Travaux Dirigés d'ALGOrithmique AVancée

Tiffany Grenier

septembre-décembre 2013
(dernière révision : 29 décembre 2013)

Merci à Salah Eddine Alaoui, Maxim Vasilishin et Cyrille Pressat dont les notes de cours et de travaux dirigés m'ont servi à compléter les miennes pour la création de ce fichier.

Sommaire

Dans ces TDs, on utilise la notation $[\![a\,;b]\!] = [a;b] \cap \mathbb{Z}$ où a et b sont des entiers relatifs quelconques.

TD 1 :

Sommaire

1.I Enoncés

1.I.1 Exercice 1

(voir la version "exercice résolu" page 11)

On considère le graphe G de la figure suivante (1.4).

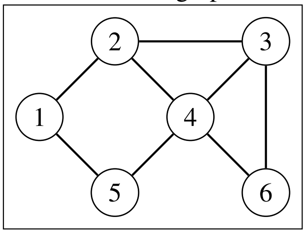


FIGURE 1.4

Écrire la matrice d'adjacence de ce graphe ainsi que son codage en terme de liste d'arêtes et de liste d'adjacence.

1.I.2 Exercice 2

(voir la version "exercice résolu" page 13)

Montrer que, dans un groupe de plus de deux personnes, il y en a toujours deux qui ont le même nombre d'amis (la relation d'amitié est supposée réciproque).

1.I.3 Exercice 3

(voir la version "exercice résolu" page 14)

On considère un graphe simple G.

1.I.3.a Question 1

Soit A la matrice d'adjacence de G. Que valent la somme des coefficients des lignes et des colonnes pour un graphe orienté ?

1.I.3.b Question 2

Soit M la matrice d'incidence de G, c'est-à-dire une matrice de taille $n(G) \times e(G)$ telle que $m_{ve} = 1$ si l'arête e est incidente au sommet v, et $m_{ve} = 0$ sinon. Écrire la matrice d'incidence du graphe 1.4.

1.I.3.c Question 3

Que valent les sommes par colonne des coefficients de M ? par ligne ?

1.I.3.d Question 4

Calculer $M{}^tM$ et en déduire la relation entre A et M.

1.I.4 Exercice 4

(voir la version "exercice résolu" page 16)

Soit $(d_1, \ldots, d_n)$ la suite des degrés des sommets d'un graphe non-orienté G à n sommets. On note δ le minimum et Δ le maximum de ces degrés.

1.I.4.a Question 1

Montrer que $\sum_{i=1}^{n} d_i = 2e(G)$ (on comptera de deux façons différentes le nombre d'incidences entre sommets et arêtes).

1.I.4.b Question 2

En déduire que $\delta \leq \frac{e(G)}{n(G)}$

1.I.5 Exercice 5

(voir la version "exercice résolu" page 16)

Un graphe non-orienté G et dit *biparti* si V(G) peut être partitionné en deux ensembles A et B tels que toute arête de G relie un sommet de A et un sommet de B.

1.I.5.a Question 1

Montrer que si G est biparti, la densité de G, d(G) vérifie $d(G) \leq \frac{n(A)n(B)}{\binom{n(A)+n(B)}{2}}$

1.I.5.b Question 2

Montrer que si G est biparti, tout cycle de G est forcément de longueur paire.

1.I.5.c Question 3

Soit G un graphe connexe dont tous les cycles sont pairs. Montrer que G est biparti. Généraliser au cas non connexe.

Indication : Introduire une sommet v. Montrer que pour tout sommet w, tous les chemins de v à w sont de même parité.

1.I.6 Exercice 6

(voir la version "exercice résolu" page 19)

Une marche entre u et v est une suite d'arêtes adjacentes entre elles, dont la première est adjacente à u et la dernière adjacente à v. La différence avec un chemin est est qu'elle peut repasser plusieurs fois parle même sommet.

1.I.6.a Question 1

Soit M la matrice d'adjacence d'un graphe non-orienté G dont les sommets sont numérotés de 1 à n. Montrer par récurrence sur k que le coefficient (i; j) de M^k est le nombre de marches de longueur k entre i et j.

1.I.6.b Question 2

Montrer que l'existence d'une marche entre i et j implique l'existence d'un chemin entre i et j.

1.I.6.c Question 3

En déduire qu'un graphe est connexe si et seulement si M^n a tous ses coefficients qui sont non nuls.

1.I.6.d Question 4

Dans le cas d'un graphe connexe, quel est le plus petit entier k tel que M^k a des coefficients tous non nuls.

Remarque : Cet exercice reste valide pour les graphes orientés en remplaçant *connexe* par *fortement connexe*

1.I.7 Exercice 7

(voir la version "exercice résolu" page 23)

Montrer que toutes les paires de plus longs chemins dans un graphe non-orienté connexe ont un sommet en commun.

Indication : Raisonner par l'absurde

1.I.8 Exercice 8

(voir la version "exercice résolu" page 24)

Appliquer l'algorithme de Dijkstra au graphe de la figure suivante (1.6) pour déterminer le chemin de poids minimal entre u et v.

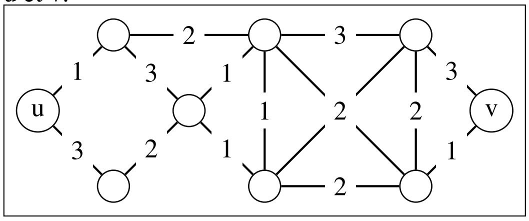


Figure 1.6

1.I.9 Exercice 9

(voir la version "exercice résolu" page 26)

À partir de l'algorithme de Dijkstra, construire un algorithme qui détermine le diamètre de chaque composante connexe d'un graphe. Quelle en est la complexité ?

1.II Exercices résolus

Exercice 1.II.1 (*Exercice 1 (TD1)*)

On considère le graphe G de la figure suivante (1.4).

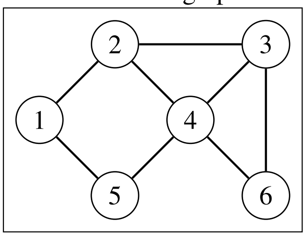


FIGURE 1.4

Écrire la matrice d'adjacence de ce graphe ainsi que son codage en terme de liste d'arêtes et de liste d'adjacence. (solution page 13)

Solution de l'exercice 1.II.2

On considère le graphe G **de la figure suivante (1.4).**

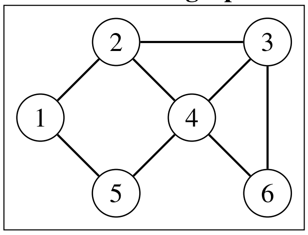


FIGURE 1.4

Écrire la matrice d'adjacence de ce graphe ainsi que son codage en terme de liste d'arêtes et de liste d'adjacence.

– Matrice d'adjacence :

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

– Liste d'arêtes :
  – (1; 2)
  – (1; 5)
  – (2; 3)
  – (2; 4)
  – (3; 4)
  – (3; 6)
  – (4; 5)
  – (4; 6)
– Liste d'adjacence :
  – (1; {2; 5})
  – (2; {1; 3; 4})
  – (3; {2; 4; 6})
  – (4; {2; 3; 5; 6})
  – (5; {1; 4})
  – (6; {3; 4})

Exercice 1.II.3 (*Exercice 2 (TD1)*)

Montrer que, dans un groupe de plus de deux personnes, il y en a toujours deux qui ont le même nombre d'amis (la relation d'amitié est supposée réciproque).(solution page 13)

Solution de l'exercice 1.II.4

Montrer que, dans un groupe de plus de deux personnes, il y en a toujours deux qui ont le même nombre d'amis (la relation d'amitié est supposée réciproque).

– Sommets : $n \in \mathbb{N}$ individus
– Arêtes : $p \in \mathbb{N}$ liens d'amitié
– Amitiés toutes réciproques : graphe non-orienté
– Amitiés toujours entre deux individus différents : graphe sans boucles
– Amitiés et individus non pondérés : graphe non-sommet-valué non-arête-valué

On peut donc modéliser la situation par un graphe simple non-orienté sans boucles non-sommet-valué et non-arête-valué.
On note A_i ($\forall i \in [\![1\,;n]\!]$) les individus.
Alors, on a $0 \leq \sum_{i=1}^{n} d(A_i) = 2p \leq n(n-1)$.
Supposons qu'il n'existe pas deux individus avec le même nombre d'amis. D'où $0 \leq d(A_i) \leq \frac{n(n-1)}{n}$ car il y a n relations différentes d'amitiés par personne : de 0 à $n-1$ amis puisqu'on ne peut être ami avec soi-même et qu'il n'y a pas plus d'un individu sans ami.
Il existe donc au moins un sommet de degré 0 et un sommet de degré $n-1$. Or, les liens d'amitié étant réciproques, on ne peut

avoir une personne amie avec tous (sauf elle-même), y compris avec un individu sans amis.
Par conséquent, il existe au moins deux individus ayant le même nombre d'amis.

Exercice 1.II.5 (*Exercice 3 (TD1)*)

On considère un graphe simple G.

Question #1 : Soit A la matrice d'adjacence de G. Que valent la somme des coefficients des lignes et des colonnes pour un graphe orienté ?

(solution page 15)

Question #2 : Soit M la matrice d'incidence de G, c'est-à-dire une matrice de taille $n(G) \times e(G)$ telle que $m_{ve} = 1$ si l'arête e est incidente au sommet v, et $m_{ve} = 0$ sinon. Écrire la matrice d'incidence du graphe 1.4.

(solution page 15)

Question #3 : Que valent les sommes par colonne des coefficients de M ? par ligne ?

(solution page 16)

Question #4 : Calculer M^tM et en déduire la relation entre A et M.

(solution page 16)

Solution de l'exercice 1.II.6

On considère un graphe simple G**.**

Notons $G = (V; E)$ avec $m = |E|$, $n = |V|$ et $V = \{v_i\}_{1\leq i\leq n}$.

$A = \left(a_{ij}\right)$ est la matrice d'adjacence de G. Elle est de dimension $n \times n$.

$M = (m_{ve})$ est la matrice d'incidence de G. Elle est de dimension $n \times m$.

1. **Soit** A **la matrice d'adjacence de** G**. Que valent la somme des coefficients des lignes et des colonnes pour un graphe orienté ?**

   – $\forall (i; j) \in [\![1\,;n]\!]^2, \begin{cases} a_{ij} = 1 & \text{si } \left(v_i; v_j\right) \in E \\ a_{ij} = 0 & \text{sinon} \end{cases} \implies \begin{cases} \forall i \in [\![1\,;n]\!] & \sum_{j=1}^{n} a_{ij} = d^+(v_i) \\ \forall j \in [\![1\,;n]\!] & \sum_{i=1}^{n} a_{ij} = d^-\left(v_j\right) \end{cases}.$

   On peut en déduire que la somme des coefficients d'une ligne vaut le degré sortant du sommet correspondant à cette ligne et que la somme des coefficients d'une colonne vaut le degré entrant du sommet correspondant à cette colonne.

   – *Question supplémentaire : Serait-ce également le cas pour un graphe non-orienté ?*
   Dans le cas d'un graphe non-orienté, $\left(v_i; v_j\right) = \left(v_j; v_i\right)$. La matrice d'adjacence est donc symétrique et $\forall i \in [\![1\,;n]\!]$, $d^-(v_i) = d^+(v_i) = d(v_i)$.

2. **Soit** M **la matrice d'incidence de** G**, c'est-à-dire une matrice de taille** $n(G) \times e(G)$ **telle que** $m_{ve} = 1$ **si l'arête** e **est incidente au sommet** v**, et** $m_{ve} = 0$ **sinon. Écrire la matrice d'incidence du graphe 1.4.**

   $\forall (i; j) \in [\![1\,;n]\!] \times [\![1\,;m]\!], \begin{cases} m_{v_i e_j} = 1 & \text{si } \exists v_k \in V \setminus \{v_i\} \,/\, e_j \in \{(v_i; v_k); (v_k; v_i)\} \\ m_{v_i e_j} = 0 & \text{sinon} \end{cases}.$

   La matrice d'incidence du graphe de l'exercice précédent (l'TD**??**) est :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccccc} (1;2) & (1;5) & (2;3) & (2;4) & (3;4) & (3;6) & (4;5) & (4;6) \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

3. **Que valent les sommes par colonne des coefficients de** M **? par ligne ?**
Une arête relie deux sommets, donc il y aura toujours deux fois la valeur 1 par colonne et il y aura sur chaque ligne autant de fois la valeur 1 que d'arêtes incidentes au sommet correspondant à cette ligne.
Ainsi, $\begin{cases} \forall i \in [\![1\,;n]\!] & \sum_{j=1}^{m} m_{v_i e_j} = d(v_i) \\ \forall j \in [\![1\,;m]\!] & \sum_{j=1}^{m} m_{v_i e_j} = 2 \end{cases}$

4. **Calculer** M^tM **et en déduire la relation entre** A **et** M.
Soit G un graphe non-orienté. Notons $\mu_{v_i e_j}$ les coefficients de M^tM.
$$\mu_{v_i e_j} = \sum_{k=1}^{m} m_{v_i e_k} m_{v_j e_k} = \begin{cases} a_{ij} & \text{si } i \neq j \text{ car } \left(m_{v_i e_k} m_{v_j e_k} = 1 \iff e_k \in \{(v_i\,; v_j)\,; (v_j\,; v_i)\}\right) \\ d(v_i) & \text{si } i = j \end{cases}$$

Exercice 1.II.7 (*Exercice 4 (TD1)*)

Soit $(d_1, \ldots, d_n)$ la suite des degrés des sommets d'un graphe non-orienté G à n sommets. On note δ le minimum et Δ le maximum de ces degrés.

Question #1 : Montrer que $\sum_{i=1}^{n} d_i = 2e(G)$ (on comptera de deux façons différentes le nombre d'incidences entre sommets et arêtes).

(solution page 16)

Question #2 : En déduire que $\delta \leq \frac{e(G)}{n(G)}$

(solution page 16)

Solution de l'exercice 1.II.8

Soit $(d_1, \ldots, d_n)$ **la suite des degrés des sommets d'un graphe non-orienté** G **à** n **sommets. On note** δ **le minimum et** Δ **le maximum de ces degrés.**

1. **Montrer que** $\sum_{i=1}^{n} d_i = 2e(G)$ **(on comptera de deux façons différentes le nombre d'incidences entre sommets et arêtes).**
Quand on somme tous les degrés, on compte chaque arête une fois par sommet qu'elle relie, c'est-à-dire 2 fois.

2. **En déduire que** $\delta \leq \frac{e(G)}{n(G)}$
$$\begin{cases} \sum_{i=1}^{n(G)} d_i = 2e(G) \\ d_i \geq \delta \end{cases} \implies \sum_{i=1}^{n(G)} d_i = 2e(G) \geq n(G)\delta \implies \delta \leq \frac{2e(G)}{n(G)} \text{ car } n(G) > 0$$

Exercice 1.II.9 (*Exercice 5 (TD1)*)

Un graphe non-orienté G et dit *biparti* si V(G) peut être partitionné en deux ensembles A et B tels que toute arête de G relie un sommet de A et un sommet de B.

Question #1 : Montrer que si G est biparti, la densité de G, d(G) vérifie $d(G) \leq \frac{n(A)n(B)}{\binom{n(A)+n(B)}{2}}$

(solution page 17)

Question #2 : Montrer que si G est biparti, tout cycle de G est forcément de longueur paire.

(solution page 17)

Question #3 : Soit G un graphe connexe dont tous les cycles sont pairs. Montrer que G est biparti. Généraliser au cas non connexe.

Indication : Introduire une sommet v. Montrer que pour tout sommet w, tous les chemins de v à w sont de même parité.(solution page 19)

Solution de l'exercice 1.II.10

Un graphe non-orienté G **et dit** ***biparti*** **si** V(G) **peut être partitionné en deux ensembles** A **et** B **tels que toute arête de** G **relie un sommet de** A **et un sommet de** B**.**

1. **Montrer que si** G **est biparti, la densité de** G**,** d(G) **vérifie** $d(G) \leq \frac{n(A)n(B)}{\binom{n(A)+n(B)}{2}}$

   La densité du graphe G est définie par $d(G) = \frac{\text{nombre arêtes présentes}}{\text{nombre arêtes possibles/permises}}$.
   Dans le cas d'un graphe simple non-orienté sans boucles, $d(G) = \frac{e(G)}{\binom{n}{2}}$.
   Par ailleurs, il y a au plus 1 arête entre chaque sommet de A et chaque sommet de B, donc $e(G) \leq n(A)n(B)$, et $n = n(A) + n(B)$.
   D'où $d(G) \leq \frac{n(A)n(B)}{\binom{n(A)+n(B)}{2}}$.

2. **Montrer que si** G **est biparti, tout cycle de** G **est forcément de longueur paire.**
   On part de A ou de B. Chaque lien doit être entre A et B, donc pour créer un chemin entre deux sommets d'un même ensemble, on doit faire des "aller-retour" via l'autre ensemble. //TODO finir

Alga - TD1 - Partie 2

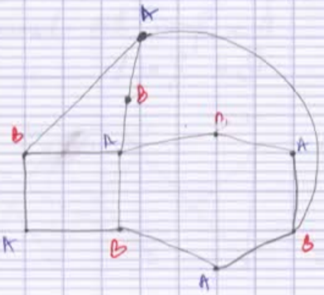
Exercice 1.5

2_ Soit C un cycle de G et v on colore parcourt le cycle et on colore en rouge chaque arête exprimée le sens A -> B et en bleu B -> A

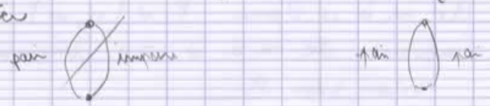
Le graphe étant biparti, c'est alternativement coloré en bleu et en rouge. Il est donc de longueur paire

(on s'arrête pas avec la même couleur => il y a une arête entre A et A faut)

3_



du sommet de départ où j'en viens arriver, il faut avoir la même parité



Soit u un sommet de graphe G

Alors, pour tout sommet w de G, tous les chemins de u à w sont de même parité

en effet : s'il exist

1er cas s'il existe deux chemins distincts, l'un de longueur paire et l'autre de longueur impaire on

on construit un cycle de longueur impaire.
$\Rightarrow$ contradiction

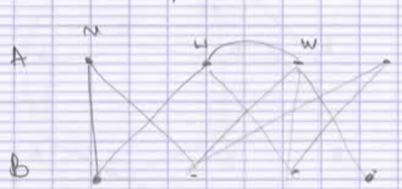
Sinon s'il existe 2 chemins non disjoints l'un pair et l'autre impaire, on peut également trouver un cycle impaire en regardant tous les cycles créés :

leur somme est impaire $\Rightarrow$ l'un d'eux est impaire $\Rightarrow$ contradiction

On peut donc écrire $V(G) = A \cup B$

$A = \{w \mid$ les chemins de v à w sont pairs$\}$

$B = \{w \mid$ ____________ impairs$\}$



il est pas possible d'avoir une arête entre (u,w) et sommet de A.

en effet, si c'est le cas, il existe un chemin pair P de v à u donc $P \cup (u, w)$ est un chemin impaire de v à w $\Rightarrow$ impossible car $w \in A$

Idem pour B

3. **Soit** G **un graphe connexe dont tous les cycles sont pairs. Montrer que** G **est biparti. Généraliser au cas non connexe.**

Indication : **Introduire une sommet** v**. Montrer que pour tout sommet** w**, tous les chemins de** v **à** w **sont de même parité.**

//TODO

Exercice 1.II.11 (*Exercice 6 (TD1)*)

Une marche entre u et v est une suite d'arêtes adjacentes entre elles, dont la première est adjacente à u et la dernière adjacente à v. La différence avec un chemin est est qu'elle peut repasser plusieurs fois parle même sommet.

Question #1 : Soit M la matrice d'adjacence d'un graphe non-orienté G dont les sommets sont numérotés de 1 à n. Montrer par récurrence sur k que le coefficient $(i; j)$ de M^k est le nombre de marches de longueur k entre i et j.

(solution page 20)

Question #2 : Montrer que l'existence d'une marche entre i et j implique l'existence d'un chemin entre i et j.

(solution page 20)

Question #3 : En déduire qu'un graphe est connexe si et seulement si M^n a tous ses coefficients qui sont non nuls.

(solution page 20)

Question #4 : Dans le cas d'un graphe connexe, quel est le plus petit entier k tel que M^k a des coefficients tous non nuls.

(solution page 20)

Remarque : Cet exercice reste valide pour les graphes orientés en remplaçant *connexe* par *fortement connexe*

Solution de l'exercice 1.II.12

Une marche entre u **et** v **est une suite d'arêtes adjacentes entre elles, dont la première est adjacente à** u **et la dernière adjacente à** v**. La différence avec un chemin est est qu'elle peut repasser plusieurs fois parle même sommet.**

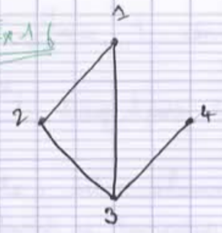
1. **Soit** M **la matrice d'adjacence d'un graphe non-orienté** G **dont les sommets sont numérotés de** 1 **à** n**. Montrer par récurrence sur** k **que le coefficient** $(i; j)$ **de** M^k **est le nombre de marches de longueur** k **entre** i **et** j**.**
   Raisonnement par récurrence :
   - cas où $k = 1$: une marche de longueur 1 est une arête, donc $m_{ij} = 1 \iff (i; j) \in E(G)$. La propriété est vraie pour $k = 1$
   - Supposons la propriété vraie à un rang k fixé, où $k \in \mathbb{N}$. Alors m^k_{uw} est le nombre de marches de longueur k entre u et w, forall$(u; w) \in V(G)$.
     $M^{k+1} = M^k.M \implies m^{k+1}_{uw} = \sum_{w=1}^{n} m^k_{uv}.m_{vw}$.
     De plus, si l'on note t^{k+1}_{uw} le nombre de marches de longueur $k + 1$ entre u et w, on obtient $t^{k+1}_{uw} = \sum_{v \in N(u)??????} t^k_{uv}$ car une marche de longueur $k + 1$ //TODO à finir et à corriger
   - $\begin{cases} \forall i \in [\![1\,;n]\!] \sum_{j=1}^{e} m_{ij} = d(v_i) \\ \forall j \in [\![1\,;e]\!] \sum_{i=1}^{n} m_{ij} = 2 \end{cases}$
   - Pour les graphes non-orientés :
     $M.{}^tM = \mu_{ij}$ où $(i; j) \in [\![1\,;n]\!]$ avec $\mu_{ij} = \sum_{k=1}^{e} m_{ik}{}^tm_{kj} = \sum_{k=1}^{e} m_{ik}m_{jk}$

2. **Montrer que l'existence d'une marche entre** i **et** j **implique l'existence d'un chemin entre** i **et** j**.**

3. **En déduire qu'un graphe est connexe si et seulement si** M^n **a tous ses coefficients qui sont non nuls.**

4. **Dans le cas d'un graphe connexe, quel est le plus petit entier** k **tel que** M^k **a des coefficients tous non nuls.**
   //TODO

Ex 1.6



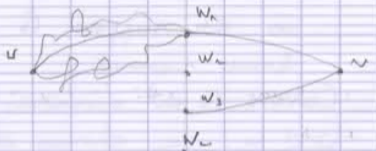
$$M = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} & \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

$K = 1$ Une marche de longueur 1 est une arête.
On a bien $m_{ij} = 1 \iff (i,j) \in E(G)$
la propriété est vrai pour $K = 1$

Supposons que la propriété est vrai au rang K
m^K_{uw} est le nombre de marches de longueur K entre u et w, $\forall (u,w)$

$M^{K+1} = M^K . M$ donc $\qquad C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$

$$m^{K+1}_{uv} = \sum_{w=1}^{n} m^K_{uw} m_{wv}$$



Or, si on note t^{K+1}_{uv} le nombre de marche de longueur $K+1$ entre u et v

$$t^{K+1}_{uv} = \sum_{w \in N(v)} t^K_{uw} = \sum_{w \in V(G)}$$

car une marche de longueur $K+1$ s'arrêtant en v correspond à une marche de longueur K s'arrêtant en un voisin de v.

