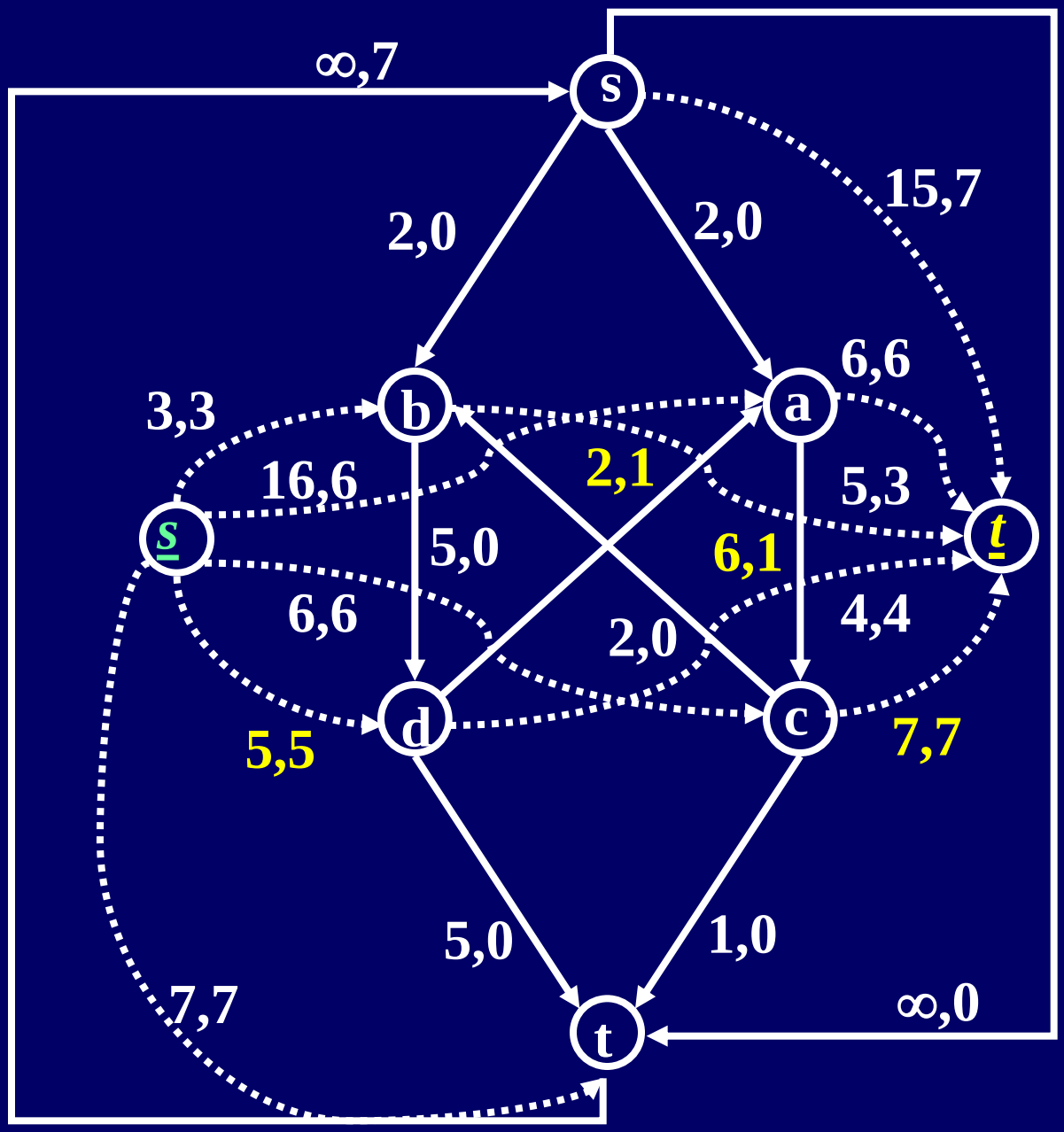
G_{aux}

G_{aux}



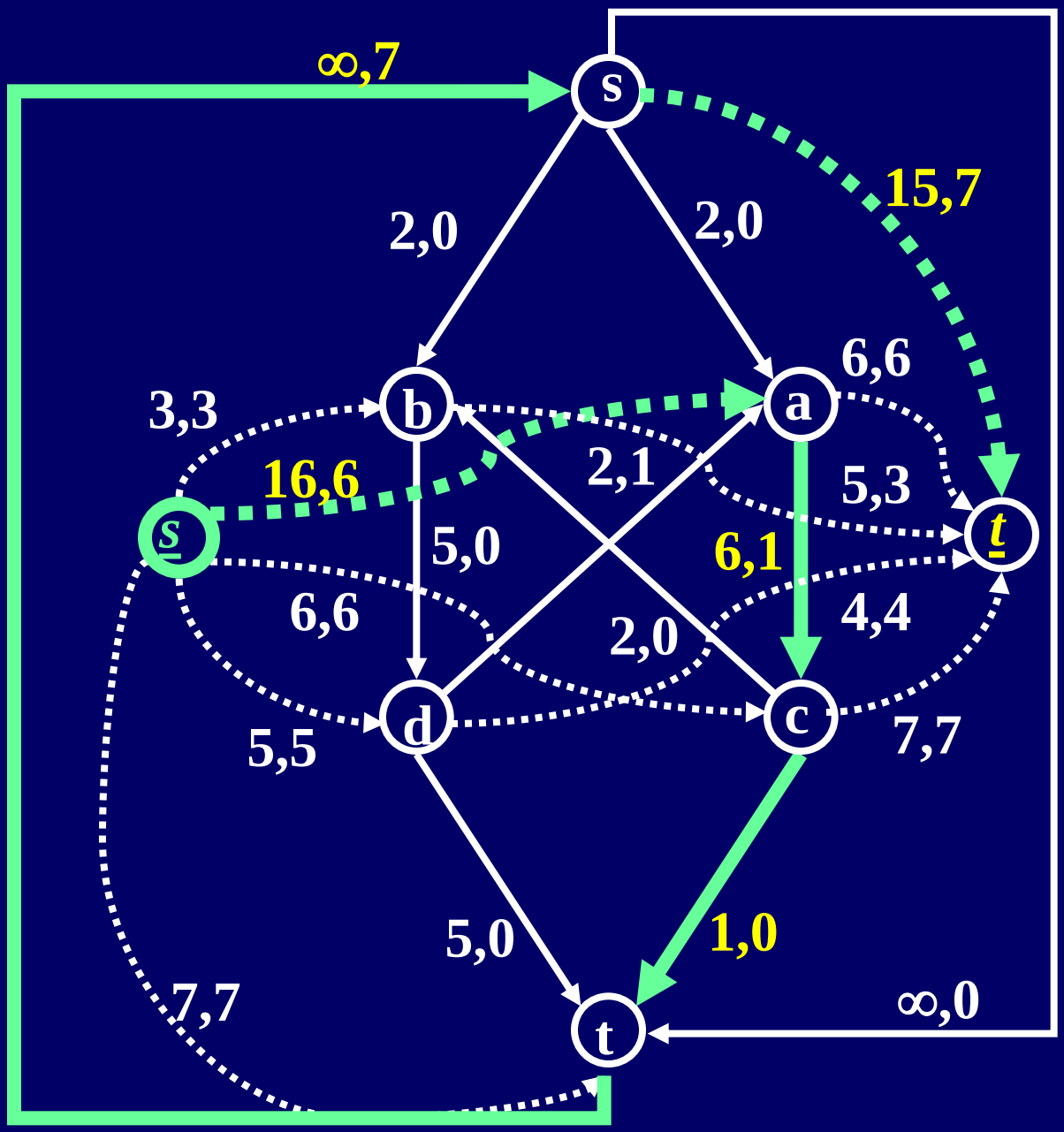