1 si voisin
0 sinon

$$= \sum_{w \in V(G)} t^K_{uw} m_{wv} \quad (2)$$

Par hypothèse récurrence $t^K_{uv} = m^K_{uv}$ donc

les égalités (1) et (2) impliquent $t^{K+1}_{uv} = m^{K+1}_{uv}$

2- s'il existe une marche entre i et j qui n'est pas un chemin, il y a un sommet u répété sur la marche. On peut décomposer la marche en

M_1 : marche de i à u

M_2 : ——— u à u

M_3 : ——— u à j

alors $M_1 \cup M_3$ est encore une marche.

On répète cette opération tant qu'il y existe un sommet répété dans la marche.

Le résultat est un chemin entre i et j

3- Montrer qu'un graphe est connexe si et seulement si $\sum_{K=1}^{n} M^K$ a des coefficients tous non nuls

$\Rightarrow$ on suppose G connexe. Soit u, v deux sommets.

Comme G est connexe, il existe un chemin entre u et v. ce chemin est de longueur $K \leq n-1$

Or un chemin est une marche, donc $m^K_{uv} \geq 1$

$\Leftarrow$ On suppose que tous les coefficients de $\sum M^K$ sont non nuls

Soit u et v 2 sommets,

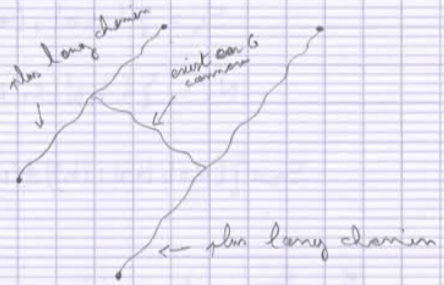
On sait $\sum_{K=1}^{n} m^K_{uv} \geq 1$

donc il existe K tel que $m^K_{uv} \geq 1$

Ceci implique qu'il existe une marche entre w et u
donc d'après 2, il existe un chemin
Ceci est vrai pour toute paire de sommets ⇒
G est connexe

4 - $\left(\sum_{j=1}^{k} M^{j}\right)$ est non nul si et seulement si
il existe une marche (donc un chemin) de longueur k
entre w et u
Or par déf le diamètre est le plus petit entier
tel qu'il existe un chemin de longueur $\leq K$ entre
toute paire de sommet

Ex _ 1.7



Exercice 1.II.13 (*Exercice 7 (TD1)*)

Montrer que toutes les paires de plus longs chemins dans un graphe non-orienté connexe ont un sommet en commun.

Indication : Raisonner par l'absurde(solution page 24)

Solution de l'exercice 1.II.14

Montrer que toutes les paires de plus longs chemins dans un graphe non-orienté connexe ont un sommet en commun.

Indication : **Raisonner par l'absurde**

Exercice 1.II.15 (*Exercice 8 (TD1)*)

Appliquer l'algorithme de Dijkstra au graphe de la figure suivante (1.6) pour déterminer le chemin de poids minimal entre u et v.

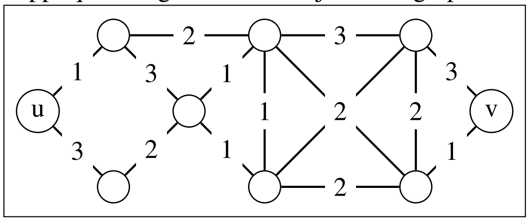


FIGURE 1.6

(solution page 26)

Solution de l'exercice 1.II.16

Appliquer l'algorithme de Dijkstra au graphe de la figure suivante (1.6) pour déterminer le chemin de poids minimal entre u **et** v**.**

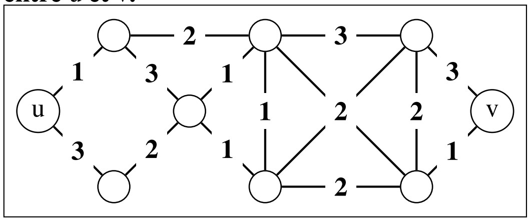
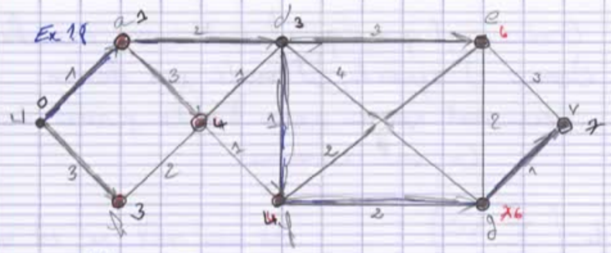


Figure 1.6

//TODO

Ex 18



S_i : l'ensemble des sommets dont on connait la distance à u au pas i, chacun étant valué par sa distance à u.

On note N_i les voisins des sommets de S_i non compris dans S.

On note E_w la dernière arête d'un plus court chemin entre u et w ← prend a distance

$\Rightarrow N_0 = \cancel{\{a, b\}}\ \{(a, 1)\ (b, 3)\}$

$S_1 = \{(u, 0), (a, 1)\}$ $\quad E_a = \{(u, a)\}$

$N_1 = \{(b, 3)\ (c, 4)\ (d, 3)\}$

$S_2 = \{(u, 0); (a, 1); (b, 3); (d, 3)\}$ $\quad E_b = \{(u, b)\}$, $E_d = \{(a, d)\}$

$\vdots$

Le chemin de poids minimal est (v, a, d, f, g, u) qui est de poids 7.

Ex 9

On a un code Dijkstra en $O(n^2)$

Sol : $\forall u, v$ applique Dijkstra(u, v) puis on prend le max …

↳ complexité en $O(n^4)$

car on appelle Dijkstra $\binom{n}{2}$ fois

Exercice 1.II.17 (***Exercice 9 (TD1)***)

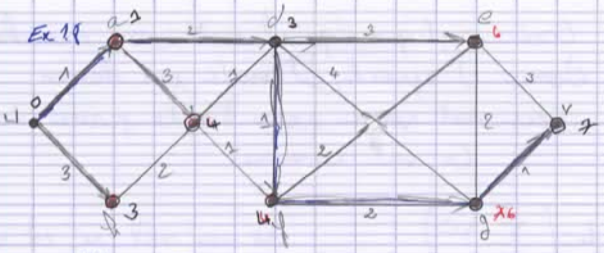
À partir de l'algorithme de Dijkstra, construire un algorithme qui détermine le diamètre de chaque composante connexe d'un graphe. Quelle en est la complexité ?(solution page 27)

Solution de l'exercice 1.II.18

À partir de l'algorithme de Dijkstra, construire un algorithme qui détermine le diamètre de chaque composante connexe

d'un graphe. Quelle en est la complexité ?

//TODO



S_i : S_i l'ensemble des sommets dont on connait la distance à u au pas i, chacun étant valué par sa distance à u.

On note N_i les voisins des sommets de S_i non compris dans S.

On note E_w la dernière arête d'un plus court chemin entre u et w

prend la distance

$\Rightarrow$ $N_0 = \cancel{\{a, b\}}\ \{(a, 1)(b, 3)\}$

$S_1 = \{(u, 0), (a, 1)\}$ $\quad E_a = \{(u, a)\}$

$N_1 = \{(b, 3)(c, 4)(d, 3)\}$

$S_2 = \{(u, 0); (a, 1); (b, 3), (d, 3)\}$ $\quad E_b = \{(u, b)\}$

$E_d = \{(a, d)\}$

$\vdots$

Le chemin de poids minimal est (v, a, d, f, g, u) qui est de poids 7.

Ex 9

On a un code Dijkstra en $O(n^2)$

Sol $\forall u, v$ applique Dijkstra (u, v) puis on prend le max
$\hookrightarrow$ complexité en $O(n^4)$
car on appelle Dijkstra $\binom{n}{2}$ fois

1.III Ce qui a été montré

- Dans une matrice d'adjacence d'un graphe orienté, la somme des coefficients d'une ligne (resp. colonne) vaut le degré sortant (resp. entrant) du sommet correspondant à cette ligne (resp. colonne)
- Dans une matrice d'adjacence d'un graphe non-orienté, la somme des coefficients d'une ligne (resp. colonne) vaut le degré du sommet correspondant à cette colonne (resp. colonne)

- $M{}^tM = A + D$ où D est une matrice diagonale contenant les degrés des sommets
- $\sum_{i=1}^{|V(G)|} d(v_i) = 2|E(G)|$
- $\delta \leq \frac{|E(G)|}{|V(G)|}$ où δ est le minimum des degrés du graphe G
- Si G est biparti en A et B composantes connexes disjointes, la densité de G, d(G), vérifie $d(G) \leq \frac{|V(A)||V(B)|}{\binom{|V(G)|}{2}}$
- Tout cycle d'un graphe connexe biparti est de longueur paire
- Un graphe connexe dont tous les cycles sont de longueur paire est biparti
- Si A est matrice d'adjacence de G, graphe non-orienté connexe ou graphe orienté fortement connexe, et que $A^k = \left(a^k{}_{ij}\right)$, $a^k{}_{ij}$ est le nombre de marches de longueur k allant du sommet v_i au sommet v_j
- L'existence d'une marche entre v_i et v_j implique l'existence d'un chemin entre v_i et v_j
- Un graphe G non-orienté est connexe si et seulement si $A^{|V(G)|}$ a tous ses coefficients qui sont non nuls
- Un graphe G orienté est fortement connexe si et seulement si $A^{|V(G)|}$ a tous ses coefficients qui sont non nuls
- Toutes les paires de plus longs chemins dans un graphe non-orienté connexe ont un sommet en commun

TD 2 :

Sommaire

2.I Énoncés

2.I.1 Exercice 1

(voir la version "exercice résolu" page 38)

Montrer qu'un arbre ayant un sommet de degré au moins k a au moins k feuilles.

2.I.2 Exercice 2

(voir la version "exercice résolu" page 39)

Un *centre* d'un arbre v est un sommet tel que $\max_u d(u; v)$ est minimal. Montrer qu'un arbre a un centre unique ou deux centres adjacents.

2.I.3 Exercice 3

(voir la version "exercice résolu" page 39)

Une molécule d'hydrocarbure saturé est une molécule de formule C_mH_n telle que chaque molécule de carbone a quatre liaisons, chaque molécule d'hydrogène a une seule liaison et il n'y a pas de cycle. Montrer que $n = 2m + 2$

2.I.4 Exercice 4

(voir la version "exercice résolu" page 40)

On considère un graphe non-orienté connexe G et une arête e de G.

2.I.4.a Question 1

Montrer qu'il existe un arbre couvrant de G utilisant l'arête e.

2.I.4.b Question 2

Une arête est un *pont* s'il existe une partition des sommets de G en V(G) = A ∪ B telle que e est la seule arête reliant A et B. Montrer que e appartient à tous les arbres couvrants de G si et seulement si e est un pont.

2.I.5 Exercice 5

(voir la version "exercice résolu" page 41)

Appliquer un algorithme de parcours en largeur au graphe de la figure suivante (2.7), en partant du sommet 3.

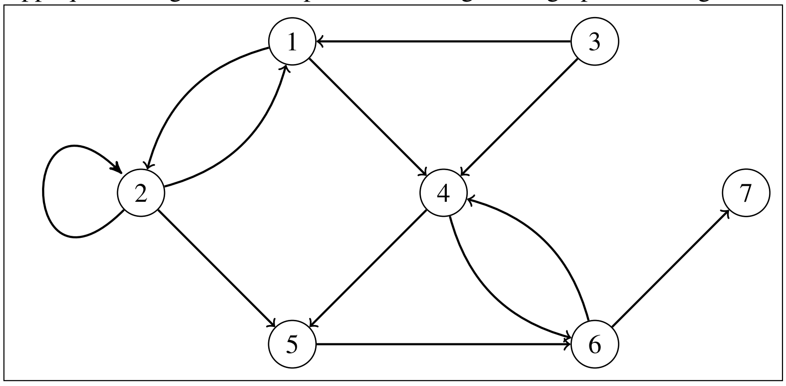


FIGURE 2.7

2.I.6 Exercice 6

(voir la version "exercice résolu" page 44)

Appliquer des parcours en profondeur au graphe 2.7 de l'exercice 2.I.5 en partant du sommet 1, en pré-ordre et en post-ordre.

2.I.7 Exercice 7

(voir la version "exercice résolu" page 49)

Soit G un graphe orienté et v un sommet de G. On appelle *descendant* de v tout sommet w tel qu'il existe un chemin orienté de v à w. On appelle *ascendant* (ou *ancêtre*) de v tout sommet u tel qu'il existe un chemin orienté de u à v.

2.I.7.a Question 1

On considère le graphe 2.7 de l'exercice 2.I.5. Déterminer l'ensemble des descendants et des ascendants du sommet 4.

2.I.7.b Question 2

Proposer deux façons de déterminer les ancêtres et les descendants d'un sommet. Quelle est la complexité de ces algorithmes ?

2.I.8 Exercice 8

(voir la version "exercice résolu" page 50)

Une *forêt* est une réunion disjointe d'arbres.

2.I.8.a Question 1

Proposer à partir du parcours en profondeur un procédé permettant de déterminer une forêt courante pour tout graphe G.

2.I.8.b Question 2

Dans le cas non-orienté, que peut-on dire du nombre d'arbres composant la forêt couvrante ? En déduire que la taille de la forêt ne dépend pas du sommet de départ choisi.

2.I.8.c Question 3

Les constatations précédents sont-elles encore vraies dans le cas orienté ?

2.I.9 Exercice 9

(voir la version "exercice résolu" page 50)

Soit v un sommet d'un graphe non-orienté connexe G et T l'arbre obtenu en appliquant un parcours en largeur depuis v. Montrer que pour tout sommet w de G, le chemin de T reliant v à w est un plus court chemin entre v et w dans G. en déduire un algorithme plus efficace que l'algorithme de Dijkstra pour calculer le diamètre d'un graphe.

2.I.10 Exercice 10

(voir la version "exercice résolu" page 51)

Écrire le pseudo-code d'un algorithme qui, étant donné un graphe non-orienté G, permet soit de renvoyer un cycle de G, soit de conclure que G est acyclique.

2.I.11 Exercice 11

(voir la version "exercice résolu" page 52)

Reprendre l'exercice 2.I.10 dans le cas d'un graphe orienté. Déduire à partir de cet algorithme que, si tous les sommets d'un graphe orienté sont de degré sortant au moins 1, ce graphe contient un cycle orienté.

2.I.12 Exercice 12

(voir la version "exercice résolu" page 53)

Écrire le pseudo-code d'un algorithme qui, étant donné un sommet v dans un graphe orienté G, détermine un plus court cycle passant par v, en lançant deux parcours en largeur. Comment faire avec un seul parcours en largeur ?

2.I.13 Exercice 13

(voir la version "exercice résolu" page 55)

Déterminer un arbre couvrant de poids minimal pour le graphe de la figure suivante (2.8).

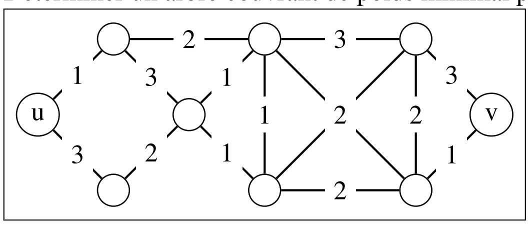


FIGURE 2.8

2.I.14 Exercice 14

(voir la version "exercice résolu" page 56)

Les graphes suivants contiennent-ils un chemin eulérien, un cycle eulérien, un chemin hamiltonien, un cycle hamiltonien ?

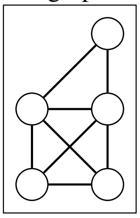


FIGURE 2.9

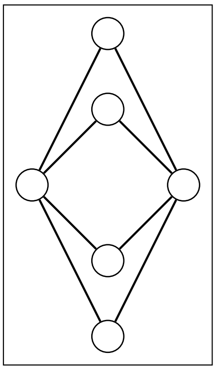


FIGURE 2.10

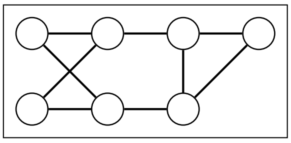


FIGURE 2.11

2.I.15 Exercice 15

(voir la version "exercice résolu" page 57)

Résoudre le problème du postier sur le graphe de la figure suivante (2.12).

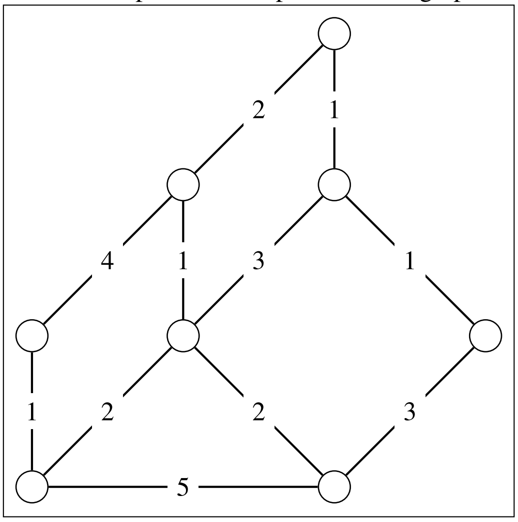


FIGURE 2.12

2.I.16 Exercice 16

(voir la version "exercice résolu" page 57)

Pour résoudre le problème du voyageur de commerce, on propose l'heuristique suivante :
- construire un arbre couvrant de poids minimal
- parcourir les sommets de cet arbre comme un parcours en profondeur

2.I.16.a Question 1

Quelle est la complexité de cet algorithme ?

2.I.16.b Question 2

Soit H le graphe induit par les arêtes utilisées par un parcours optimal. Montrer que H contient un arbre couvrant. En déduire que la solution proposée par l'heuristique est une 2-approximation, c'est-à-dire qu'elle est au pire de poids deux fois plus important que la solution optimale.

2.II Exercices résolus

Exercice 2.II.1 (*Exercice 1 (TD2)*)

Montrer qu'un arbre ayant un sommet de degré au moins k a au moins k feuilles.

Solution de l'exercice 2.II.2 (*Solution 1*)

Montrer qu'un arbre ayant un sommet de degré au moins k **a au moins** k **feuilles.**
Soit T un graphe connexe acyclique, c'est-à-dire un arbre. On peut enraciner l'arbre par rapport à n'importe lequel de ss sommets. Soit u un sommet de degré k de T. Alors u a $k - 1$ fils et 1 père dans chacun des arbres enracinés sur T où il n'est pas racine,et k fils et 0 père dans ceux où il est racine, car, dans un arbre enraciné, un sommet ne peut avoir ni plus ni moins qu'un père s'il n'est pas la racine, qui n'a pas de père.
Pour chacun de ces fils, deux cas se présentent :
- il s'agit d'une feuille
- il ne s'agit pas d'une feuille et le degré de ce fils est $l > 1$. Ce fils a donc lui-même $l - 1$ fils car il ne peut être racine de l'arbre enraciné sur T.
  Comme il n'y a pas de cycle dans un arbre, il y a forcément au moins un descendant de ce fils de u qui est une feuille, c'est ce que l'on considère dans un parcours en largeur ou en profondeur au moment de remonter un branche faute de voisins

Solution de l'exercice 2.II.3 (*Solution 2*)

Montrer qu'un arbre ayant un sommet de degré au moins k **a au moins** k **feuilles.**
On veut montrer que le nombre de sommets de degré 1 est nécessairement supérieur ou égal à k dans un arbre possédant au moins un sommet de degré au moins k.
Soient T un arbre à n sommets, v le degré de degré au moins k et N le nombre de sommets de degré 1 de cet arbre.
T étant un arbre, c'est-à-dire un graphe connexe acyclique, il possède exactement $n - 1$ arêtes.
Donc $\sum_{u \in V(T)} d(u) = 2|E(T)| = 2(n - 1)$.
De plus, T contient a sommet de degré au moins k (c'est v), N sommets de degré 1 et $n - N - 1$ autres sommets. Ces autres sommets sont forcément de degré au moins 2 car tout arbre étant connexe, T ne peut contenir de sommet de degré 0.

Ainsi peut-on minorer la somme des degrés des sommets de T de la façon suivante :

$$\sum_{u \in V(T)} d(u) \geq 1 \times k + N \times 1 + (n - N - 1) \times 2$$

$\iff$ $2n - 2 \geq k + N + 2n - 2n - 2$

$\iff$ $0 \geq k - N$

$\iff$ $N \geq k$

T possède par conséquent au moins k sommets de degré 1.

Solution de l'exercice 2.II.4 (*Solution 3*)

Montrer qu'un arbre ayant un sommet de degré au moins k a au moins k feuilles.

Raisonnons par récurrence sur le nombre de sommets dans un arbre.

Soit T un arbre à $n \geq 2$ sommets dont 1 sommet de degré k noté u_0, k étant un entier supérieur ou égal à 1 fixé mais quelconque.

Initialisation : Pour $n = 2$

T a deux sommets notés u_0 et u_1. Comme T est connexe acyclique, u_0 et u_1 sont forcément reliés par une et une seule arête, donc $k = d(u_0) = d(u_1) = 1$ et T possède $2 \geq= k$ feuilles.

Itération sur n ***fixé*** : Supposons désormais que T possède n sommets, n étant un entier naturel fixé, dont u_0 de degré $k \geq 1$, et que T vérifie l'hypothèse, c'est-à-dire qu'il a au moins k feuilles. Notons N le nombre de ses feuilles ($N \geq k$).