G_{aux}

G_{aux}



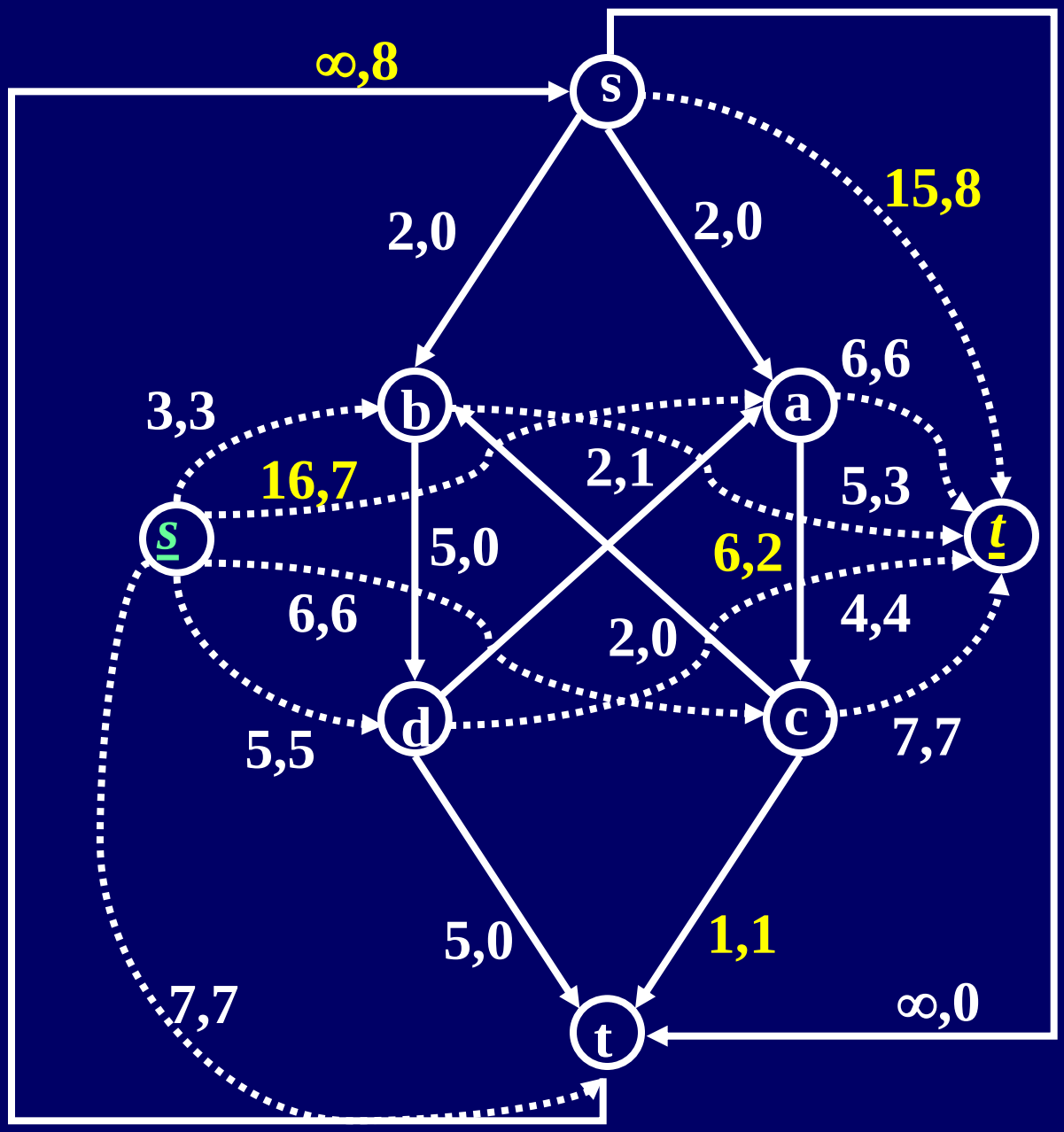
G_{aux}