Ajoutons un sommet v à T, notons T' le nouvel arbre formé et regardons sur l'hypothèse est vérifiée pour T'. 3 cas se présentent :

1. Le sommet ajouté v est voisin de u_0 dans T'. Comme T' est connexe acyclique, v n'a pas d'autre voisin dans T' que u_0. Le degré de u_0 devient $k + 1$ et T' possède une feuille de plus que T, c'est-à-dire $N + 1 \geq k + 1 = d(u_0)$ feuilles.
2. Le sommet ajouté v n'est pas voisin de u_0 mais d'un autre sommet w qui était une feuille dans T. De même que précédemment, v n'a pas d'autre voisin dans T' que w et v est une feuille de T'. De plus, w n'est pas une feuille dans T' et le degré de u_0 est le même dans T et dans T'. T' possède donc $N + 1 - 1 = N \geq k = d(u_0)$ feuilles.
3. Le sommet ajouté v n'est pas voisin de u_0 mais d'un autre sommet w qui n'était pas une feuille dans T. De même que précédemment, v n'a pas d'autre voisin dans T' que w et v est une feuille de T'. De plus, w n'est pas plus une feuille dans T' quand dans T et le degré de u_0 est le même dans T que dans T'. T' possède donc $N + 1 \geq k = d(u_0)$ feuilles.

Conclusion : On a ainsi montré de tout arbre T possédant n sommets, $n \in N$, dont un sommet de degré au moins $k \in \mathbb{N}$ a au moins k feuilles.

Exercice 2.II.5 (*Exercice 2 (TD2)*)

Un *centre* d'un arbre v est un sommet tel que $\max_u d(u; v)$ est minimal. Montrer qu'un arbre a un centre unique ou deux centres adjacents.(solution page 39)

Solution de l'exercice 2.II.6

Un *centre* d'un arbre v est un sommet tel que $\max_u d(u; v)$ est minimal. Montrer qu'un arbre a un centre unique ou deux centres adjacents.

NON FAIT

Exercice 2.II.7 (*Exercice 3 (TD2)*)

Une molécule d'hydrocarbure saturé est une molécule de formule C_mH_n telle que chaque molécule de carbone a quatre liaisons, chaque molécule d'hydrogène a une seule liaison et il n'y a pas de cycle. Montrer que $n = 2m + 2$(solution page 40)

Solution de l'exercice 2.II.8

Une molécule d'hydrocarbure saturé est une molécule de formule C_mH_n telle que chaque molécule de carbone a quatre

liaisons, chaque molécule d'hydrogène a une seule liaison et il n'y a pas de cycle. Montrer que $n = 2m + 2$
NON FAIT

Exercice 2.II.9 (*Exercice 4 (TD2)*)

On considère un graphe non-orienté connexe G et une arête e de G.

Question #1 : Montrer qu'il existe un arbre couvrant de G utilisant l'arête e.

(solution page 40)

Question #2 : Une arête est un *pont* s'il existe une partition des sommets de G en $V(G) = A \cup B$ telle que e est la seule arête reliant A et B. Montrer que e appartient à tous les arbres couvrants de G si et seulement si e est un pont.

(solution page 40)

Solution de l'exercice 2.II.10

1. **Montrer qu'il existe un arbre couvrant de** G **utilisant l'arête** e**.**

2. **Une arête est un *pont* s'il existe une partition des sommets de** G **en** $V(G) = A \cup B$ **telle que** e **est la seule arête reliant** A **et** B**. Montrer que** e **appartient à tous les arbres couvrants de** G **si et seulement si** e **est un pont.**
   Si un graphe G est connexe mais qu'il n'y a qu'une arête reliant deux de ses composantes connexes disjointes A et B, cette arête est un pont.
   - Soit $e \in E(G)$ un pont. Tout arbre couvrant de G contient tous les sommets de A et tous les sommets de B, et les arbres étant connexes, tout arbre couvrant de G contient au moins une arête reliant un sommet de A à un sommet de B. Par définition d'un pont, il ne peut s'agir que de l'arête e, seule arête reliant A et B. Par conséquent, tout arbre couvrant de G contient e.
   - ***option 1*** : Soit e une arête appartenant à tous les arbres couvrants.
     S'il existait 2 arêtes (dont e) reliant des sommets de A à des sommets de B, A et B étant connexes disjoints, alors il existerait parmi tous les arbres couvrants de G au moins un arbre couvrant de G contenant un arbre couvrant de A et un arbre couvrant de B disjoints reliés par l'une de ces deux arêtes uniquement (sans quoi il y aurait une boucle et ce ne serait plus un arbre). Il n'existe ainsi pas d'autre arête reliant des sommets de A à des sommets de B si e appartient à tous les arbres couvrants. e est donc un pont.

     option 2, à préférer : Supposons que e n'est pas un pont.
     Il existe alors une partition de G en A et B tous deux connexes telle que e rejoint A et B mais n'est pas la seule arête dans ce cas. À l'aide d'un parcours en largeur ou en profondeur, on peut donc obtenir un arbre T_A couvrant A et un arbre T_B couvrant B et disjoint de T_A. En ajoutant à $T_A \cup T_B$ n'importe quelle arête différente de e entre A et B, on obtient un arbre T couvrant G mais ne contenant pas e. Par conséquent, si e n'est pas un pont, e n'appartient pas à tous les arbres couvrants de G. Réciproquement, toute arête contenue dans tous les arbres couvrants de G est un pont.

Exercice 2.II.11 (*Exercice 5 (TD2)*)

Appliquer un algorithme de parcours en largeur au graphe de la figure suivante (2.7), en partant du sommet 3.

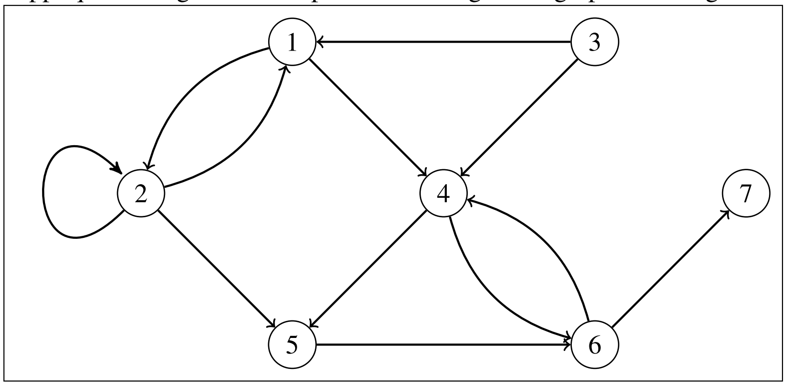
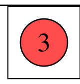
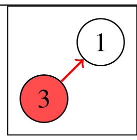
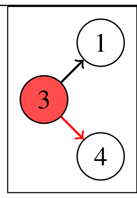
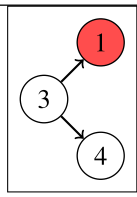
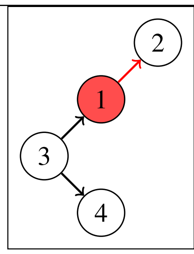
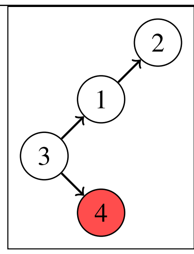
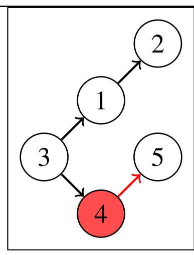


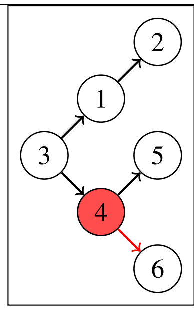
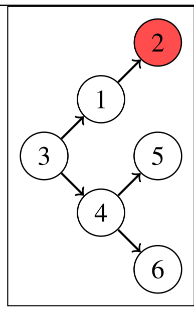
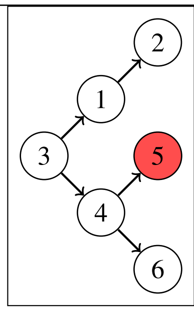
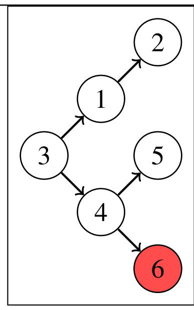
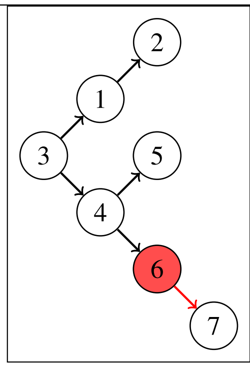
FIGURE 2.7

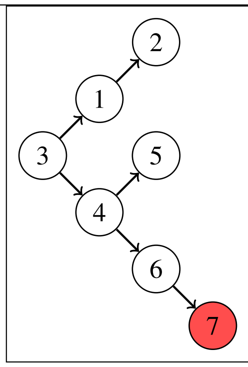
(solution page 42)

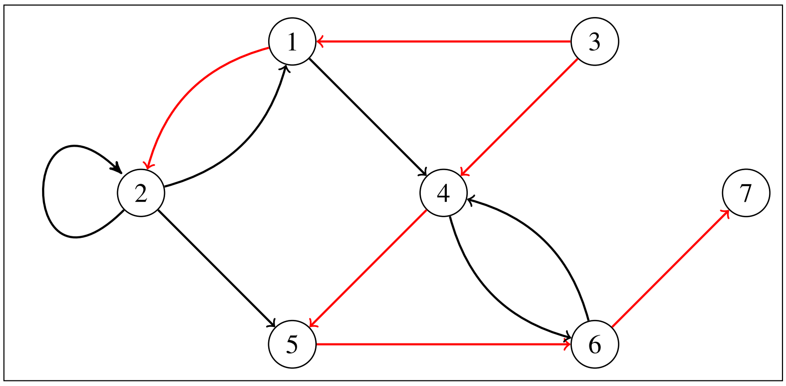
Solution de l'exercice 2.II.12

Appliquer un algorithme de parcours en largeur au graphe de la figure suivante (2.7), en partant du sommet 3**.**

Sommet considéré	Sommets déjà vus	Descendants déjà vus	Descendants restants	Parcours réalisé
3	3	-	1 ;4	
3	3 ;1	1	4	
3	3 ;1 ;4	1 ;4	-	
1	3 ;1 ;4	4	2	
1	3 ;1 ;4 ;2	2 ;4	-	
4	3 ;1 ;4 ;2	-	5 ;6	
4	3 ;1 ;4 ;2 ;5	5	6	

4	3 ;1 ;4 ;2 ;5 ;6	5 ;6	-	
2	3 ;1 ;4 ;2 ;5 ;6	1 ;5 ;2	-	
5	3 ;1 ;4 ;2 ;5 ;6	6	-	
6	3 ;1 ;4 ;2 ;5 ;6	4	7	
6	3 ;1 ;4 ;2 ;5 ;6 ;7	4 ;7	-	

7	3 ;1 ;4 ;2 ;5 ;6 ;7	-	-	

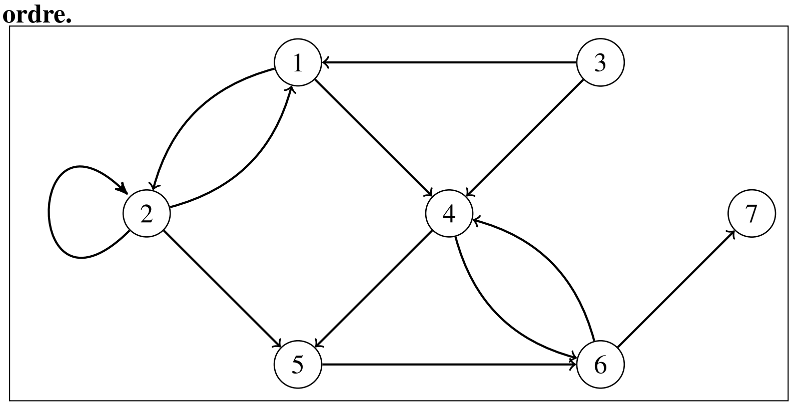


Exercice 2.II.13 (*Exercice 6 (TD2)*)

Appliquer des parcours en profondeur au graphe 2.7 de l'exercice 2.I.5 en partant du sommet 1, en pré-ordre et en post-ordre. (solution page 44)

Solution de l'exercice 2.II.14

Appliquer des parcours en profondeur au graphe 2.7 de l'exercice 2.I.5 en partant du sommet 1**, en pré-ordre et en post-ordre.**

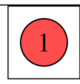
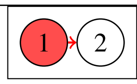
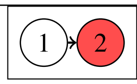
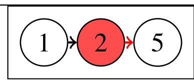
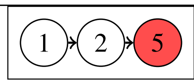
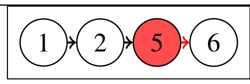
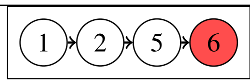
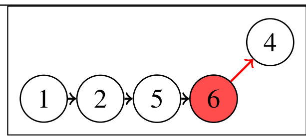
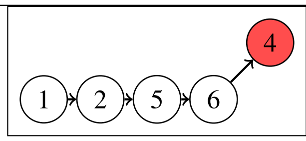
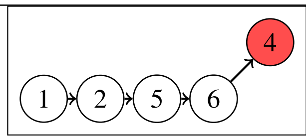
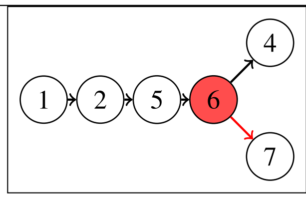
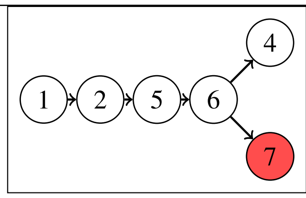
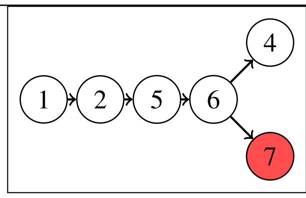
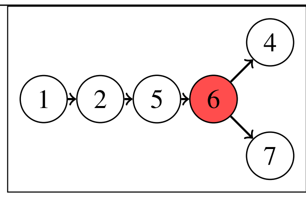


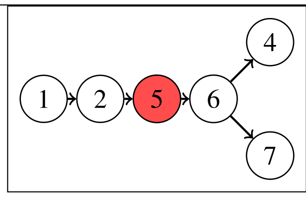
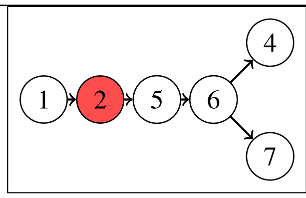
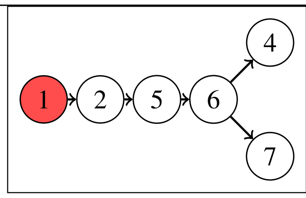
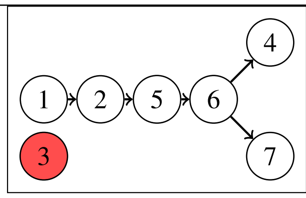
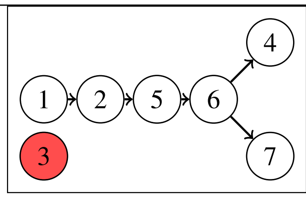
Pré-ordre : on numérote les sommets au moment où on lance le parcours en profondeur sur eux

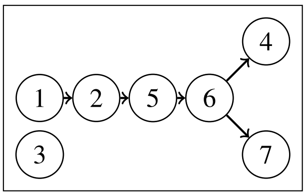
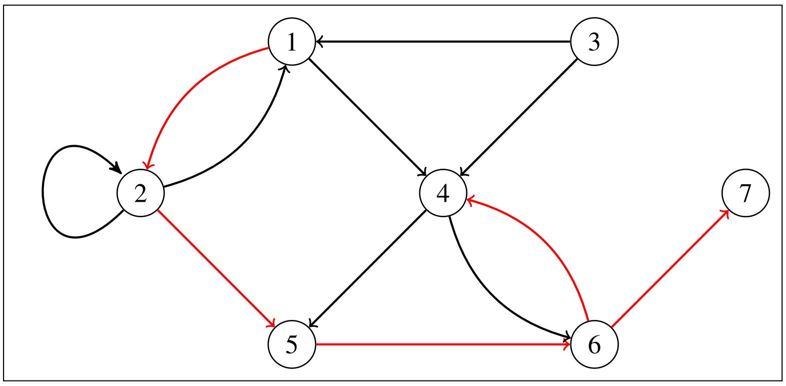
Post-ordre : on numérote les sommets au moment où on clôt le parcours en profondeur sur eux

Faisons d'abord un parcours en profondeur en respectant l'orientation des arêtes et en partant du sommet 1, sans considération de numérotation pré-ordre ou post-ordre.

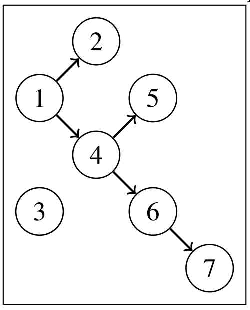
Sommet considéré	Sommets déjà vus	Descendants déjà vus	Descendants restants	Parcours réalisé

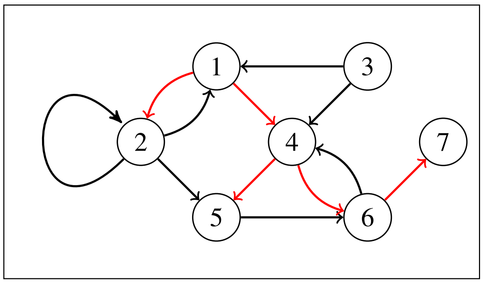
1 (ouvert)	1	-	2 ;4	
1 (ouvert)	1 ;2	2	4	
2 (ouvert)	1 ;2	2 ;1	5	
2 (ouvert)	1 ;2 ;5	2 ;1 ;5	-	
5 (ouvert)	1 ;2 ;5	-	6	
5 (ouvert)	1 ;2 ;5 ;6	6	-	
6 (ouvert)	1 ;2 ;5 ;6	-	4 ;7	
6 (ouvert)	1 ;2 ;5 ;6 ;4	4	7	
4 (ouvert)	1 ;2 ;5 ;6 ;4	5 ;6	-	
4 (fermé)	1 ;2 ;5 ;6 ;4	5 ;6	-	
6 (ouvert)	1 ;2 ;5 ;6 ;4	4 ;7	-	
7 (ouvert)	1 ;2 ;5 ;6 ;4 ;7	-	-	
7 (fermé)	1 ;2 ;5 ;6 ;4 ;7	-	-	
6 (fermé)	1 ;2 ;5 ;6 ;4 ;7	4 ;7	-	

5 (fermé)	1 ;2 ;5 ;6 ;4 ;7	6	-	
2 (fermé)	1 ;2 ;5 ;6 ;4 ;7	2 ;1 ;5	-	
1 (fermé)	1 ;2 ;5 ;6 ;4 ;7	2 ;4	-	
3 (ouvert)	1 ;2 ;5 ;6 ;4 ;7 ;3	-	-	
3 (fermé)	1 ;2 ;5 ;6 ;4 ;7 ;3	-	-	

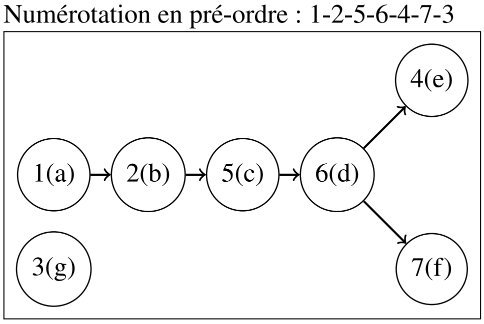

1
2
5
6
4
7
3

1
3
2
4
7
5
6

D’autres résultats sont possibles, tels que le parcours suivant :

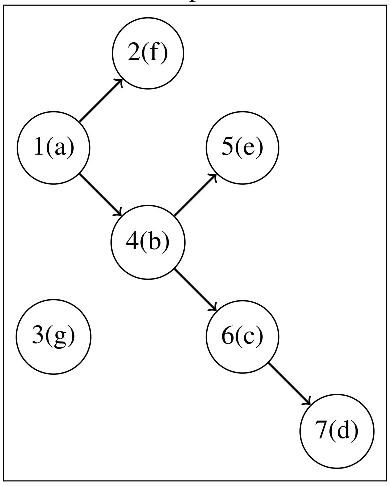




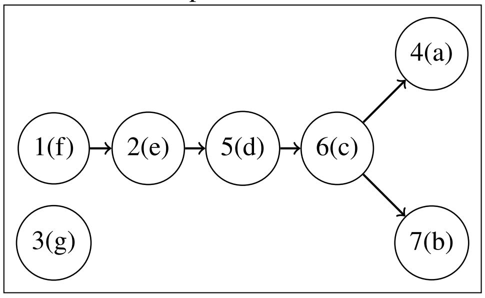
1. – Numérotation en pré-ordre : 1-2-5-6-4-7-3



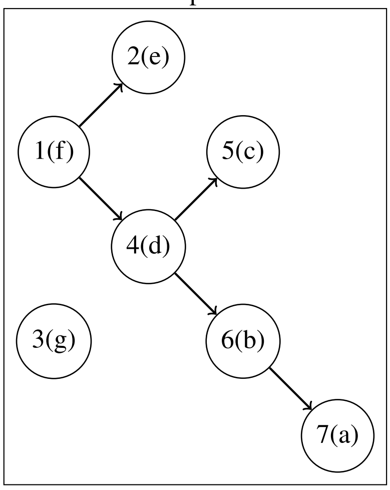
– Numérotation en pré-ordre : 1-4-6-7-5-2-3



2. – Numérotation en post-ordre : 4-7-6-5-2-1-3



– Numérotation en post-ordre : 7-6-5-4-2-1-3



Exercice 2.II.15 (*Exercice 7 (TD2)*)

Soit G un graphe orienté et v un sommet de G. On appelle *descendant* de v tout sommet w tel qu'il existe un chemin orienté de v à w. On appelle *ascendant* (ou *ancêtre*) de v tout sommet u tel qu'il existe un chemin orienté de u à v.

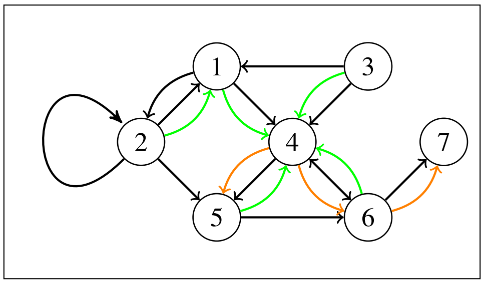
Question #1 : On considère le graphe 2.7 de l'exercice 2.I.5. Déterminer l'ensemble des descendants et des ascendants du sommet 4.

(solution page **??**)

Question #2 : Proposer deux façons de déterminer les ancêtres et les descendants d'un sommet. Quelle est la complexité de ces algorithmes ?