G_{aux}



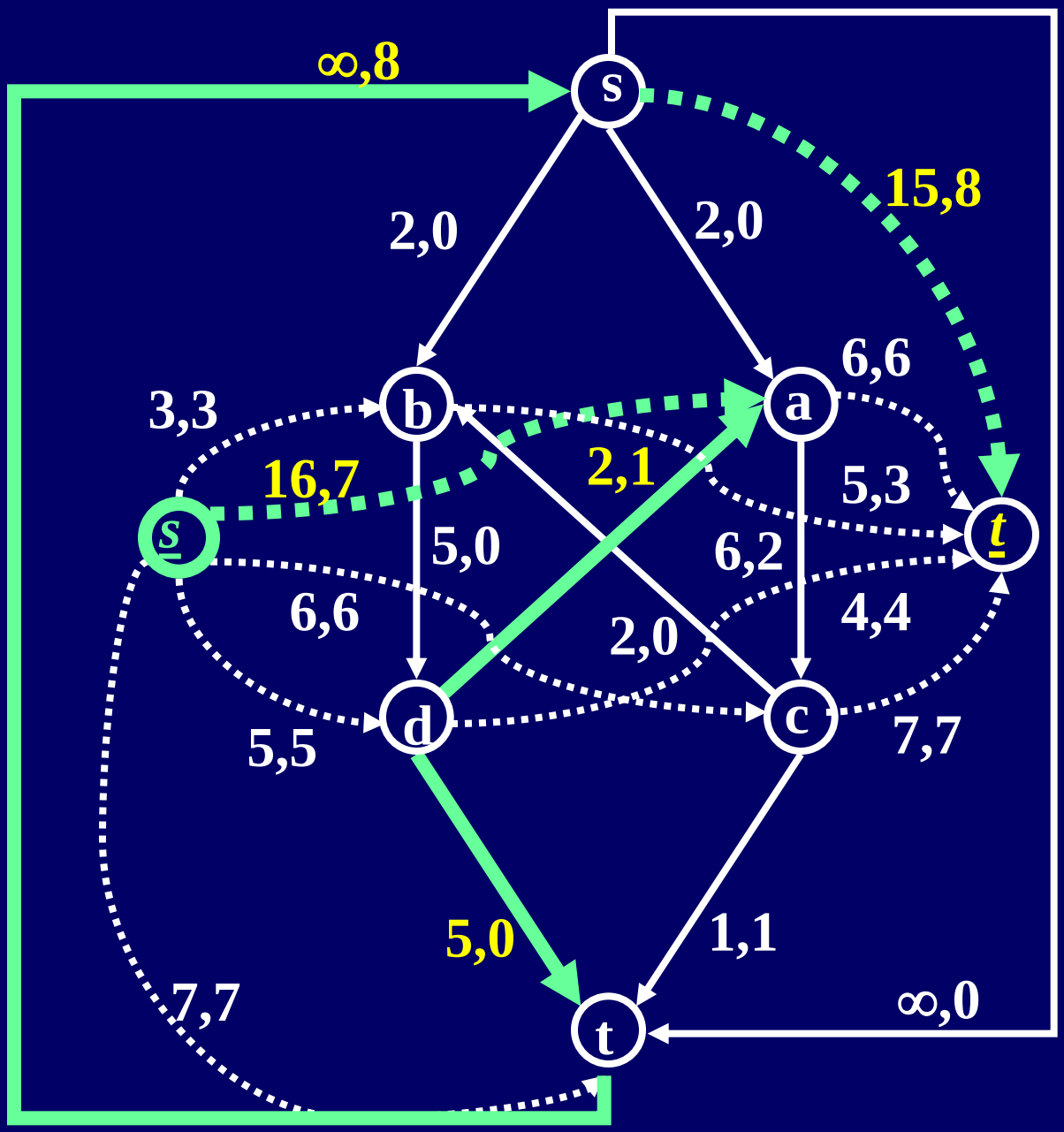
G_{aux}

G_{aux}