(solution page **??**)

Solution de l'exercice 2.II.16



- Descendants du sommet 4 (en orange) : sommets 5,6,7
  Pour les descendants, on fait un parcours en profondeur orienté ou un parcours en largeur orienté partant du sommet 4 et en s'arrêtant quand on revient à la racine : les sommets non-inclus dans cet arbre ne sont pas des descendants du sommet-racine. Si on considère qu'un sommet peut être son propre descendant, on on l'ajoute si le graphe a des cycles le contenant. Complexité en O(m) avec m le nombre d'arêtes.
- Ascendants du sommet 4 (en vert) : sommets 1,3,6,5,2
  Pour les ascendants, on inverse le sens de toutes les arêtes et procède comme pour les descendants. La complexité est la même.

Exercice 2.II.17 (*Exercice 8 (TD2)*)

Une *forêt* est une réunion disjointe d'arbres.

Question #1 : Proposer à partir du parcours en profondeur un procédé permettant de déterminer une forêt courante pour tout graphe G.

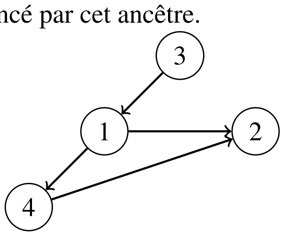
(solution page **??**)

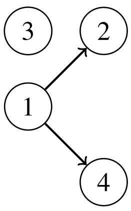
Question #2 : Dans le cas non-orienté, que peut-on dire du nombre d'arbres composant la forêt couvrante ? En déduire que la taille de la forêt ne dépend pas du sommet de départ choisi.

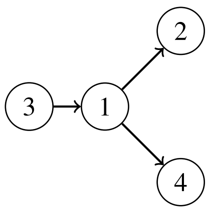
(solution page **??**)

Solution de l'exercice 2.II.18

- //TODO c.f. parcours en profondeur sur graphe orienté,option orientation conservée, ou parcours sur graphe non-connexe
- Dans le cas non-orienté, la forêt couvrante contient un arbre couvrant par composante connexe.
- Dans le cas orienté, si l'on commence par un sommet qui a un ancêtre isolé,cela mènera à au moins un arbre de plus que si l'on avait commencé par cet ancêtre.

  Par exemple, 

peut donner 

ou 

Exercice 2.II.19 (*Exercice 9 (TD2)*)

Soit v un sommet d'un graphe non-orienté connexe G et T l'arbre obtenu en appliquant un parcours en largeur depuis v. Montrer que pour tout sommet w de G, le chemin de T reliant v à w est un plus court chemin entre v et w dans G. en déduire un algorithme plus efficace que l'algorithme de Dijkstra pour calculer le diamètre d'un graphe.(solution page 51)

Solution de l'exercice 2.II.20

Soit v **un sommet d'un graphe non-orienté connexe** G **et** T **l'arbre obtenu en appliquant un parcours en largeur depuis**

v. **Montrer que pour tout sommet** w **de** G**, le chemin de** T **reliant** v **à** w **est un plus court chemin entre** v **et** w **dans** G**. en déduire un algorithme plus efficace que l'algorithme de Dijkstra pour calculer le diamètre d'un graphe.**

//TODO c.f. cours sur le plus court chemin. On lance un parcours en largeur sur chaque sommet.
Le diamètre d'un graphe est donc le plus petit nombre de niveaux de l'arbre couvrant obtenu pour un arbre connexe non-orienté.

Exercice 2.II.21 (*Exercice 10 (TD2)*)

Écrire le pseudo-code d'un algorithme qui, étant donné un graphe non-orienté G, permet soit de renvoyer un cycle de G, soit de conclure que G est acyclique.(solution page 51)

Solution de l'exercice 2.II.22

Écrire le pseudo-code d'un algorithme qui, étant donné un graphe non-orienté G**, permet soit de renvoyer un cycle de** G**, soit de conclure que** G **est acyclique.**

Il faut lancer un parcours en profondeur ou un parcours en largeur et changeant le critère d'arrêt : dès qu'un sommet a un voisin déjà visité, on s'arrête en déclarant que l'arête entre ces deux sommets fait partie d'un cycle. S'il n'y a pas de telle arête, le graphe est déclaré acyclique et l'arbre ou la forêt qui résulte du parcours en est un graphe isomorphe.

– Dans le cas d'un parcours en largeur, pour reconstituer le cycle, il faut comparer la liste des ancêtres et conserver le chemin entre chacun de ces sommets et leur premier ancêtre commun.
– Dans le cas d'un parcours en profondeur, cas le plus facile pour reconstituer le cycle, il suffit de considérer le chemin dans l'arbre partant du voisin visité et allant jusqu'à l'autre extrémité de l'arête détectée comme faisant partie d'un cycle, car tout voisin déjà visité est forcément un ancêtre dans l'arbre.

Algorithme 1 : Recherche d'un cycle de G

Input: (G,u) un graphe et l'un de ses sommets

Output: C un cycle de G ou un ensemble vide s'il n'y en a pas

```
parcoursProfondeurCycle(G,u,∅)
function parcoursProfondeurCycle(G,u,visites)
    voisins = ∅
    cycle = ∅
    visites = visites ∪ {u}
    T = {u}
    while V(T) ≠ V(G) ∧ cycle == ∅ do
        while (u; v) ∈ E(G) do
            voisins = voisins ∪ {v}
            if v ∈ visites then
                cycle = {u}
                w = u
                while w ≠ v do
                    w = pre(w)
                    cycle = cylce ∪ {w}
                end while
            else
                visites = visites ∪ {v}
                T = T ∪ (u; v)
                cycle =parcoursProfondeurCycle(G,v,visites)
            end if
        end while
    end while
    return cycle
end function
```

Exercice 2.II.23 (*Exercice 11 (TD2)*)

Reprendre l'exercice 2.I.10 dans le cas d'un graphe orienté. Déduire à partir de cet algorithme que, si tous les sommets d'un graphe orienté sont de degré sortant au moins 1, ce graphe contient un cycle orienté.(solution page 52)

Solution de l'exercice 2.II.24

Reprendre l'exercice 2.I.10 dans le cas d'un graphe orienté. Déduire à partir de cet algorithme que, si tous les sommets d'un graphe orienté sont de degré sortant au moins 1, ce graphe contient un cycle orienté.

On parcoure les arêtes selon leur orientation. On ne regarde donc que les voisins extérieurs. On remonte ensuite le chemin parcouru depuis la racine jusqu'à arriver au sommet ayant un voisin visité, plusieurs cas se présentent :

1. ce voisin est la racine et l'arête supplémentaire ajoutée au chemin parcouru est le cycle
2. ce voisin n'est pas la racine et fait partie du chemin parcouru et le morceau du chemin parcouru auquel on ajoute l'arête supplémentaire est le cycle
3. ce voisin n'est pas la racine et il ne fait pas partie du chemin parcouru : il n'y a donc pas de cycle entre ces deux sommets

Exercice 2.II.25 (*Exercice 12 (TD2)*)

Écrire le pseudo-code d'un algorithme qui, étant donné un sommet v dans un graphe orienté G, détermine un plus court cycle passant par v, en lançant deux parcours en largeur. Comment faire avec un seul parcours en largeur ?(solution page 53)

Solution de l'exercice 2.II.26

Écrire le pseudo-code d'un algorithme qui, étant donné un sommet v **dans un graphe orienté** G**, détermine un plus court cycle passant par** v**, en lançant deux parcours en largeur. Comment faire avec un seul parcours en largeur ?**

2.II.26..1 Idée

On lance un parcours en largeur partant de v dès qu'on détecte que v est le voisin de l'un de ses descendants, on a bien le plus court cycle.
Pour le faire avec deux parcours,on en lance un sur le graphe ainsi qu'un autre sur le graphe ayant les orientations des arêtes orientées, et le sommet de plus petit niveau commun aux deux parcours permet de déterminer un graphe.

2.II.26..2 Algorithme

Algorithme 2 : Recherche du plus court cycle de G contenant v

Input: (G,v) un graphe et l'un de ses sommets

Output: C un cycle de G contenant v ou un ensemble vide s'il n'y en a pas

```
function RechCourtCycle(G,v)
    C = ∅
    F = {v}
    visites = {v}
    T = {v}
    while C == ∅ ∧ F ≠ ∅ do
        w =pop(F)            # renvoie le 1er sommet de F et le retire de la file
        while u ∈ V ∧ (w; u) ∈ E do            # pour tout u voisin externe de w
            if u ∉ visites then
                visites = visites ∪ {u}
                T = T ∪ (w; u)
                append(F,u)            # ajoute u à la fin de F
            else if u == v then            # extraire le cycle en ajoutant l'arête reliant à son père et faisant de même
pour le père tant que le sommet courant n'est pas la racine
                C = (w; v)
                while (x; w) ∈ T do
                    C = C ∪ (x; w)
                    w = x            # on remonte au père dans l'arbre
                end while
            end if
        end while
    end while
    return C
end function
```

OU

Algorithme 3 : Recherche du plus court cycle de G contenant v, version du prof

Input: (G,v) un graphe et l'un de ses sommets

Output: C un cycle de G contenant v ou un ensemble vide s'il n'y en a pas

```
function PlusCourtCircuit(G,v)
    CourtCycle = ∅
    C =FALSE
    F = {v}
    T = {v}
    pour tout sommet u ≠ v
    visite(u) =FALSE
    fin pour
    visite(v) =TRUE
    while C == FALSE ∧ F ≠ ∅ do
        u = F(1)
        F = F − u
        pour tout voisin externe w de u tel que visite(w) =FALSE
        visite(w) =TRUE
        F = F + w
        T = T + (u; w)
        fin pour
        si v est un voisin externe de u
        C =TRUE
        CourtCycle = (u; v)        # extraire dans CourtCycle le chemin de v à u dansT et l'ajouter à CourtCycle
        fin si
    end while
    return CourtCycle
end function
```

2.II.26..3 Explication de la validité Soit u le premier sommet ayant v pour voisin externe. On a construit l'arbre niveau par niveau, donc à l'instant où l'on constate que v est voisin externe de u, si u est de niveau k, on a déjà parcouru tous les plus courts chemins partant de v de longueur (k − 1) Par conséquent, si u est bien le *premier* sommet rencontré qui ait v comme voisin externe, il n'y a pas de cycle contenant v de longueur inférieure ou égale à k. En revanche, il se peut qu'il y ait d'autres cycles de longueur (k + 1) contenant v dans le graphe : en passant par des sommets de niveau k autres que u mais qui auraient été considérés après u dans l'arbre de parcours du parcours en largeur final/total.

Exercice 2.II.27 (*Exercice 13 (TD2)*)

Déterminer un arbre couvrant de poids minimal pour le graphe de la figure suivante (2.8).

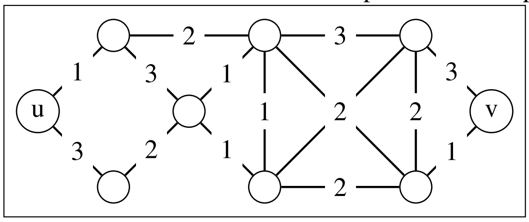


FIGURE 2.8

(solution page 56)

Solution de l'exercice 2.II.28

Déterminer un arbre couvrant de poids minimal pour le graphe de la figure suivante (2.8).

Avec l'algorithme de Kruskal : liste des arêtes avec leur poids, triées par poids croissant, parcourue dans l'ordre et où les arêtes conservées sont soulignées :

$\{\underline{AB(1)}, \underline{DE(1)}, \underline{DF(1)}, EF(1), \underline{HI(1)}, \underline{BE(2)}, \underline{CD(2)}, \underline{FH(2)}, \underline{GH(2)}, FG(2), BD(3), GI(3), EG(3), AC(3), EH(4)\}$

//TODO : graphe

Avec l'algorithme de Primm : liste des arêtes avec leur poids, triées par poids décroissant, parcourue dans l'ordre et où les arêtes conservées sont soulignées :

$\{EH(4), BD(3), GI(3), EG(3), AC(3), \underline{BE(2)}, \underline{CD(2)}, \underline{FH(2)}, \underline{GH(2)}, FG(2), \underline{AB(1)}, \underline{DE(1)}, \underline{DF(1)}, EF(1), \underline{HI(1)}\}$

//TODO : graphe

Exercice 2.II.29 (*Exercice 14 (TD2)*)

Les graphes suivants contiennent-ils un chemin eulérien, un cycle eulérien, un chemin hamiltonien, un cycle hamiltonien ?

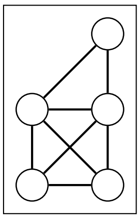
1. G_1 

FIGURE 2.9

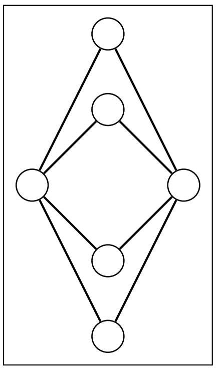
2. G_2 

FIGURE 2.10

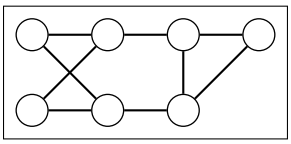
3. G_3 

FIGURE 2.11

Solution de l'exercice 2.II.30

cycle eulérien :

1. G_1 n'a pas de cycle eulérien car le graphe contient des sommets de degré impair
2. G_2 a pas un cycle eulérien car tous les sommets du graphe sont de degré pair

3. G_3 n'a pas de cycle eulérien car le graphe contient des sommets de degré impair

chemin eulérien :

1. G_1 a un chemin eulérien car il y seulement 2 sommets de degré impair : on dédouble l'arête entre ces 2 sommets et le nouveau graphe admet un cycle eulérien.Le chemin désiré commence en l'un de ces 2 sommets de degré impair et se termine en l'autre.
2. G_2 a un chemin eulérien car il a un cycle eulérien
3. //TODO

cycle hamiltonien :

1. //TODO
2. //TODO
3. //TODO

chemin hamiltonien :

1. //TODO
2. //TODO
3. //TODO

Exercice 2.II.31 (*Exercice 15 (TD2)*)

Résoudre le problème du postier sur le graphe de la figure suivante (2.12).

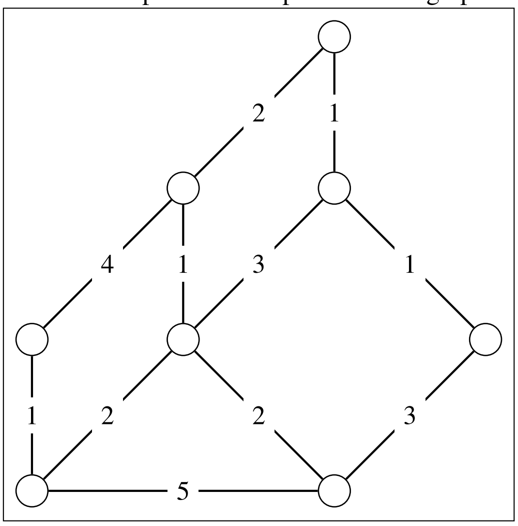


FIGURE 2.12

(solution page 57)

Solution de l'exercice 2.II.32

Résoudre le problème du postier sur le graphe de la figure suivante (2.12).

Exercice 2.II.33 (*Exercice 16 (TD2)*)

Pour résoudre le problème du voyageur de commerce, on propose l'heuristique suivante :

– construire un arbre couvrant de poids minimal
– parcourir les sommets de cet arbre comme un parcours en profondeur

Question #1 : Quelle est la complexité de cet algorithme ?

(solution page 58)

Question #2 : Soit H le graphe induit par les arêtes utilisées par un parcours optimal. Montrer que H contient un arbre couvrant. En déduire que la solution proposée par l'heuristique est une 2-approximation, c'est-à-dire qu'elle est au pire de poids deux fois plus important que la solution optimale.

(solution page 58)

Solution de l'exercice 2.II.34

Pour résoudre le problème du voyageur de commerce, on propose l'heuristique suivante :
- **construire un arbre couvrant de poids minimal**
- **parcourir les sommets de cet arbre comme un parcours en profondeur**

1. **Quelle est la complexité de cet algorithme ?**

2. **Soit** H **le graphe induit par les arêtes utilisées par un parcours optimal. Montrer que** H **contient un arbre couvrant. En déduire que la solution proposée par l'heuristique est une 2-approximation, c'est-à-dire qu'elle est au pire de poids deux fois plus important que la solution optimale.**

2.III Ce qui a été montré

- Un arbre ayant un sommet de degré au moins k a au moins k feuilles
- Un arbre a un centre unique ou deux centres adjacents
- Une arête appartient à tous les arbres couvrants si et seulement si elle est un pont
- Une forêt couvrante d'un graphe non-orienté contient un arbre couvrant par composante connexe
- Une forêt couvrante d'un graphe orienté peut contenir un nombre différent d'arbres selon le sommet de départ
- Dans un arbre couvrant obtenu grâce à un parcours en largeur, le chemin reliant tout sommet à la racine est le plus court chemin entre ces deux sommets.
- Si tous les sommets d'un graphe orienté sont de degré sortant au moins 1, ce graphe contient un cycle orienté

TD 3 :

Sommaire

3.I Énoncés

3.I.1 Exercice 1

(voir la version "exercice résolu" page 65)

Soit le graphe 3.5 suivant :

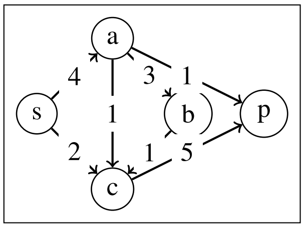


FIGURE 3.5

3.I.1.a Question 1

Déterminer tous les flots entiers du graphe précédent. En déduire un flot maximal.

3.I.1.b Question 2

Déterminer toutes les coupes du graphe précédent. En déduire une coupe minimale.

3.I.2 Exercice 2

(voir la version "exercice résolu" page 68)

3.I.2.a Question 1

Soit f un flot. Montrer que pour tout ensemble A de sommets, $\sum_{v \in A}\left(f^+(v) - f^-(v)\right) = f^+(A) - f^-(A)$. En déduire que pour toute coupe $\left(A, \overline{A}\right)$, on a $f^+(A) - f^-(A) = val(f)$

3.I.3 Exercice 3

(voir la version "exercice résolu" page 68)

Soient $\left(A, \overline{A}\right)$ et $\left(B, \overline{B}\right)$ deux coupes minimales. Montrer que $\left(A \cup B, \overline{A \cup B}\right)$ est aussi une coupe minimale.

3.I.4 Exercice 4

(voir la version "exercice résolu" page 70)

Les produits des usines u_1 et u_2 doivent être acheminés vers les ports $p_{1,2}$ et p_3. Le nombre de tonnes pouvant être acheminé quotidiennement sont indiqués sur les routes de la figure 3.7 suivante :

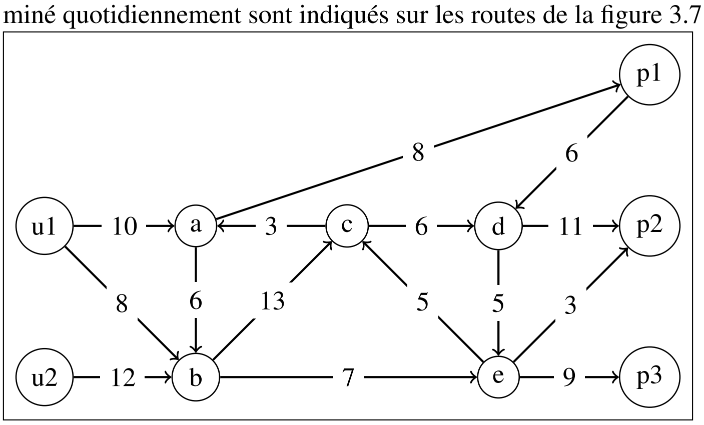


FIGURE3.7

3.I.4.a Question 1

Quelle est la quantité maximale de produits pouvant être acheminés ?

3.I.4.b Question 2

Des crédits d'infrastructure sont disponibles. Sur quelles routes faut-il investir pour augmenter la capacité d'acheminement ?

3.I.5 Exercice 5

(voir la version "exercice résolu" page 74)

On considère un graphe orienté valué par des capacités, ayant un ensemble de sources S et de puits P. À chaque source $s_i \in S$ correspond une capacité d'émission $\sigma(s_i)$ et à chaque puit $p_i \in P$ correspond une demande de réception $\delta(p_i)$. Un flot est dit *admissible* si $\forall\ s_i \in S,\ f^+s_i - f^-s_i \leqslant \sigma(s_i)$ et $\forall\ p_i \in P,\ f^+p_i - f^-p_i \geqslant \delta(p_i)$.
On construit un graphe H obtenu en ajoutant

- une source s_0, et pour tout sommet s_i, l'arête (s_0, s_i) avec une capacité $\sigma(s_i)$
- un puit p_0, et pour tout sommet p_i, l'arête (p_i, p_0) avec une capacité $\delta(p_i)$

3.I.5.a Question 1

Construire le graphe H correspondant au graphe G de la figure 3.7 avec $\sigma(u_1) = 11, \sigma(u_2) = 11, \delta(p_1) = 6, \delta(p_2) = 8$ et $\delta(p_3) = 7$.

3.I.5.b Question 2

Montrer qu'un flot admissible existe dans G si et seulement s'il existe un flot dans H qui sature toutes les arêtes de la forme (p_1, p_0).

3.I.5.c Question 3

Montrer qu'un tel flot est forcément maximal dans H.

3.I.5.d Question 4

Existe-t-il un flot admissible dans le graphe de la figure 3.7 pour les valeurs d'émission et de réception données à la question 1 ?

3.I.6 Exercice 6

(voir la version "exercice résolu" page 75)

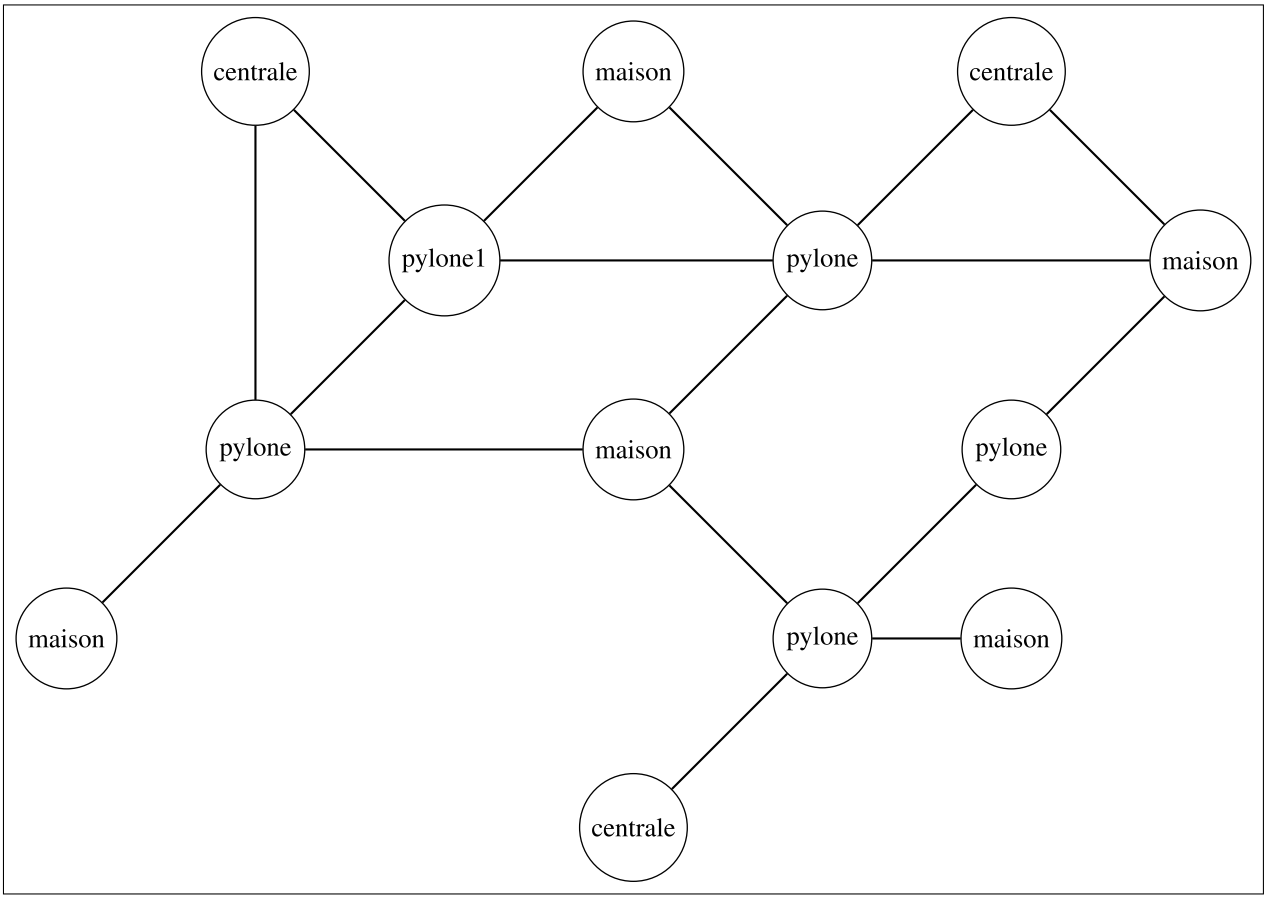

centrale
maison
centrale
pylone1
pylone
maison
pylone
maison
pylone
maison
pylone
maison
centrale

FIGURE3.3

Le graphe 3.3 ci-dessus représente un réseau électrique. Quel est le nombre minimum de rupture de liaisons nécessaires pour qu'aucune maison ne soit plus alimentée, chaque centrale étant supposée assez puissante pour alimenter toutes les maisons ?

3.I.7 Exercice 7

(voir la version "exercice résolu" page 76)

Soit G un réseau de sources s et de puits t, à capacités entières. La capacité maximum sur l'ensemble du réseau est notée $C = \max_e c(e)$. On note m le nombre d'arêtes du réseau. On considère l'algorithme suivant, dit de flot maximal par échelonnement :

ALGORITHME 4 : Recherche du flot maximal par échelonnement

Input: $(G = (V, E, c))$ un réseau

Output: f Le flot maximal

1: **function** RECHFLOTMAXECHELONNEMENT(G)
2:     $C = \max_{e \in E(G)} c(e)$
3:     $f = \emptyset$
4:     $K = 2^{\lfloor \log_2(C) \rfloor}$
5:     **while** $K \geqslant 1$ **do**
6:         **while** il existe un chemin non saturé P de capacité résiduelle supérieure ou égale à K **do**
7:             augmenter f le long de P
8:         **end while**
9:         $K = K/2$
10:     **end while**
11:     **return** f
12: **end function**

3.I.7.a Question 1

Montrer qu'une coupe de G a une capacité au plus égale à C_m.

3.I.7.b Question 2

Pour un nombre K donné, montrer qu'un chemin non-saturé améliorant la capacité au moins égale à K peut être trouvé en temps O(m), si un tel chemin existe.

3.I.7.c Question 3

Montrer que l'algorithme par échelonnement calcule bien un flot maximal et l'appliquer au graphe de la figure 3.7.

3.I.7.d Question 4

Montrer que la capacité d'une coupe minimum du graphe résiduel vaut au plus 2mK à chaque exécution de la ligne 4.

3.I.7.e Question 5

Montrer que la ligne 6 est exécutée O(m) fois pour chaque valeur de K.

3.I.7.f Question 6

En déduire une borne de complexité pour cet algorithme.

3.II Exercices résolus

Exercice 3.II.1 (*Exercice 1 (TD3)*)

Soit le graphe 3.5 suivant :

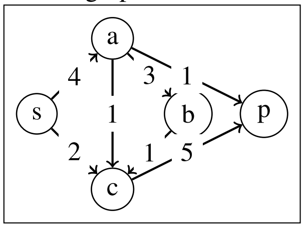


FIGURE3.5

Question #1 : Déterminer tous les flots entiers du graphe précédent. En déduire un flot maximal.

(solution page 67)

Question #2 : Déterminer toutes les coupes du graphe précédent. En déduire une coupe minimale.

(solution page 68)

Solution de l'exercice 3.II.2

Soit le graphe 3.5 suivant :

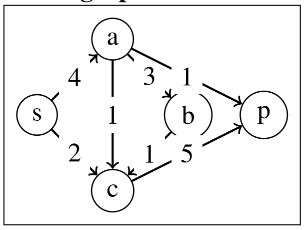


FIGURE3.5

1. **Déterminer tous les flots entiers du graphe précédent. En déduire un flot maximal.**

	(s,a)	(s,c)	(a,b)→(b,c)	(a,c)	(c,p)	(a,p)	val(f)
	0	0	0 → 0	0	0	0	0
(s,a)=0	0	1	0 → 0	0	1	0	1
	0	2	0 → 0	0	2	0	2
	1	0	0 → 0	0	0	1	1
	1	1	0 → 0	0	1	1	2
	1	2	0 → 0	0	2	1	3
	1	0	0 → 0	1	1	0	1
(s,a)=1	1	1	0 → 0	1	2	0	2
	1	2	0 → 0	1	3	0	3
	1	0	1 → 1	0	1	0	1
	1	1	1 → 1	0	2	0	2
	1	2	1 → 1	0	3	0	3
	2	0	0 → 0	1	1	1	2
	2	0	1 → 1	0	1	1	2
	2	0	1 → 1	1	2	0	2
	2	1	0 → 0	1	2	1	3
(s,a)=2	2	1	1 → 1	0	2	1	3
	2	1	1 → 1	1	3	0	3
	2	2	0 → 0	1	3	1	4
	2	2	1 → 1	0	3	1	4
	2	2	1 → 1	1	4	0	4
	3	0	1 → 1	1	2	1	3
(s,a)=3	3	1	1 → 1	1	3	1	4
	3	2	1 → 1	1	4	1	5

Le flot maximal est d'intensité 5.

2. **Déterminer toutes les coupes du graphe précédent. En déduire une coupe minimale.**

L'ensemble des coupes est $\{(A, \overline{A}) / A \in \{\{s\} \cup \{s;a\} \cup \{s;b\} \cup \{s;c\} \cup \{s;a;b\} \cup \{s;a;c\} \cup \{s;b;c\} \cup \{s;a;b;c\}\}$.

- $K(\{s\}, \{a;b;c;p\}) = \{(s,a);(s,c)\} \implies cap(K) = 6$
- $K(\{s;a\}, \{b;c;p\}) = \{(s,c);(a,p);(a,c);(a,b)\} \implies cap(K) = 7$
- $K(\{s;b\}, \{a;c;p\}) = \{(s,a);(s,c);(b,c)\} \implies cap(K) = 7$
- $K(\{s;c\}, \{a;b;p\}) = \{(s,a);(c,p)\} \implies cap(K) = 9$
- $K(\{s;a;b\}, \{c;p\}) = \{(s,c);(a,c);(b,c);(a,p)\} \implies cap(K) = 5$
- $K(\{s;a;c\}, \{b;p\}) = \{(a,b);(a,p);(c,p)\} \implies cap(K) = 9$
- $K(\{s;b;c\}, \{a;p\}) = \{(s,a);(c,p)\} \implies cap(K) = 9$
- $K(\{s;a;b;c\}, \{p\}) = \{(a,p);(c,p)\} \implies cap(K) = 6$

La coupe minimale est de capacité 5.

Exercice 3.II.3 (*Exercice 2 (TD3)*)

Soit f un flot. Montrer que pour tout ensemble A de sommets, $\sum_{v \in A}\left(f^+(v) - f^-(v)\right) = f^+(A) - f^-(A)$. En déduire que pour toute coupe $(A, \overline{A})$, on a $f^+(A) - f^-(A) = val(f)$(solution page 68)

Solution de l'exercice 3.II.4

Soit f **un flot. Montrer que pour tout ensemble** A **de sommets,** $\sum_{v \in A}\left(f^+(v) - f^-(v)\right) = f^+(A) - f^-(A)$**. En déduire que pour toute coupe** $(A, \overline{A})$**, on a** $f^+(A) - f^-(A) = val(f)$

$$\sum_{v \in A} f^-(v) = \sum_{v \in A}\{\sum_{\substack{e=(u,v)\\u \in A}} f(e) + \sum_{\substack{e=(u,v)\\u \in \overline{A}}} f(e)\}$$

$$\sum_{v \in A} f^+(v) = \sum_{v \in A}\{\sum_{\substack{e=(v,u)\\u \in A}} f(e) + \sum_{\substack{e=(v,u)\\u \in \overline{A}}} f(e)\} = \sum_{v \in A}\{\sum_{\substack{e=(u,v)\\u \in A}} f(e) + \sum_{\substack{e=(v,u)\\u \in \overline{A}}} f(e)\}$$

$$\implies \sum_{v \in A}\{f^+(v) - f^-(v)\} = \sum_{v \in A}\{\sum_{\substack{e=(v,u)\\u \in \overline{A}}} f(e) - \sum_{\substack{e=(u,v)\\u \in \overline{A}}} f(e)\}$$

$$\implies \sum_{v \in A}\{f^+(v) - f^-(v)\} = \sum_{e \in (A,\overline{A})} f(e) - \sum_{e \in (\overline{A},A)} f(e) = f^+(A) - f^-(A)$$

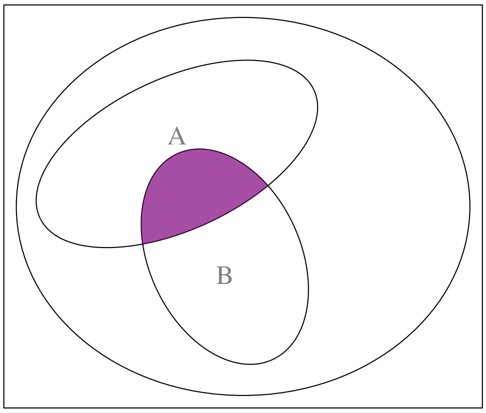
De façon plus littéraire, notons $e_0 = (u, w)$ une arête interne à A. Dans $\sum_{v \in A}\{f^+(v) - f^-(v)\}$, $f(e_0)$ apparaît dans $f^+(u)$ et dans $f^-(w)$ qui se simplifient finalement dans cette somme. En revanche, les arêtes entre A et $\overline{A}$ n'apparaissent qu'une seule fois dans cette somme, en positif ou en négatif selon qu'elles entrent ou sortent de A.

Exercice 3.II.5 (*Exercice 3 (TD3)*)

Soient $(A, \overline{A})$ et $(B, \overline{B})$ deux coupes minimales. Montrer que $(A \cup B, \overline{A \cup B})$ est aussi une coupe minimale.(solution page 68)

Solution de l'exercice 3.II.6

Soient $(A, \overline{A})$ **et** $(B, \overline{B})$ **deux coupes minimales. Montrer que** $(A \cup B, \overline{A \cup B})$ **est aussi une coupe minimale.**



Exemple :

On obtient :

– $\mathrm{cap}\left(A, \bar{A}\right) = \mathrm{cap}\left(A \cap B, \overline{A \cup B}\right) + \mathrm{cap}\left(A \setminus B, \overline{A \cup B}\right) + \mathrm{cap}(A \cap B, B \setminus A) + \mathrm{cap}(A \setminus B, B \setminus A)$

– $\mathrm{cap}\left(B, \bar{B}\right) = \mathrm{cap}\left(A \cap B, \overline{A \cup B}\right) + \mathrm{cap}\left(B \setminus A, \overline{A \cup B}\right) + \mathrm{cap}(A \cap B, A \setminus B) + \mathrm{cap}(B \setminus A, A \setminus B)$

– $\mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) = \mathrm{cap}\left(A \cap B, \overline{A \cup B}\right) + \mathrm{cap}\left(A \setminus B, \overline{A \cup B}\right) + \mathrm{cap}\left(B \setminus A, \overline{A \cup B}\right)$

– $\mathrm{cap}\left(A \cap B, \overline{A \cap B}\right) = \mathrm{cap}\left(A \cap B, \overline{A \cup B}\right) + \mathrm{cap}(A \cap B, B \setminus A) + \mathrm{cap}(A \cap B, A \setminus B)$

D'où $\mathrm{cap}\left(A, \bar{A}\right) + \mathrm{cap}\left(B, \bar{B}\right) = \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) + \mathrm{cap}\left(A \cap B, \overline{A \cap B}\right) + \mathrm{cap}(A \setminus B, B \setminus A) + \mathrm{cap}(B \setminus A, A \setminus B)$.

Notons $c_{\min} = \mathrm{cap}\left(A, \bar{A}\right)$.

Comme $\left(A, \bar{A}\right)$ et $\left(B, \bar{B}\right)$ sont des coupes minimales, on a aussi : $\begin{cases} \mathrm{cap}\left(B, \bar{B}\right) = c_{\min} \\ \mathrm{cap}\left(A \cap B, \overline{A \cap B}\right) \geqslant c_{\min} \\ \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) \geqslant c_{\min} \end{cases}$

De plus, $\mathrm{cap}(A \setminus B, B \setminus A) \geqslant 0$ et $\mathrm{cap}(B \setminus A, A \setminus B) \geqslant 0$.

$\Longrightarrow\ 2c_{\min} = \mathrm{cap}\left(A, \bar{A}\right) + \mathrm{cap}\left(B, \bar{B}\right) = \underbrace{\mathrm{cap}\left(A \cup B, \overline{A \cup B}\right)}_{\geqslant c_{\min}} + \underbrace{\mathrm{cap}\left(A \cap B, \overline{A \cap B}\right)}_{\geqslant c_{\min}} + \underbrace{\mathrm{cap}(A \setminus B, B \setminus A)}_{\geqslant 0} + \underbrace{\mathrm{cap}(B \setminus A, A \setminus B)}_{\geqslant 0}$

$\Longrightarrow\ 2c_{\min} \leqslant= \mathrm{cap}\left(A \cap B, \overline{A \cap B}\right) + \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) = \mathrm{cap}\left(A, \bar{A}\right) + \mathrm{cap}\left(B, \bar{B}\right) - (\mathrm{cap}(B \setminus A, A \setminus B) + \mathrm{cap}(A \setminus B, B \setminus A)) \leqslant 2c_{\min}$

$\Longrightarrow\ \mathrm{cap}\left(A \cap B, \overline{A \cap B}\right) + \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) = 2c_{\min}$

$\Longrightarrow\ \mathrm{cap}\left(A \cap B, \overline{A \cap B}\right) + \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) \leqslant \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) + \mathrm{cap}\left(B, \bar{B}\right) \leqslant \mathrm{cap}\left(A, \bar{A}\right) + \mathrm{cap}\left(B, \bar{B}\right)$

$\Longrightarrow\ \mathrm{cap}\left(A, \bar{A}\right) \geqslant \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right)$

$\Longrightarrow\ \mathrm{cap}\left(A, \bar{A}\right) = \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right)$ car $\left(A, \bar{A}\right)$ est minimale

$\Longrightarrow\ \mathrm{cap}\left(A \cup B, \overline{A \cup B}\right) = c_{\min} = \mathrm{cap}\left(A \cap B, \overline{A \cap B}\right)$

Donc $\mathrm{cap}\left(A \cap B, \overline{A \cap B}\right)$ est minimale.

Exercice 3.II.7 (*Exercice 4 (TD3)*)

Les produits des usines u_1 et u_2 doivent être acheminés vers les ports $p_{1,2}$ et p_3. Le nombre de tonnes pouvant être acheminé quotidiennement sont indiqués sur les routes de la figure 3.7 suivante :

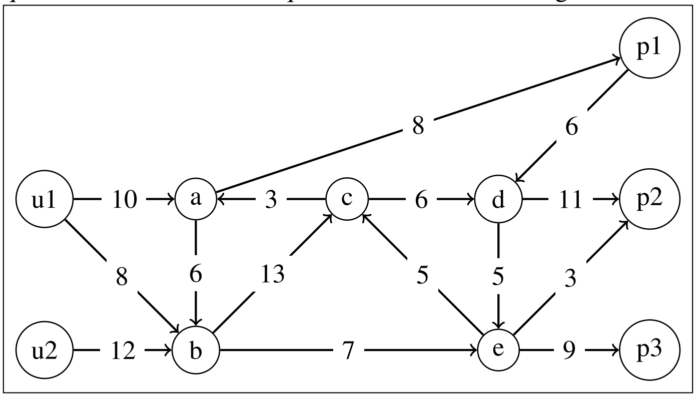


FIGURE3.7

Question #1 : Quelle est la quantité maximale de produits pouvant être acheminés ?

(solution page 73)

Question #2 : Des crédits d'infrastructure sont disponibles. Sur quelles routes faut-il investir pour augmenter la capacité d'acheminement ?

(solution page 74)

Solution de l'exercice 3.II.8

Les produits des usines u_1 et u_2 doivent être acheminés vers les ports p_1,$_2$ et p_3. Le nombre de tonnes pouvant être acheminé quotidiennement sont indiqués sur les routes de la figure 3.7 suivante :

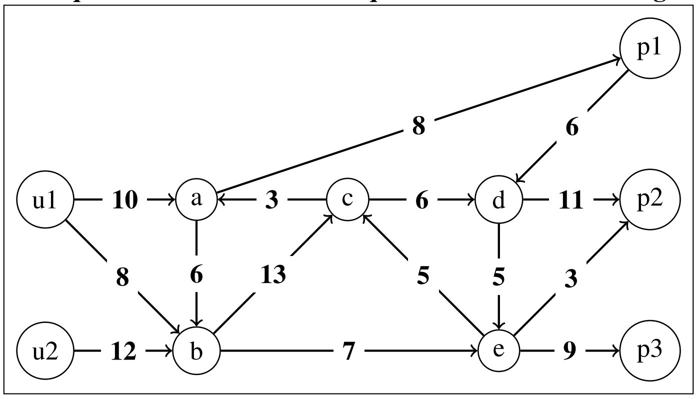
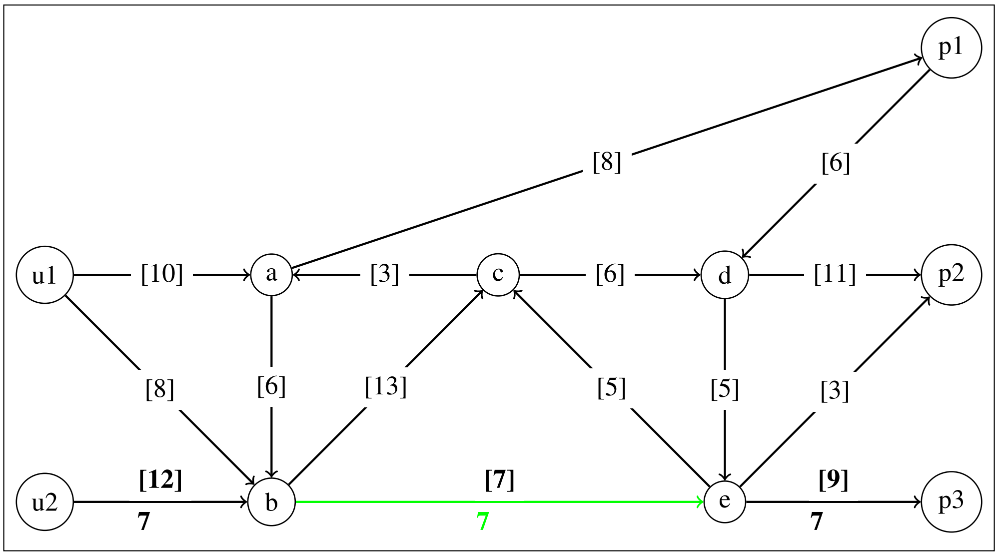


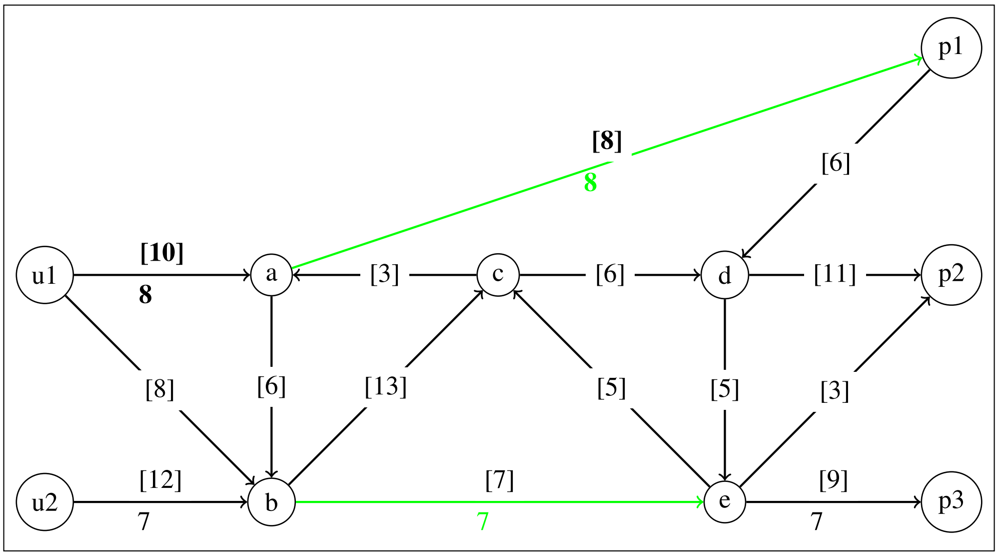
FIGURE3.7

1. **Quelle est la quantité maximale de produits pouvant être acheminés ?**

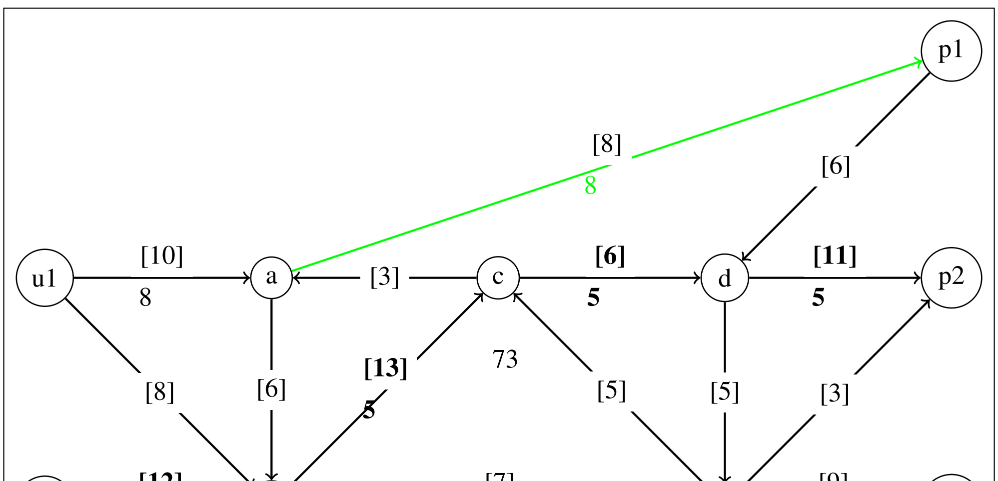
étape 1 :