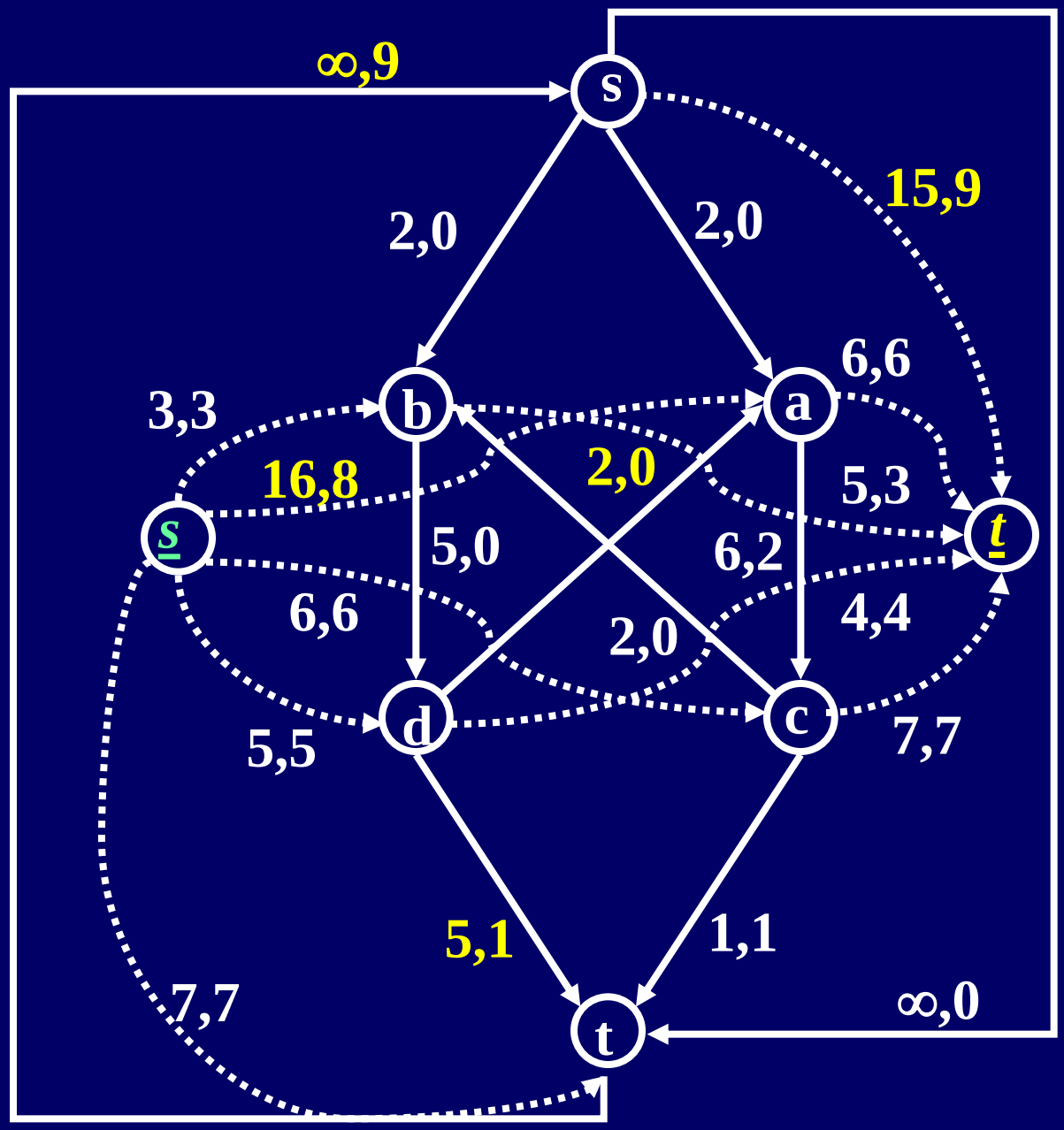
G_{aux}

G_{aux}



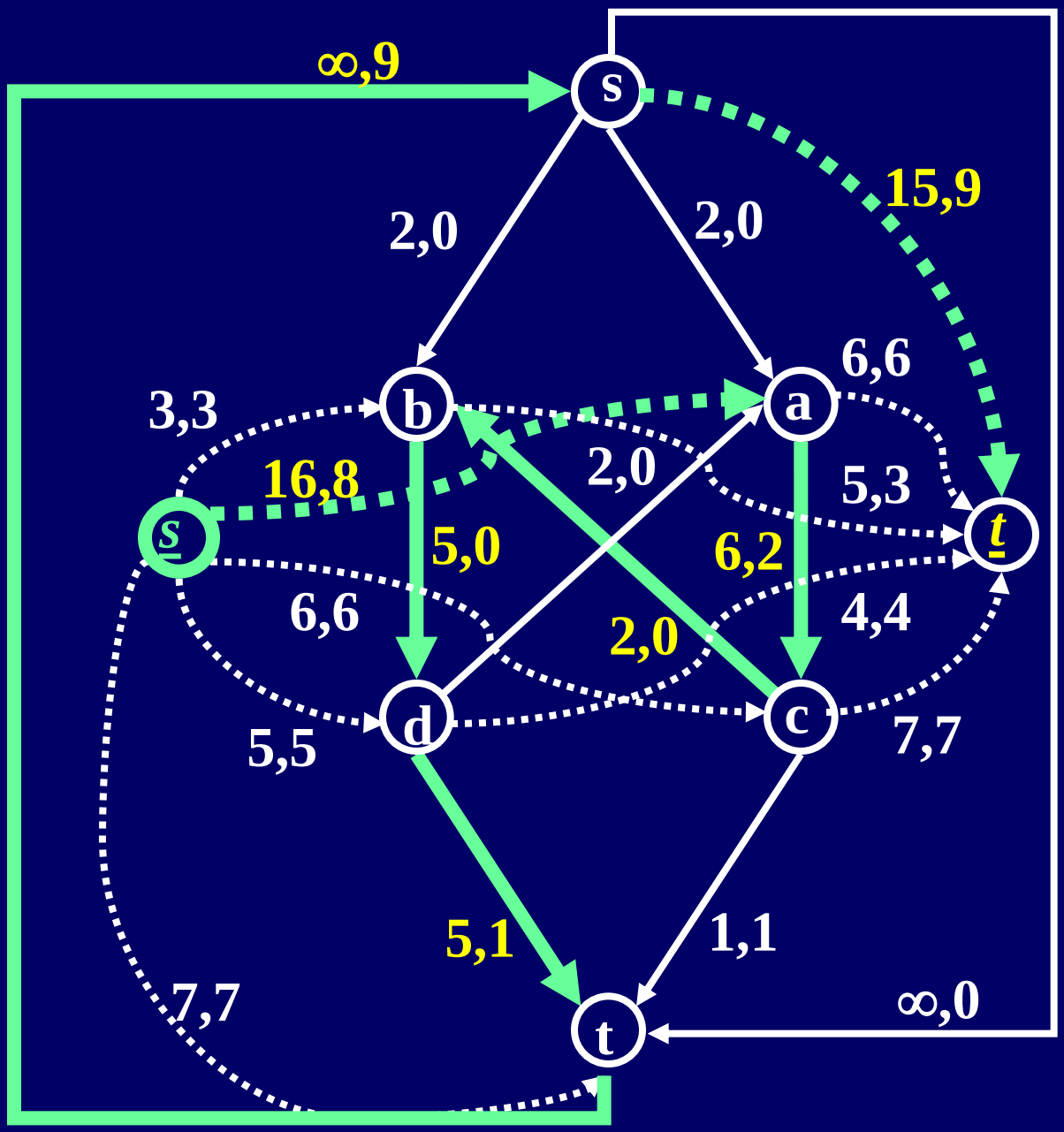