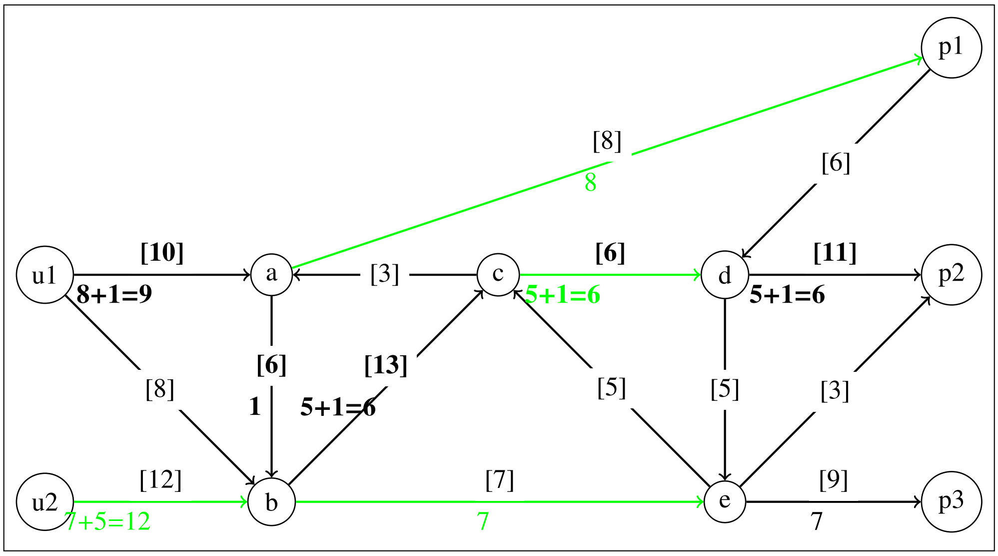
étape 2 :



étape 3 :



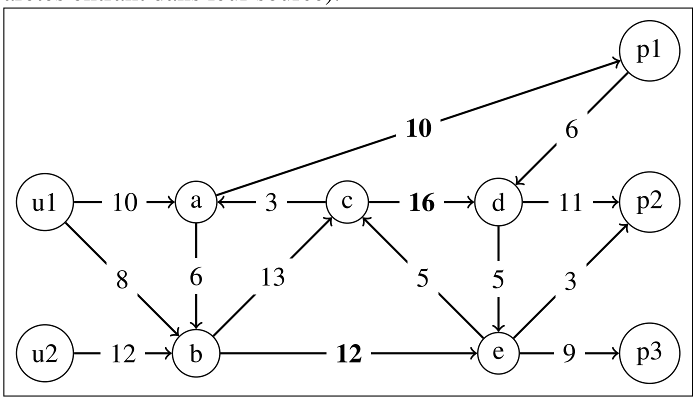
étape 4 :



val(f) = 8 + 6 + 7 = 12 + 9 = 21

2. **Des crédits d'infrastructure sont disponibles. Sur quelles routes faut-il investir pour augmenter la capacité d'acheminement ?**

   Les arêtes saturées non adjacentes à une source forment une coupe. C'est de ces arêtes qu'il faut d'abord augmenter la capacité (jusqu'au minimum entre la somme des capacités des arêtes sortant de leur cible et la somme des capacités des arêtes entrant dans leur source).



Exercice 3.II.9 (*Exercice 5 (TD3)*)

On considère un graphe orienté valué par des capacités, ayant un ensemble de sources S et de puits P. À chaque source $s_i \in S$ correspond une capacité d'émission $\sigma(s_i)$ et à chaque puit $p_i \in P$ correspond une demande de réception $\delta(p_i)$. Un flot est dit *admissible* si $\forall\ s_i \in S,\ f^+s_i - f^-s_i \leqslant \sigma(s_i)$ et $\forall\ p_i \in P,\ f^+p_i - f^-p_i \geqslant \delta(p_i)$.
On construit un graphe H obtenu en ajoutant

– une source s_0, et pour tout sommet s_i, l'arête (s_0, s_i) avec une capacité $\sigma(s_i)$
– un puit p_0, et pour tout sommet p_i, l'arête (p_i, p_0) avec une capacité $\delta(p_i)$

Question #1 : Construire le graphe H correspondant au graphe G de la figure 3.7 avec $\sigma(u_1) = 11, \sigma(u_2) = 11, \delta(p_1) = 6, \delta(p_2) = 8$ et $\delta(p_3) = 7$.

(solution page 75)

Question #2 : Montrer qu'un flot admissible existe dans G si et seulement s'il existe un flot dans H qui sature toutes les arêtes de la forme (p_1, p_0).

(solution page 75)

Question #3 : Montrer qu'un tel flot est forcément maximal dans H.

(solution page 75)

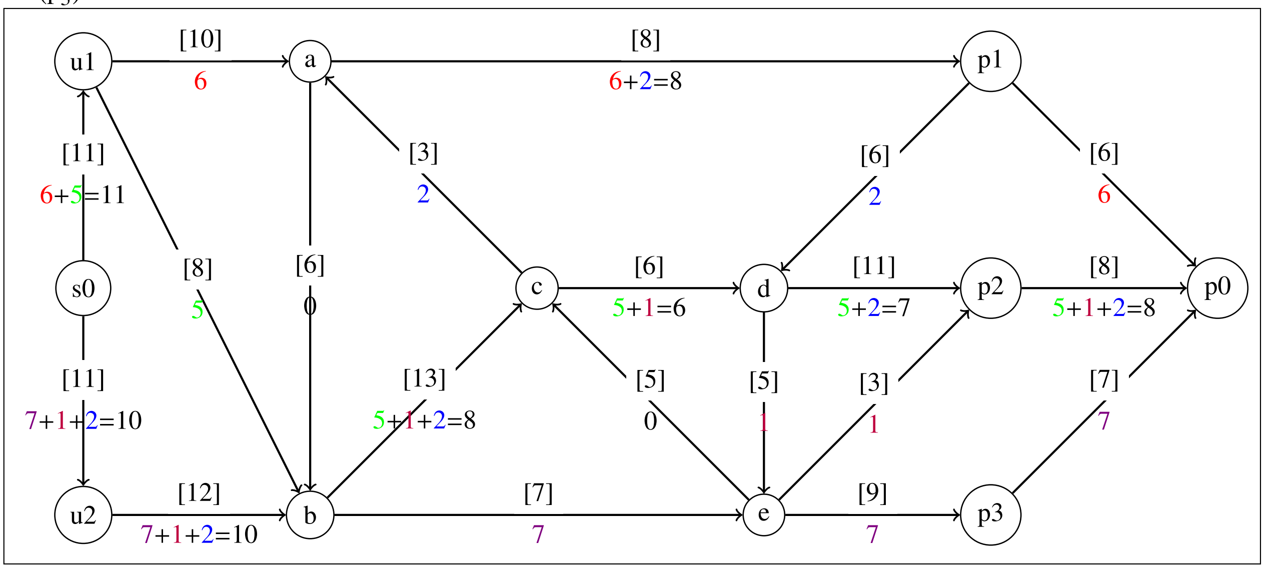
Question #4 : Existe-t-il un flot admissible dans le graphe de la figure 3.7 pour les valeurs d'émission et de réception données à la question 1 ?

(solution page 75)

Solution de l'exercice 3.II.10

On considère un graphe orienté valué par des capacités, ayant un ensemble de sources S **et de puits** P**. À chaque source** $s_i \in S$ **correspond une capacité d'émission** $\sigma(s_i)$ **et à chaque puit** $p_i \in P$ **correspond une demande de réception** $\delta(p_i)$**. Un flot est dit *admissible* si** $\forall\ s_i \in S,\ f^+s_i - f^-s_i \leqslant \sigma(s_i)$ **et** $\forall\ p_i \in P,\ f^+p_i - f^-p_i \geqslant \delta(p_i)$**.**
On construit un graphe H **obtenu en ajoutant**
- **une source** s_0**, et pour tout sommet** s_i**, l'arête** (s_0, s_i) **avec une capacité** $\sigma(s_i)$
- **un puit** p_0**, et pour tout sommet** p_i**, l'arête** (p_i, p_0) **avec une capacité** $\delta(p_i)$

1. **Construire le graphe** H **correspondant au graphe** G **de la figure 3.7 avec** $\sigma(u_1) = 11, \sigma(u_2) = 11, \delta(p_1) = 6, \delta(p_2) = 8$ **et** $\delta(p_3) = 7$**.**



2. **Montrer qu'un flot admissible existe dans** G **si et seulement s'il existe un flot dans** H **qui sature toutes les arêtes de la forme** (p_1, p_0)**.**
   Soit f un flot dans H qui sature les arêtes de la forme (p_i, p_0). Considérons le flot f' obtenu en restreignant f aux arêtes de G (les arêtes de la forme (p_i, p_0) et (s_0, s_i) n'appartiennent donc pas au flot f').
   On constate que pour toute source s_i, le flot envoyé par f' vaut ce que s_i reçoit de s_0 par le flot f. Concrètement, on a $\forall i\ f'^+(s_i) - f'^-s_i = f(s_0, s_i) = \delta(s_i)$.
   Similairement, pour tout puit p_i, //TODO finir
3. **Montrer qu'un tel flot est forcément maximal dans** H**.**

4. **Existe-t-il un flot admissible dans le graphe de la figure 3.7 pour les valeurs d'émission et de réception données à la question 1 ?**

Exercice 3.II.11 (*Exercice 6 (TD3)*)

Le graphe 3.3 ci-dessus représente un réseau électrique. Quel est le nombre minimum de rupture de liaisons nécessaires pour qu'aucune maison ne soit plus alimentée, chaque centrale étant supposée assez puissante pour alimenter toutes les maisons ?(solution page 76)

Solution de l'exercice 3.II.12

Le graphe 3.3 ci-dessus représente un réseau électrique. Quel est le nombre minimum de rupture de liaisons nécessaires pour qu'aucune maison ne soit plus alimentée, chaque centrale étant supposée assez puissante pour alimenter toutes les maisons ?

Exercice 3.II.13 (*Exercice 7 (TD3)*)

Soit G un réseau de sources s et de puits t, à capacités entières. La capacité maximum sur l'ensemble du réseau est notée $C = \max_e c(e)$. On note m le nombre d'arêtes du réseau. On considère l'algorithme suivant, dit de flot maximal par échelonnement :

ALGORITHME 5 : Recherche du flot maximal par échelonnement

Input: $(G = (V, E, c))$ un réseau

Output: f Le flot maximal

```
1:  function RechFlotMaxEchelonnement(G)
2:      C = max_{e ∈ E(G)} c(e)
3:      f = ∅
4:      K = 2^⌊log2(C)⌋
5:      while K ⩾ 1 do
6:          while il existe un chemin non saturé P de capacité résiduelle supérieure ou égale à K do
7:              augmenter f le long de P
8:          end while
9:          K = K/2
10:     end while
11:     return f
12: end function
```

Question #1 : Montrer qu'une coupe de G a une capacité au plus égale à C_m.

(solution page 78)

Question #2 : Pour un nombre K donné, montrer qu'un chemin non-saturé améliorant la capacité au moins égale à K peut être trouvé en temps O(m), si un tel chemin existe.

(solution page 78)

Question #3 : Montrer que l'algorithme par échelonnement calcule bien un flot maximal et l'appliquer au graphe de la figure 3.7.

(solution page 78)

Question #4 : Montrer que la capacité d'une coupe minimum du graphe résiduel vaut au plus 2mK à chaque exécution de la ligne 4.

(solution page 79)

Question #5 : Montrer que la ligne 6 est exécutée O(m) fois pour chaque valeur de K.

(solution page 79)

Question #6 : En déduire une borne de complexité pour cet algorithme.

(solution page 79)

Solution de l'exercice 3.II.14

Soit G **un réseau de sources** s **et de puits** t**, à capacités entières. La capacité maximum sur l'ensemble du réseau est notée** $C = \max_e c(e)$**. On note** m **le nombre d'arêtes du réseau. On considère l'algorithme suivant, dit de flot maximal par échelonnement :**

ALGORITHME 6 : Recherche du flot maximal par échelonnement

Input: $(G = (V, E, c))$ **un réseau**

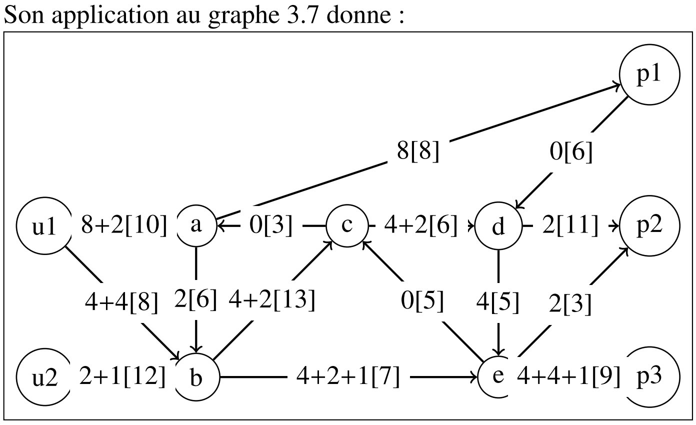
Output: f **Le flot maximal**

```
function RechFlotMaxEchelonnement(G)
    C = max_{e ∈ E(G)} c(e)
    f = ∅
    K = 2^⌊log2(C)⌋
    while K ⩾ 1 do
        while il existe un chemin non saturé P de capacité résiduelle supérieure ou égale à K do
            augmenter f le long de P
        end while
        K = K/2
    end while
    return f
end function
```

1. **Montrer qu'une coupe de** G **a une capacité au plus égale à** C_m**.**

2. **Pour un nombre** K **donné, montrer qu'un chemin non-saturé améliorant la capacité au moins égale à** K **peut être trouvé en temps** $O(m)$**, si un tel chemin existe.**

3. **Montrer que l'algorithme par échelonnement calcule bien un flot maximal et l'appliquer au graphe de la figure 3.7.**
   À la dernière étape, $K = 1$, et à ce moment-là,l'algorithme redevient l'algorithme "habituel" (vu en cours). On trouve donc bien un flot maximal.
   Son application au graphe 3.7 donne :



On cherche le nombre maximum de chemins qui peuvent être trouvés pour un K donné (pour une puissance de 2 donnée,sachant qu'il n'y a pas de chemin des sources vers les puits qui soit de capacité supérieure à cette puissance de 2 dans le graphe résiduel).

$c = 13 \in]2^3 ; 2^4[\implies \lfloor \log_2(c) \rfloor = 3$

étape	K	chemins ajoutés
1	$8 = 2^3$	$u_1 \to a \to p_1$
2	$4 = 2^2$	$u_1 \to b \to e \to p_3$ et $u_1 \to b \to c \to d \to e \to p_3$
3	$2 = 2^1$	$u_1 \to b \to c \to d \to p_2$ et $u_1 \to a \to b \to e \to p_3$
4	$1 = 2^0$	$u_2 \to b \to e \to p_3$

On retrouve bien $8 * 1 + 4 * 2 + 2 * 2 + 1 * 1 = 21$

4. **Montrer que la capacité d'une coupe minimum du graphe résiduel vaut au plus** $2mK$ **à chaque exécution de la ligne** 4**.**

   Lorsqu'on applique la ligne 4, il n'y a plus de chemins des sources vers les puits qui augmente le flot d'au moins 2K.
   il existe alors une coupe entre A, l'ensemble des sommets vers lesquels il existe un chemin depuis l'une des sources qui soit de capacité résiduelle supérieure ou égale à 2K, et son complémentaire $\overline{A}$.
   Alors les puits sont éléments de $\overline{A}$ et toute arête reliant un sommet de a à son complémentaire est de capacité résiduelle strictement inférieure à 2K. En effet, si (u, v) est une telle arête et qu'elle est de capacité résiduelle supérieure ou égale à 2K,alors $v \in A$.
   En conclusion, $\left(A, \overline{A}\right)$ est une coupe d'arêtes toutes de capacité résiduelle est d'au plus 2K, ce qui fait que sa capacité est d'au plus 2mK.

5. **Montrer que la ligne** 6 **est exécutée** $O(m)$ **fois pour chaque valeur de** K**.**

   D'une part, la capacité minimale d'une coupe est inférieure ou égale à 2mK. D'autre part, chaque chemin trouvé par l'algorithme augmente le flot d'une valeur supérieure ou égale à K et strictement inférieure à 2K.
   Comme la capacité de la coupe minimale vaut l'intensité du flot maximal, on trouve au plus $\frac{2mK}{K} = 2m$ chemins. Par conséquent, le nombre d'exécutions de la ligne 6 à K fixé est en $O(m)$

6. **En déduire une borne de complexité pour cet algorithme.**

   On essaie $1 + \lfloor \log_2(c) \rfloor$ valeurs de K car le couple ligne 4/ligne 7 est exécuté $1 + \lfloor \log_2(c) \rfloor$ fois, de $2^{\lfloor \log_2(c) \rfloor}$ à 2^0.
   Finalement, la complexité de l'algorithme est de $O\left(m^2 \log_2(c)\right)$.

3.III Ce qui a été montré

TD 4 : TD5

Sommaire

4.I Énoncés

4.I.1 Exercice 1

(voir la version "exercice résolu" page 83)

On considère la chaîne de Markov définie par la figure 4.3 suivante, en imposant le point de départ en A, les probabilités de transition étant toutes égales à $\frac{1}{2}$

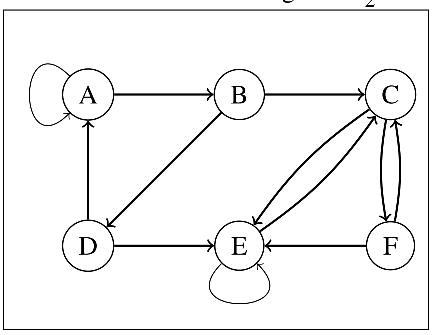


FIGURE4.3

4.I.1.a Question 1

Lancer un grand nombre de marches depuis le point A. Donner les distributions empiriques de ces marches après 1, 5, 10 et 15 pas. Quelle semble être la loi limite de la position du marcheur ?

4.I.1.b Question 2

Justifier le support de cette loi limite (c'est-à-dire les points où elle est non nulle).

4.I.1.c Question 3

Écrire la matrice de transition P de la chaîne de Markov. Vérifier que la loi limite est bien vecteur propre à gauche de P.

4.I.2 Exercice 2

(voir la version "exercice résolu" page 87)

Pour chacune des matrices suivantes, vérifier qu'il s'agit bien d'une matrice de transition d'une chaîne de Markov, puis représenter graphiquement la chaîne, et enfin déterminer l'ensemble des états récurrents et des états transients ainsi que l'ensemble des mesures invariantes.

- $P_1 = \begin{pmatrix} \frac{2}{5} & \frac{3}{5} & 0 \\ \frac{1}{5} & \frac{1}{2} & \frac{3}{10} \\ 0 & \frac{2}{5} & \frac{3}{5} \end{pmatrix}$
- $P_2 = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- $P_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

4.I.3 Exercice 3

(voir la version "exercice résolu" page 93)

On considère la propagation d'un information d'un individu 0 à un individu n via n − 1 relais. On note α la probabilité que l'information soit fausse à l'individu k + 1 sachant qu'elle était vraie en k et β la probabilité qu'elle soit vraie en k + 1 alors qu'elle était fausse en k.

4.I.3.a Question 1

Modéliser l'état de l'information à l'aide d'une chaîne de Markov.

4.I.3.b Question 2

En déduire un moyen de calculer la probabilité que l'information soit encore vraie pour un nombre de pas tendant vers l'infini.

4.I.4 Exercice 4

(voir la version "exercice résolu" page 95)

On considère un graphe non-orienté connexe ayant un circuit de longueur p et un circuit de longueur l, avec l etp premiers entre eux, et une marche aléatoire équiprobable sur ce graphe (quand elle est sur un sommet v de degré d(v), elle prend chaque arête possible avec la probabilité $\frac{1}{\mathrm{d(v)}}$).

4.I.4.a Question 1

Justifier l'unicité de la mesure invariante μ pour cette chaîne de Markov.

4.I.4.b Question 2

Montrer que pour tout état i, $\mu(\mathrm{i}) = \sum_{\mathrm{j} \in \mathrm{N(i)}} \frac{1}{\mathrm{d(j)}} \mu(\mathrm{j})$.

4.I.4.c Question 3

En déduire que si G est régulier (tous les sommets sont de même degré), μ est la mesure uniforme.

Remarque : μ(i) étant liée à la fréquence avec laquelle une marche aléatoire infinie passe par le sommet i, cette quantité peut être vue comme une popularité liée au sommet i. L'égalité de la question 2 se rtaduit alor par le fait qu'un état est d'autant plus populaire que ses voisins sont populaires. La détermination de μ est à la base de l'algorithme PageRank.

4.II Exercices résolus

Exercice 4.II.1 (*Exercice 1 (TD4)*)

On considère la chaîne de Markov définie par la figure 4.3 suivante, en imposant le point de départ en A, les probabilités de transition étant toutes égales à $\frac{1}{2}$

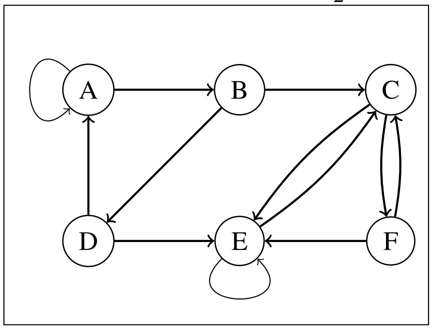


FIGURE4.3

Question #1 : Lancer un grand nombre de marches depuis le point A. Donner les distributions empiriques de ces marches après 1, 5, 10 et 15 pas. Quelle semble être la loi limite de la position du marcheur ?