G_{aux}

G_{aux}



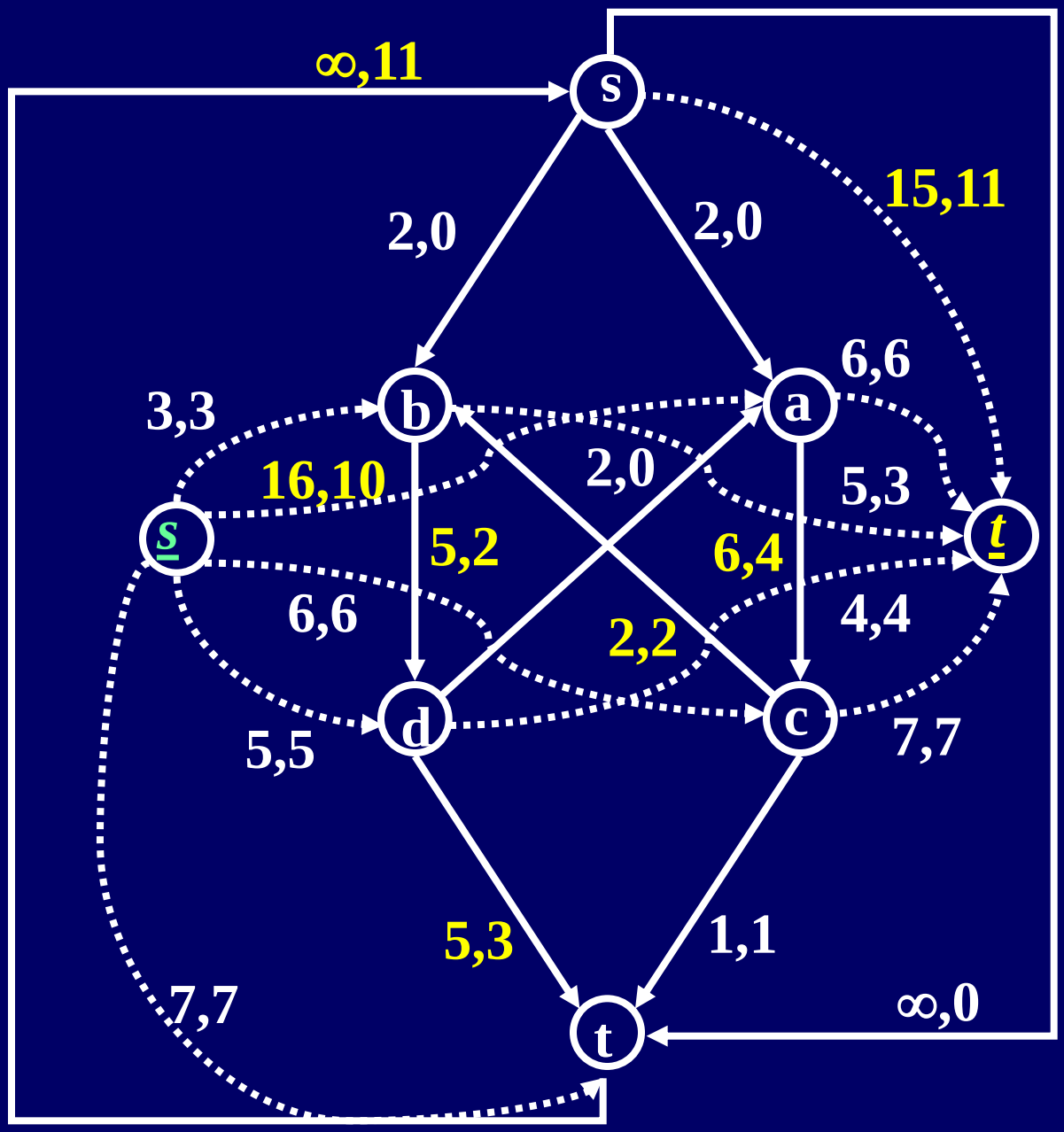
G_{aux}