(solution page 86)

Question #2 : Justifier le support de cette loi limite (c'est-à-dire les points où elle est non nulle).

(solution page 86)

Question #3 : Écrire la matrice de transition P de la chaîne de Markov. Vérifier que la loi limite est bien vecteur propre à gauche de P.

(solution page 86)

Solution de l’exercice 4.II.2

On considère la chaîne de Markov définie par la figure 4.3 suivante, en imposant le point de départ en A, les probabilités de transition étant toutes égales à $\frac{1}{2}$

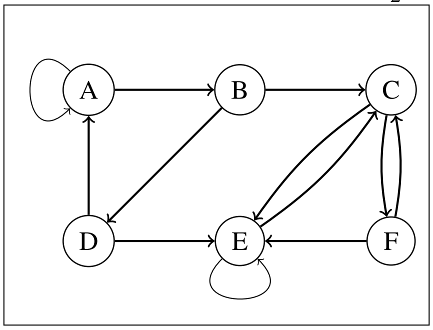


FIGURE4.3

1. **Lancer un grand nombre de marches depuis le point A. Donner les distributions empiriques de ces marches après 1, 5, 10 et 15 pas. Quelle semble être la loi limite de la position du marcheur ?**

	A	B	C	D	E	F
0	$\frac{22}{22}$	0	0	0	0	0
1	$\frac{12}{22}$	$\frac{10}{22}$	0	0	0	0
5	$\frac{2}{22}$	$\frac{1}{22}$	$\frac{6}{22}$	$\frac{2}{22}$	$\frac{7}{22}$	$\frac{4}{22}$
10	0	0	$\frac{6}{22}$	$\frac{1}{22}$	$\frac{8}{22}$	$\frac{7}{22}$
15	0	0	$\frac{5}{22}$	0	$\frac{14}{22}$	$\frac{3}{22}$
loi limite	0	0	$\frac{1}{3}$	0	$\frac{1}{2}$	$\frac{1}{6}$

2. **Justifier le support de cette loi limite (c'est-à-dire les points où elle est non nulle).**

   Les composantes fortement connexes du graphe sont ABD et CEF, avec une unique arête entre elles,allant de ABD vers CEF.

   La probabilité de rester dans ABD est nulle car il s'agit d'un état transient et que tout événement de probabilité non nulle finit par se produire. On finit donc à coup sûr par être piégé dans CEF.

3. **Écrire la matrice de transition P de la chaîne de Markov. Vérifier que la loi limite est bien vecteur propre à gauche de P.**

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

$$\left(0; 0; \tfrac{1}{3}; 0; \tfrac{1}{2}; \tfrac{1}{6}\right)P = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{4} + \frac{1}{12} \\ 0 \\ \frac{1}{6} + \frac{1}{4} + \frac{1}{12} \\ \frac{1}{6} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{3} \\ 0 \\ \frac{1}{2} \\ \frac{1}{6} \end{pmatrix}$$

Exercice 4.II.3 (***Exercice 2 (TD4)***)

Pour chacune des matrices suivantes, vérifier qu'il s'agit bien d'une matrice de transition d'une chaîne de Markov, puis représenter graphiquement la chaîne, et enfin déterminer l'ensemble des états récurrents et des états transients ainsi que l'ensemble des mesures invariantes.

- $P_1 = \begin{pmatrix} \frac{2}{5} & \frac{3}{5} & 0 \\ \frac{1}{5} & \frac{1}{2} & \frac{3}{10} \\ 0 & \frac{2}{5} & \frac{3}{5} \end{pmatrix}$
- $P_2 = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

– $P_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

Solution de l'exercice 4.II.4

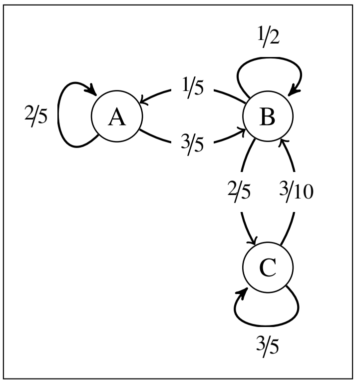
Pour chacune des matrices suivantes, vérifier qu'il s'agit bien d'une matrice de transition d'une chaîne de Markov, puis représenter graphiquement la chaîne, et enfin déterminer l'ensemble des états récurrents et des états transients ainsi que l'ensemble des mesures invariantes.

– $P_1 = \begin{pmatrix} \frac{2}{5} & \frac{3}{5} & 0 \\ \frac{1}{5} & \frac{1}{2} & \frac{3}{10} \\ 0 & \frac{2}{5} & \frac{3}{5} \end{pmatrix}$

– $P_2 = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

– $P_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

– 1. $P_1 = \begin{pmatrix} \frac{2}{5} & \frac{3}{5} & 0 \\ \frac{1}{5} & \frac{1}{2} & \frac{3}{10} \\ 0 & \frac{2}{5} & \frac{3}{5} \end{pmatrix} \Longrightarrow \begin{cases} \text{tous les coefficients sont entre 0 et 1} \\ \frac{2}{5} + \frac{3}{5} + 0 = 1 \\ \frac{1}{5} + \frac{1}{2} + \frac{3}{10} = 1 \\ 0 + \frac{2}{5} + \frac{3}{5} = 1 \end{cases}$ donc ses éléments sont bien des probabilités.

2. 

3. Il n'y a pas d'état transient car une seule composante fortement connexe (le graphe tout entier)

4. La chaîne est irréductible apériodique, donc il n'y a qu'une seule mesure invariante.
   Cherchons les vecteurs propres à gauche de P_1 associés à la valeur propre 1.

   Ils vérifient $(a; b; c)P_1 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$.

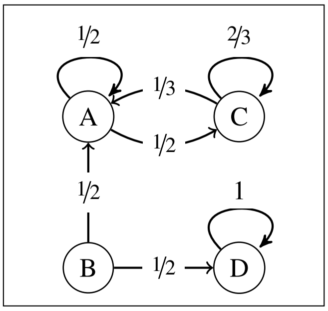
   $$\begin{cases} 5a = 2a + b \implies 3a = b \\ 5b = 3a + \dfrac{3b}{2} + 2c \\ 5c = \dfrac{3b}{2} + 3c \implies 4c = 3b \end{cases}$$

   Ce sont les vecteurs de la forme $\begin{pmatrix} b/3 \\ b \\ 3b/4 \end{pmatrix}$.

   La mesure invariante vérifie $\frac{b}{3} + b + \frac{3b}{4} = 1 \implies b = \frac{12}{25}$. C'est donc $\begin{pmatrix} \text{nicefrac425} \\ 12/25 \\ 9/25 \end{pmatrix}$.

   On trouve bien une unique mesure invariante.

– 1. $P_2 = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \implies \begin{cases} \text{tous les coefficients sont entre 0 et 1} \\ \frac{1}{2} + 0 + \frac{1}{2} + 0 = 1 \\ \frac{1}{2} + 0 + 0 + \frac{1}{2} = 1 \\ \frac{1}{3} + 0 + \frac{2}{3} + 0 = 1 \\ 0 + 0 + 0 + 1 = 1 \end{cases}$ donc ses éléments sont bien des probabilités.

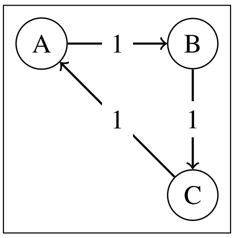
2. 

3. Il y a trois composantes fortement connexes : AC, B et D. B est transient tandis que A, C et D sont récurrents.
4. La chaîne est apériodique mais n'est pas irréductible car le graphe n'est pas fortement connexe, donc il n'y a une infinité de mesures invariantes.
   Cherchons les vecteurs propres à gauche de P_2 associés à la valeur propre 1 qui sont des mesures invariantes.
   Ils vérifient $(a; b; c; d)P_2 = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$ et $a + b + c + d = 1$.

$$\begin{cases} a = \frac{a}{2} + \frac{b}{2} + \frac{c}{3} \\ c = \frac{a}{2} + \frac{2a}{3} \\ d = \frac{b}{2} + d \\ a + b + c + d = 1 \end{cases} \implies \begin{cases} \frac{a}{2} = \frac{c}{3} \\ \frac{b}{2} = 0 \\ a + b + c + d = 1 \end{cases}$$

$$\implies \begin{cases} c = \frac{3a}{2} \\ b = 0 \\ d = 1 - (a + c) \end{cases}$$

   Ce sont les vecteurs de la forme $\begin{pmatrix} a \\ 0 \\ \frac{3a}{2} \\ 1 - \frac{5a}{2} \end{pmatrix} \forall a \in [0\,; 1]$.
   On trouve bien une infinité de mesures invariantes : celle vers laquelle tendent les trajectoires dépend de la distribution de départ.

– 1. $P_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \implies$ tous les coefficients valent 0 ou 1, à raison d'un seul 1 par ligne, donc tous les éléments de la matrice sont compris entre 0 et 1 et la somme des éléments de chaque ligne vaut bien 1 : ce sont bien des probabilités.
2. 

3. Il n'y a pas d'état transient car une seule composante fortement connexe (le graphe tout entier)
4. La chaîne est irréductible mais périodique, donc il n'y a qu'une seule mesure invariante vers laquelle ne tendront pas les trajectoires.
   Cherchons les vecteurs propres à gauche de P_3 associés à la valeur propre 1 qui sont des mesures invariantes.
   Ils vérifient $(a; b; c)P_3 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$ et $a + b + c = 1$.

$$\begin{cases} a + b + c = 1 \\ a = c \\ b = a \\ c = b \end{cases}$$ La mesure invariante est le vecteur $\begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$ On trouve bien une unique mesure invariante.

Exercice 4.II.5 (***Exercice 3 (TD4)***)

On considère la propagation d'un information d'un individu 0 à un individu n via n – 1 relais. On note α la probabilité que l'information soit fausse à l'individu k + 1 sachant qu'elle était vraie en k et β la probabilité qu'elle soit vraie en k + 1 alors qu'elle était fausse en k.

Question #1 : Modéliser l'état de l'information à l'aide d'une chaîne de Markov.

(solution page 94)

Question #2 : En déduire un moyen de calculer la probabilité que l'information soit encore vraie pour un nombre de pas tendant vers l'infini.

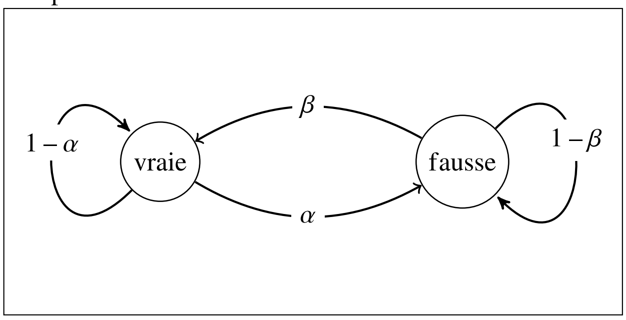
(solution page 95)

Solution de l'exercice 4.II.6

On considère la propagation d'un information d'un individu 0 à un individu n via $n-1$ relais. On note α la probabilité que l'information soit fausse à l'individu $k+1$ sachant qu'elle était vraie en k et β la probabilité qu'elle soit vraie en $k+1$ alors qu'elle était fausse en k.

1. **Modéliser l'état de l'information à l'aide d'une chaîne de Markov.**

   On peut modéliser la propagation de l'information à l'aide d'une marche aléatoire sur les états "vraie" et "fausse", c'est-à-dire par la chaîne de Markov suivante :



Le problème s'écrit alors "la distribution initiale est $(1;0)$, c'est-à-dire que l'information est initialement vraie avec probabilité 1. Quelle est la distribution après un grand nombre de pas ?"

2. **En déduire un moyen de calculer la probabilité que l'information soit encore vraie pour un nombre de pas tendant vers l'infini.**

La matrice de transition de cette chaîne de Markov est $P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$.

De plus, le graphe étant fortement connexe, la chaîne est irréductible donc admet une unique mesure invariante. En outre, si $(\alpha,\beta) \notin \{(0;0),(1;1),(0;1),(1,0)\} = \{0;1\}^2$, la chaîne est apériodique (car il y a au moins une auto-arête et une arête qui n'est pas une auto-arête) et la distribution après un grand nombre de pas tend donc vers l'unique mesure invariante qui vérifie par ailleurs $\mu P = \mu$

$$\mu = \begin{pmatrix} u \\ v \end{pmatrix} \implies \begin{cases} (u;v)P = \begin{pmatrix} u \\ v \end{pmatrix} \\ u+v=1 \end{cases} \implies \begin{cases} (1-\alpha)u+\beta v = u \\ \alpha v + (1-\beta)u = v \\ u+v=1 \end{cases} \implies \begin{cases} \beta v = \alpha u \\ u+v=1 \end{cases} \implies \begin{cases} u = \dfrac{\beta}{\alpha+\beta} \\ v = 1-u = 1-\dfrac{\beta}{\alpha+\beta} = \dfrac{\alpha}{\alpha+\beta} \end{cases}$$

Comme u est la coordonnées de μ qui correspond à "vraie" dans la mesure invariante ainsi modélisée, la probabilité d'avoir la bonne information à très longue échéance est de $\dfrac{\beta}{\alpha+\beta}$

On constate de plus que si $\beta = 1$ et $\alpha = 1$, la mesure invariante vaut $\frac{1}{2}$ et, comme la chaîne est alors périodique, la probabilité est de 1 après un nombre pair de pas et de 0 après un nombre impair de pas. Dans le cas où $\alpha = 0$ (quelque soit la valeur de β), comme l'information est initialement vraie, elle le sera toujours avec probabilité 1. Et dans le cas où $\beta = 0$ mais $\alpha \neq 0$, elle sera de probabilité 0 après un grand nombre de pas car tout événement de probabilité non nulle finit par se produire.

Exercice 4.II.7 (*Exercice 4 (TD4)*)

On considère un graphe non-orienté connexe ayant un circuit de longueur p et un circuit de longueur l, avec l etp premiers entre eux, et une marche aléatoire équiprobable sur ce graphe (quand elle est sur un sommet v de degré d(v), elle prend chaque arête possible avec la probabilité $\frac{1}{d(v)}$).

Question #1 : Justifier l'unicité de la mesure invariante μ pour cette chaîne de Markov.

(solution page 96)

Question #2 : Montrer que pour tout état i, $\mu(i) = \sum_{j \in N(i)} \frac{1}{d(j)} \mu(j)$.

(solution page 96)

Question #3 : En déduire que si G est régulier (tous les sommets sont de même degré), μ est la mesure uniforme.

(solution page 96)

Remarque : $\mu(i)$ étant liée à la fréquence avec laquelle une marche aléatoire infinie passe par le sommet i, cette quantité peut être vue comme une popularité liée au sommet i. L'égalité de la question 2 se rtaduit alor par le fait qu'un état est d'autant plus populaire que ses voisins sont populaires. La détermination de μ est à la base de l'algorithme PageRank.

Solution de l'exercice 4.II.8

On considère un graphe non-orienté connexe ayant un circuit de longueur p et un circuit de longueur l, avec l etp premiers entre eux, et une marche aléatoire équiprobable sur ce graphe (quand elle est sur un sommet v de degré d(v), elle prend chaque arête possible avec la probabilité $\frac{1}{d(v)}$).

Exemples :

1. **Justifier l'unicité de la mesure invariante μ pour cette chaîne de Markov.**
   L'arête (i; j) dans le graphe donne une transition $i \to j$ de probabilité $\frac{1}{d(i)}$ et une transition $j \to i$ de probabilité $\frac{1}{d(j)}$.
   Le graphe d'origine étant non-orienté et connexe, chacune de ses arêtes est transformée en un couple d'arêtes, une dans chaque sens, de même extrémités dans le graphe de la chaîne de Markov aléatoire équiprobable, donc ce dernier graphe est fortement connexe. En effet, tout chemin non-orienté du premier graphe peut être transformé en un chemin orienté dans le dernier graphe. La mesure invariante est par conséquent unique.
2. **Montrer que pour tout état** i**,** $\mu(i) = \sum_{j \in N(i)} \frac{1}{d(j)} \mu(j)$**.**
   Soit $P = \left(p_{ij}\right)_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant n}}$ avec $p_{ij} = \begin{cases} \frac{1}{d(i)} & \text{si } i \in N(j) \\ 0 & \text{sinon} \end{cases}$. Alors $\mathrm{sum}_i \mu(i) = 1$ et $\mu = \mu P$.
   $\Longrightarrow \ \forall\, i \in [\![1\,;n]\!],\ \mu(i) = \sum_{j=1}^{n} \mu(j) p_{ji} = \sum_{j \in N(i)} \frac{1}{d(j)} \mu(j) + \sum_{j \notin N(i)} 0.$
3. **En déduire que si** G **est régulier (tous les sommets sont de même degré),** μ **est la mesure uniforme.**
   La mesure uniforme fonctionne pour les graphes réguliers, donc, par unicité de la mesure invariante, la mesure uniforme est la mesure invariante pour les graphes réguliers.

4.III Ce qui a été montré

TD 5 : Contrôle Continu 2013

Sommaire

5.I Énoncés

5.I.1 Exercice 1

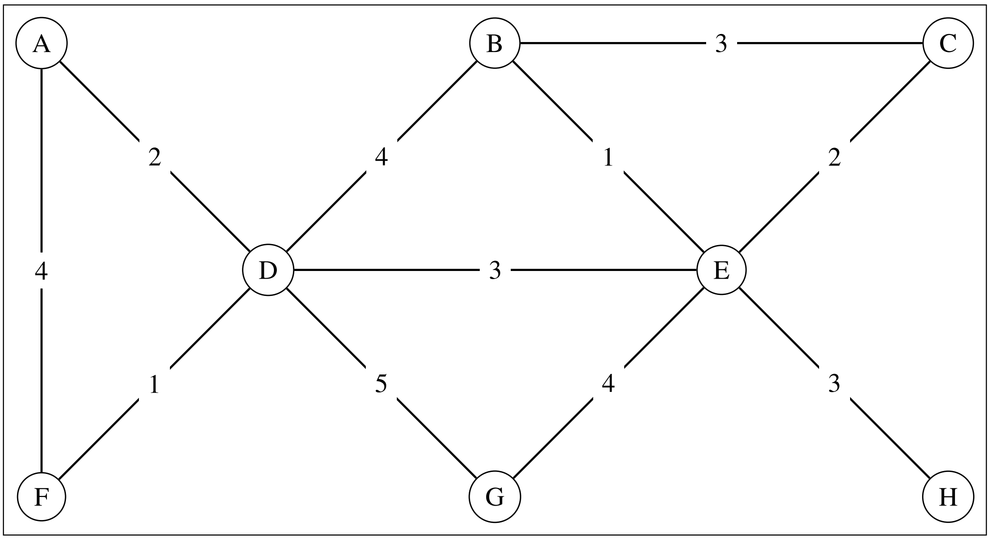


FIGURE5.4

On considère le graphe 5.4 ci-dessus.

5.I.1.a Question 1

Écrire la matrice d'adjacence du graphe.

5.I.1.b Question 2

Appliquer l'algorithme de Kruskal pour trouver un arbre couvrant de poids minimal du graphe. La rédaction indiquera l''ordre dans lequel les arêtes sont considérées et la décision qui est prise pour chacune d'elle.

5.I.2 Exercice 2

Soit G un graphe non-orienté connexe.Un sommet v est dit *séparateur* si G – v (le graphe obtenu en supprimant v) n'est plus connexe.

5.I.2.a Question 1

Le graphe 5.4 comporte-t-il un ou des sommet(s) séparateur(s), et si oui, le(s)quel(s) ?

5.I.2.b Question 2

On considère le graphe 5.4 en ne tenant pas compte des poids des arêtes. Dessiner un parcours en profondeur sur ce graphe, de racine D, en le représentant comme un arbre enraciné (la racine en haut et on descend d'un niveau à chaque parcours d'un nouveau sommet).
Que peut-on dire du degré de D dans cet arbre et du nombre de composantes connexes dans le graphe privé de D ?

5.I.2.c Question 3

On considère un graphe non-orienté quelconque G et un parcours en profondeur T de ce graphe, de racine r. Justifier que les branches issues de chacun des fils de r dans T forment des composantes connexes distinctes dans G – r. En déduire une caractérisation du fait que r est un séparateur en fonction de son degré dans T.

5.I.2.d Question 4

En déduire un algorithme permettant de dresser la liste des séparateurs d'un graphe non-orienté connexe. Quelle est sa complexité ?

Remarque : cet algorithme n'est pas optimal, il est possible d'énumérer les séparateurs en temps linéaire.

5.I.3 Exercice 3

Une entreprise, basée dans la ville A, cherche à déterminer les plans de vol les moins chers pour ses agents vers les villes B à F. Le prix moyen d'un billet entre deux villes est donné dans la matrice suivante,le signe ∞ désignant l'absence de liaison entre les villes.

$$\begin{array}{l} \text{A} \\ \text{A0} \\ \text{B50} \\ \text{C50} \\ \text{D}\infty \\ \text{E100} \\ \text{F250} \end{array}
\begin{array}{c} \begin{array}{ccccc} \text{B} & \text{C} & \text{D} & \text{E} & \text{F} \end{array} \\ \begin{pmatrix} 50 & 50 & \infty & 100 & 250 \\ 0 & 80 & 200 & \infty & 80 \\ 80 & 0 & 100 & 50 & 180 \\ 200 & 100 & 0 & \infty & 150 \\ \infty & 50 & \infty & 0 & \infty \\ 80 & 180 & 150 & \infty & 0 \end{pmatrix} \end{array}$$

5.I.3.a Question 1

Construire l'ensemble des trajets les moins chers de A vers chacune des autres villes. Si vous appliquez un algorithme connu sur un graphe donné, précisez quel algorithme et dessinez le graphe que vous utilisez. Sinon, expliquez l'algorithme que vous utilisez.

5.I.3.b Question 2

La compagnie souhaite trouver les trajets les moins chers mais en imposant au plus une escale à ses agents. Quelles modifications cela apporte-t-il au cas particulier précédent ?
Comment construiriez-vous un algorithme pour résoudre ce problème en général ?
Comment le généraliser au cas où p escales sont possibles au maximum ?