G_{aux}



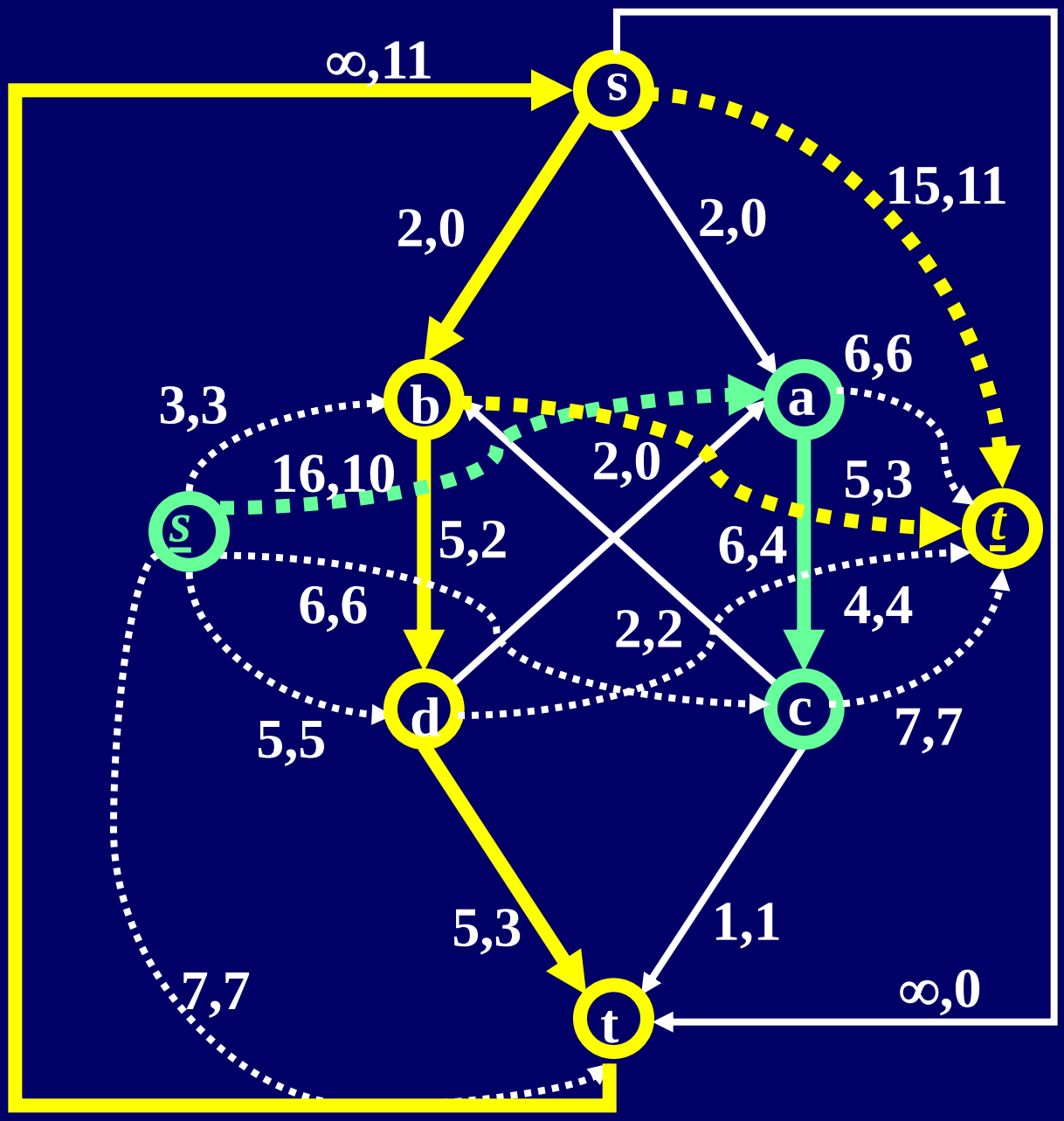
G_{aux}

G_{aux}


∞,11
s
15,11
2,0
2,0
6,6
3,3
b
a
16,10
2,0
5,3
s
5,2
6,4
t
6,6
4,4
2,2
d
c
5,5
7,7
5,3
1,1
7,7
∞,0
t

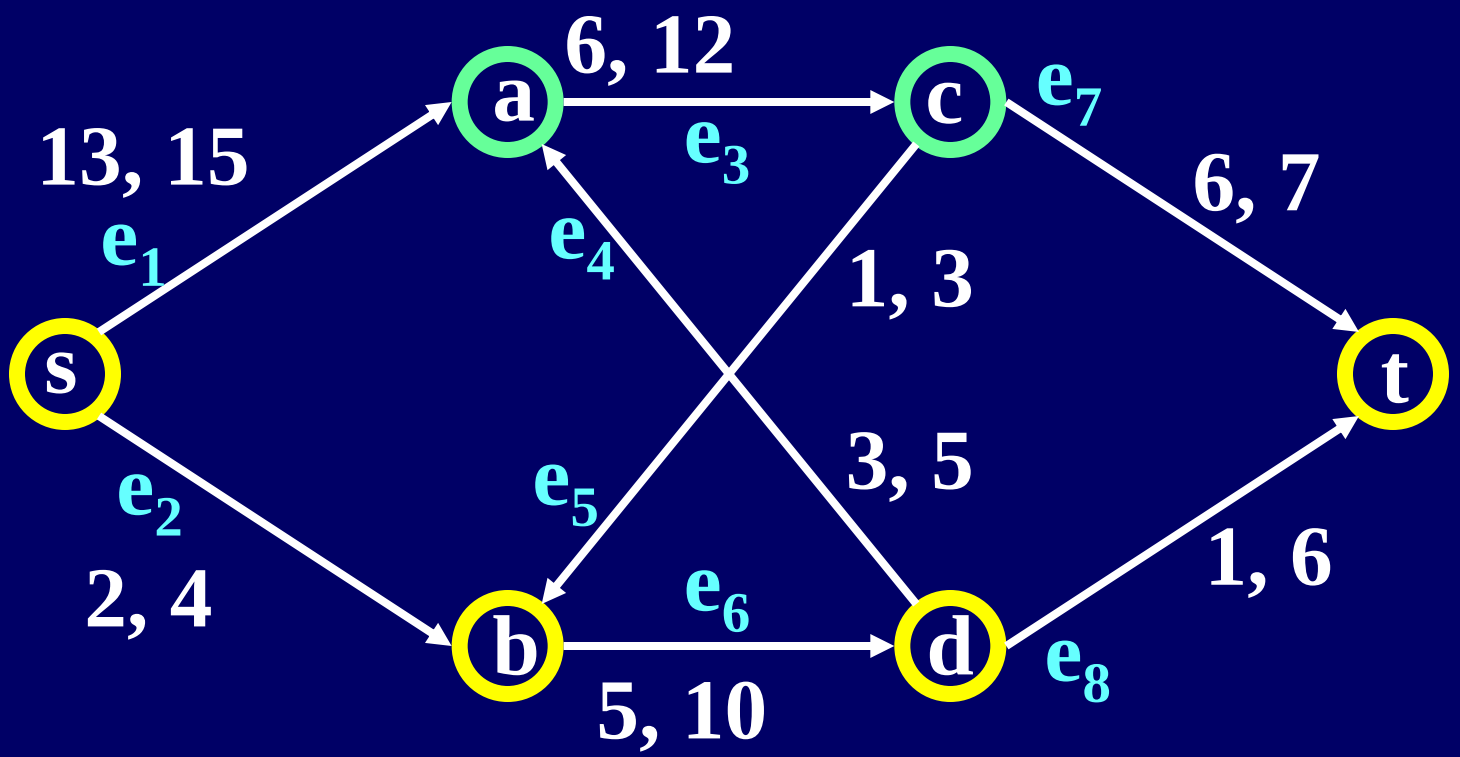
G_{aux}

G_{aux}


∞,11
s
15,11
2,0
2,0
6,6
3,3
b
a
16,10
2,0
5,3
s
t
5,2
6,4
6,6
4,4
2,2
d
c
5,5
7,7
5,3
1,1
7,7
∞,0
t

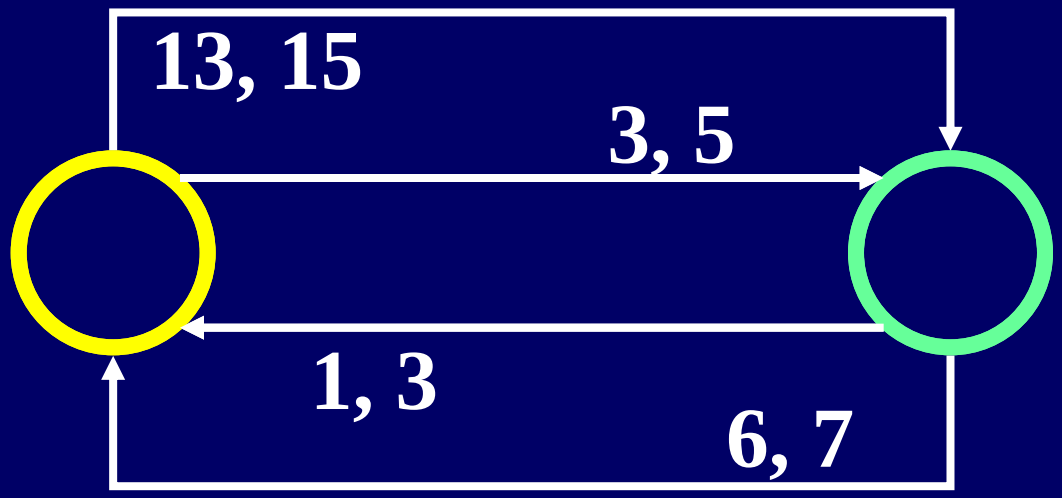
G_{aux}

Conclusion



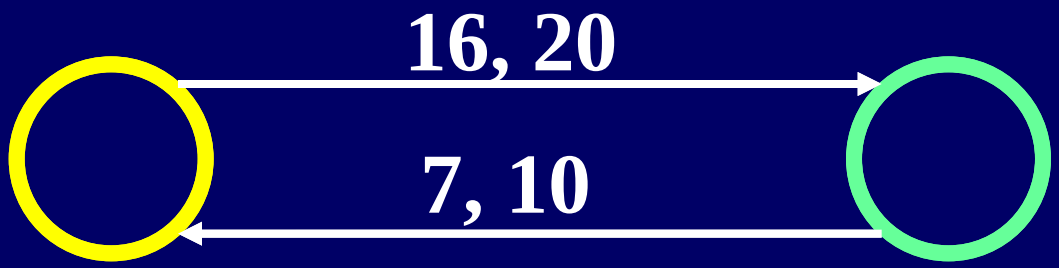
b, c

Conclusion



b, c

Conclusion



b, c