5.II Exercies résolus

Exercice 5.II.1 (*Exercice 1 (Contrôle Continu 2013)*)

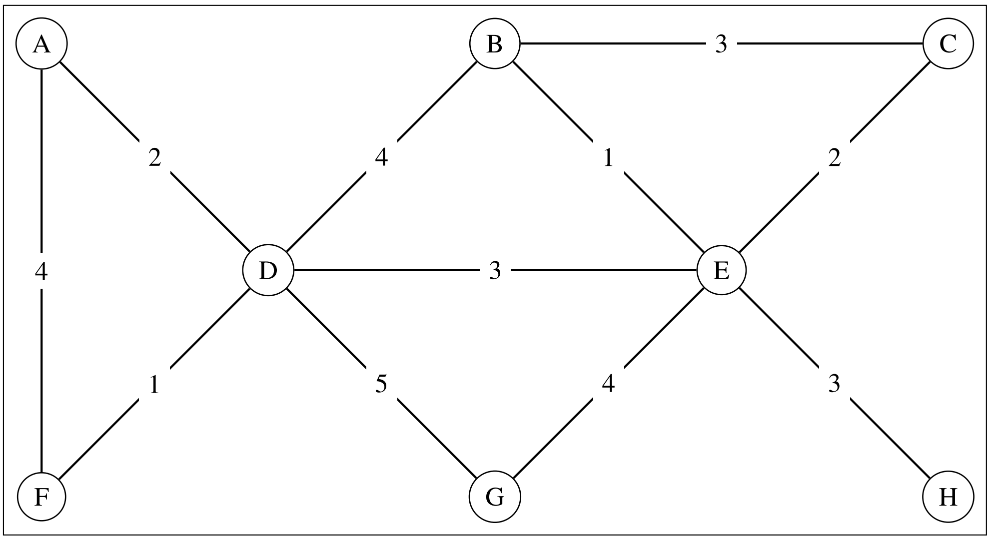


FIGURE5.4

On considère le graphe 5.4 ci-dessus.

Question #1 : Écrire la matrice d'adjacence du graphe.

(solution page 101)

Question #2 : Appliquer l'algorithme de Kruskal pour trouver un arbre couvrant de poids minimal du graphe. La rédaction indiquera l''ordre dans lequel les arêtes sont considérées et la décision qui est prise pour chacune d'elle.

(solution page 101)

Solution de l'exercice 5.II.2

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

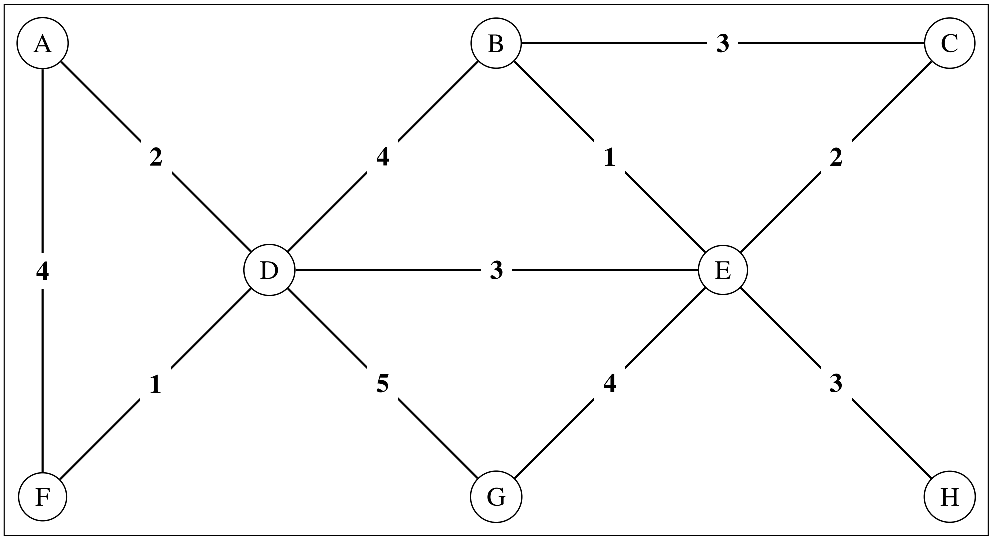


FIGURE5.4

On considère le graphe 5.4 ci-dessus.

1. **Écrire la matrice d'adjacence du graphe.**

2. **Appliquer l'algorithme de Kruskal pour trouver un arbre couvrant de poids minimal du graphe. La rédaction indiquera l''ordre dans lequel les arêtes sont considérées et la décision qui est prise pour chacune d'elle.**

Exercice 5.II.3 (*Exercice 2 (Contrôle Continu 2013)*)

Soit G un graphe non-orienté connexe.Un sommet v est dit *séparateur* si G – v (le graphe obtenu en supprimant v) n'est plus connexe.

Question #1 : Le graphe 5.4 comporte-t-il un ou des sommet(s) séparateur(s), et si oui, le(s)quel(s) ?

(solution page 102)

Question #2 : On considère le graphe 5.4 en ne tenant pas compte des poids des arêtes. Dessiner un parcours en profondeur sur ce graphe, de racine D, en le représentant comme un arbre enraciné (la racine en haut et on descend d'un niveau à chaque parcours d'un nouveau sommet).
Que peut-on dire du degré de D dans cet arbre et du nombre de composantes connexes dans le graphe privé de D ?

(solution page 102)

Question #3 : On considère un graphe non-orienté quelconque G et un parcours en profondeur T de ce graphe, de racine r. Justifier que les branches issues de chacun des fils de r dans T forment des composantes connexes distinctes dans G – r. En déduire une caractérisation du fait que r est un séparateur en fonction de son degré dans T.

(solution page 102)

Question #4 : En déduire un algorithme permettant de dresser la liste des séparateurs d'un graphe non-orienté connexe. Quelle est sa complexité ?

(solution page 102)

Remarque : cet algorithme n'est pas optimal, il est possible d'énumérer les séparateurs en temps linéaire.

Solution de l'exercice 5.II.4

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

Soit G **un graphe non-orienté connexe.Un sommet** v **est dit *séparateur* si** G – v **(le graphe obtenu en supprimant** v**) n'est plus connexe.**

1. **Le graphe 5.4 comporte-t-il un ou des sommet(s) séparateur(s), et si oui, le(s)quel(s) ?**

2. **On considère le graphe 5.4 en ne tenant pas compte des poids des arêtes. Dessiner un parcours en profondeur sur ce graphe, de racine** D**, en le représentant comme un arbre enraciné (la racine en haut et on descend d'un niveau à chaque parcours d'un nouveau sommet).**
   Que peut-on dire du degré de D **dans cet arbre et du nombre de composantes connexes dans le graphe privé de** D **?**

3. **On considère un graphe non-orienté quelconque** G **et un parcours en profondeur** T **de ce graphe, de racine** r**. Justifier que les branches issues de chacun des fils de** r **dans** T **forment des composantes connexes distinctes dans** G – r**. En déduire une caractérisation du fait que** r **est un séparateur en fonction de son degré dans** T**.**

4. **On considère le graphe 5.4 en ne tenant pas compte des poids des arêtes. Dessiner un parcours en profondeur sur ce graphe, de racine** D**, en le représentant comme un arbre enraciné (la racine en haut et on descend d'un niveau à chaque parcours d'un nouveau sommet).**
   Que peut-on dire du degré de D **dans cet arbre et du nombre de composantes connexes dans le graphe privé de** D **?**

Exercice 5.II.5 (*Exercice 3 (Contrôle Continu 2013)*)

Une entreprise, basée dans la ville A, cherche à déterminer les plans de vol les moins chers pour ses agents vers les villes B à F. Le prix moyen d'un billet entre deux villes est donné dans la matrice suivante,le signe ∞ désignant l'absence de liaison entre les villes.

$$
\begin{array}{ll}
\text{A} & \begin{array}{ccccc} \text{B} & \text{C} & \text{D} & \text{E} & \text{F} \end{array} \\
\begin{array}{l} \text{A0} \\ \text{B50} \\ \text{C50} \\ \text{D}\infty \\ \text{E100} \\ \text{F250} \end{array} &
\begin{pmatrix}
50 & 50 & \infty & 100 & 250 \\
0 & 80 & 200 & \infty & 80 \\
80 & 0 & 100 & 50 & 180 \\
200 & 100 & 0 & \infty & 150 \\
\infty & 50 & \infty & 0 & \infty \\
80 & 180 & 150 & \infty & 0
\end{pmatrix}
\end{array}
$$

Question #1 : Construire l'ensemble des trajets les moins chers de A vers chacune des autres villes. Si vous appliquez un algorithme connu sur un graphe donné, précisez quel algorithme et dessinez le graphe que vous utilisez. Sinon, expliquez l'algorithme que vous utilisez.

(solution page 103)

Question #2 : La compagnie souhaite trouver les trajets les moins chers mais en imposant au plus une escale à ses agents. Quelles modifications cela apporte-t-il au cas particulier précédent ?

Comment construiriez-vous un algorithme pour résoudre ce problème en général ?

Comment le généraliser au cas où p escales sont possibles au maximum ?

(solution page 103)

Solution de l'exercice 5.II.6

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

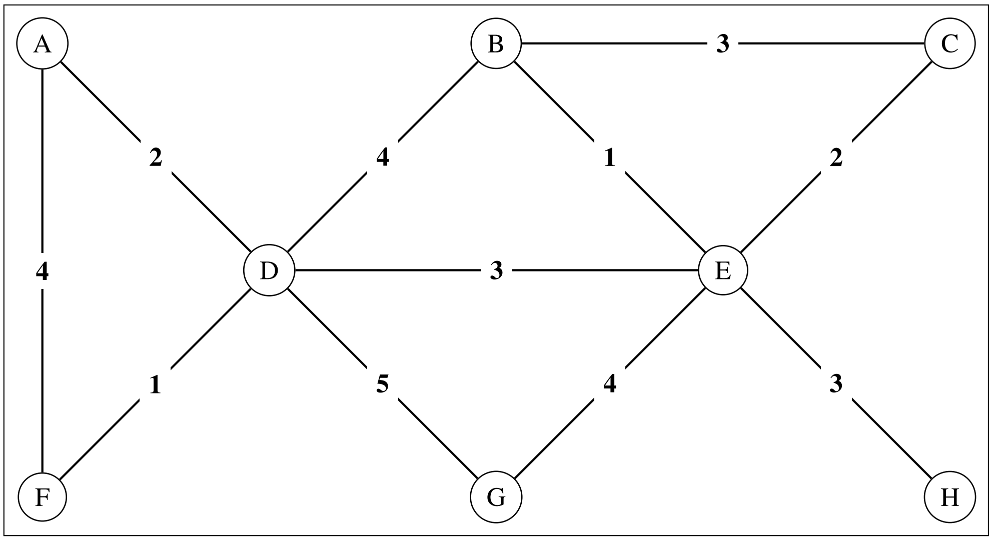


FIGURE5.4

On considère le graphe 5.4 ci-dessus.

1. **Construire l'ensemble des trajets les moins chers de** A **vers chacune des autres villes. Si vous appliquez un algorithme connu sur un graphe donné, précisez quel algorithme et dessinez le graphe que vous utilisez. Sinon, expliquez l'algorithme que vous utilisez.**

2. **La compagnie souhaite trouver les trajets les moins chers mais en imposant au plus une escale à ses agents. Quelles modifications cela apporte-t-il au cas particulier précédent ?**
   Comment construiriez-vous un algorithme pour résoudre ce problème en général ?
   Comment le généraliser au cas où p **escales sont possibles au maximum ?**

TD 6 : Annales 2012-2013 : Contrôle Continu

Sommaire

6.I Énoncés

6.I.1 Exercice 1

(voir la version "exercice résolu" page 107)

Soit G le graphe défini par la matrice d'adjacence suivante :

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccc} A & B & C & D & E & F & G & H & I & J & K & L \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \\ K \\ L \end{array} &
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

Pour chaque algorithme, vous utiliserez l'ordre alphabétique lorsque vous aurez un choix à faire.

6.I.1.a Question 1

Donnez à partir de la matrice d'adjacence les degrés entrants, sortants et totaux de chaque sommet. Expliquez comment trouver ces informations à partir de la matrice d'adjacence seulement.

6.I.1.b Question 2

Dessinez le graphe G.

6.I.1.c Question 3

G est-il non-connexe, faiblement connexe ou fortement connexe ? Expliquez pourquoi.

6.I.1.d Question 4

Donnez la bordure de $\{A; B; C; G\}$.

6.I.1.e Question 5

Donnez un parcours en profondeur de G en utilisant l'algorithme DFS. Dessinez le graphe en coloriant/surlignant les arcs formant l'arborescence et en indiquant les dates de début et de fin de chaque sommet.

6.I.1.f Question 6

Donnez la nature de chaque arc (couvrant, liaison, avant ou retour).

6.I.1.g Question 7

Calculez la décomposition en composantes connexes du graphe G en utilisant le double parcours en profondeur. Vous donnerez le graphe transposé et le résultat du parcours en profondeur.

6.I.1.h Question 8

Dessinez le graphe réduit.

6.I.2 Exercice 2

(voir la version "exercice résolu" page 111)

Soit la matrice M suivante :

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Donnez la matrice M^4. Expliquez comment vous obtenez cette matrice.

6.I.3 Exercice 3

(voir la version "exercice résolu" page 112)

Pour chaque algorithme, vous utiliserez l'ordre alphabétique lorsque vous aurez un choix à faire.

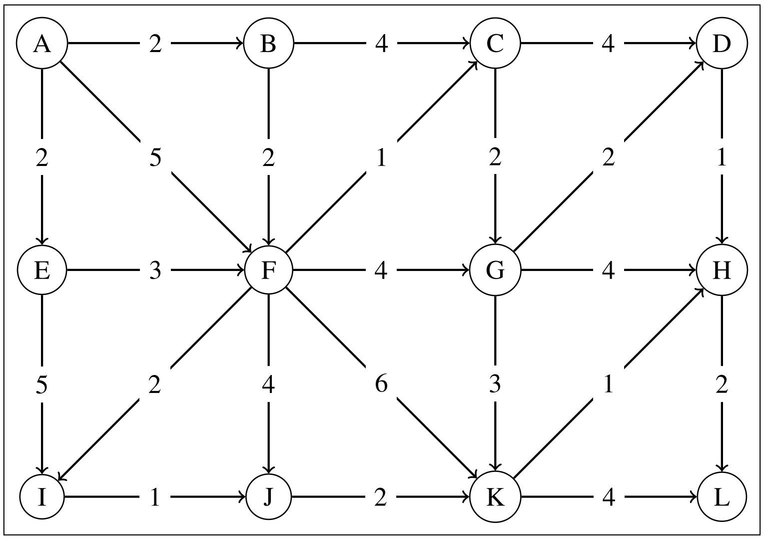


Figure 6.3

6.I.3.a Question 1

Donnez le chemin le plus court de A vers L. Dîtes quel algorithme vous utilisez et pourquoi. Indiquez les distances ez pères de tous les sommets.

6.I.3.b Question 2

Peut-on voir ce graphe comme un graphe de précédence ? Pourquoi ? Si c'est le cas, donnez un ordre des sommets.

6.I.3.c Question 3

Donnez un arbre couvrant minimum du graphe non-orienté correspondant.

6.I.4 Exercice 4

Exercice Bonus (voir la version "exercice résolu" page 115)

Montrez que dans un groupe de personnes, il y a toujours ayant le même nombre d'amis présents.

6.II Exercices résolus

Les solutions présentées ci-après ne sont en rien officielles ni validées par un professeur, il ne s'agit que de propositions.

Exercice 6.II.1 (*Exercice 1 (Annales 2012-2013 : Contrôle Continu) - 9 points*)

Soit G le graphe défini par la matrice d'adjacence suivante :

$$\begin{array}{cc}
 & \begin{array}{cccccccccccc} A & B & C & D & E & F & G & H & I & J & K & L \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \\ K \\ L \end{array} &
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}$$

Pour chaque algorithme, vous utiliserez l'ordre alphabétique lorsque vous aurez un choix à faire.

Question #1 : Donnez à partir de la matrice d'adjacence les degrés entrants, sortants et totaux de chaque sommet. Expliquez comment trouver ces informations à partir de la matrice d'adjacence seulement.

(solution page 108)

Question #2 : Dessinez le graphe G.

(solution page 109)

Question #3 : G est-il non-connexe, faiblement connexe ou fortement connexe ? Expliquez pourquoi.

(solution page 110)

Question #4 : Donnez la bordure de $\{A; B; C; G\}$.

(solution page 110)

Question #5 : Donnez un parcours en profondeur de G en utilisant l'algorithme DFS. Dessinez le graphe en coloriant/surlignant les arcs formant l'arborescence et en indiquant les dates de début et de fin de chaque sommet.

(solution page 110)

Question #6 : Donnez la nature de chaque arc (couvrant, liaison, avant ou retour).

(solution page 110)

Question #7 : Calculez la décomposition en composantes connexes du graphe G en utilisant le double parcours en profondeur. Vous donnerez le graphe transposé et le résultat du parcours en profondeur.

(solution page 110)

Question #8 : Dessinez le graphe réduit.

(solution page 110)

Solution de l'exercice 6.II.2

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

Soit G **le graphe défini par la matrice d'adjacence suivante :**

$$\begin{array}{cc}
 & \begin{array}{cccccccccccc} A & B & C & D & E & F & G & H & I & J & K & L \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \\ K \\ L \end{array} &
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}$$

Pour chaque algorithme, vous utiliserez l'ordre alphabétique lorsque vous aurez un choix à faire.

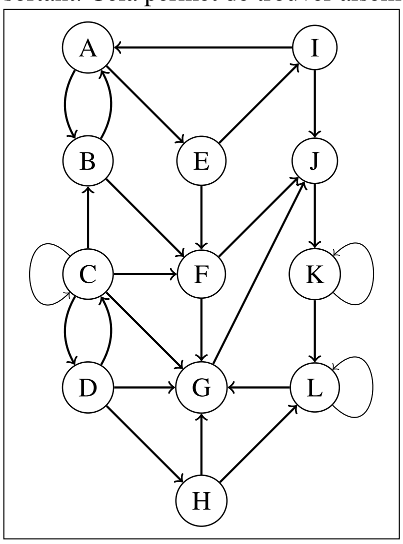
1. **Donnez à partir de la matrice d'adjacence les degrés entrants, sortants et totaux de chaque sommet. Expliquez comment trouver ces informations à partir de la matrice d'adjacence seulement.**
   La matrice d'adjacence est définie de la façon suivante :
   $$M = \left(m_{ij}\right)_{(i;j)\in V(G)^2} \text{ avec } \begin{cases} m_{ij} = 1 & \text{si } (i;j) \in E(G) \\ m_{ij} = 0 & \text{si } (i;j) \notin E(G) \end{cases}$$
   Pour connaître le degré sortant d'un sommet $i \in V(G)$, c'est-à-dire le nombre de sommets $j \in V(G)$ vérifiant $(i; j) \in E(G)$, il suffit d'additionner les valeurs de la ligne de M correspondant au sommet i.
   Réciproquement, pour son degré sortant, il suffit d'additionner les valeur de la colonne de M correspondant au sommet i.

Par conséquent, on a :

sommet	degré sortant	degré entrant
A	2	2
B	2	2
C	5	2
D	3	1
E	2	1
F	2	3
G	1	5
H	2	1
I	2	1
J	1	3
K	2	2
L	1	3

2. **Dessinez le graphe** G.

   Astuce pour la construction : partir du sommet avec le plus fort degré, ou, en cas d'égalité, celui ayant le plus fort degré sortant. Cela permet de trouver aisément le graphe avec ses sommets disposés de la façon suivante :



3. G **est-il non-connexe, faiblement connexe ou fortement connexe ? Expliquez pourquoi.**
Il n'y a pas de sommet isolé (n'étant ni source ni cible d'une arête), donc le graphe est connexe. De plus, s'il était faiblement connexe, il y aurait au moins une paire de sommets $\{u; v\}$ telle qu'il existe un chemin allant de u vers v mais pas de chemin allant de v vers u. Dans le cas contraire, le graphe serait fortement connexe.
On remarque qu'il existe un chemin allant de H à G (l'arête $(H; G)$) mais aucun permettant d'aller de G à H, car le seul voisin externe de G est J,le seul voisin externe de J est K, de K, il n'est possible d'aller que vers K ou L et L n'a d'autre voisins extérieurs que G et lui-même, ce qui fait que depuis G, on ne peut que parcourir un circuit contenant ces sommets, sans alternative.

4. **Donnez la bordure de** $\{A; B; C; G\}$**.**
//TODO : À vérifier
Par bordure de $\{A; B; C; G\}$, on entend l'union des voisinages externes des sommets de cet ensemble privée de ces mêmes sommets. Il s'agit donc de $N^{-}(A) \cup N^{-}(B) \cup N^{-}(C) \cup N^{-}(G) \setminus \{A; B; C; G\} = \{E; B\} \cup \{A; F\} \cup \{C; B; E; F; D; G\} \cup \{J\} \setminus \{A; B; C; G\} = \{E; F; D; J\}$

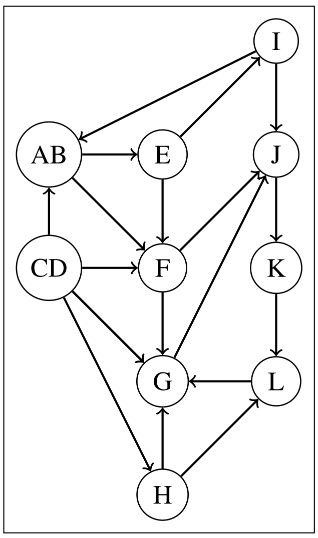
5. **Donnez un parcours en profondeur de** G **en utilisant l'algorithme DFS. Dessinez le graphe en coloriant/surlignant les arcs formant l'arborescence et en indiquant les dates de début et de fin de chaque sommet.**
//TODO

6. **Donnez la nature de chaque arc (couvrant, liaison, avant ou retour).**
//TODO

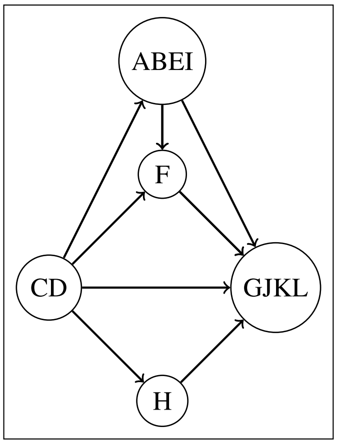
7. **Calculez la décomposition en composantes connexes du graphe** G **en utilisant le double parcours en profondeur. Vous donnerez le graphe transposé et le résultat du parcours en profondeur.**
//TODO

8. **Dessinez le graphe réduit.**
Il s'agit de regrouper/compresser chaque composante fortement connexe en un seul sommet.

Étape 1 : AB et CD



Étape 2 : ABEI et GJKL



Exercice 6.II.3 (*Exercice 2 (Annales 2012-2013 : Contrôle Continu) - 2 points*)

Soit la matrice M suivante :

$$M = \begin{array}{cc} & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{array}{c} A\\ B\\ C\\ D\\ E\\ F \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Donnez la matrice M^4. Expliquez comment vous obtenez cette matrice. (solution page 111)

Solution de l'exercice 6.II.4

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

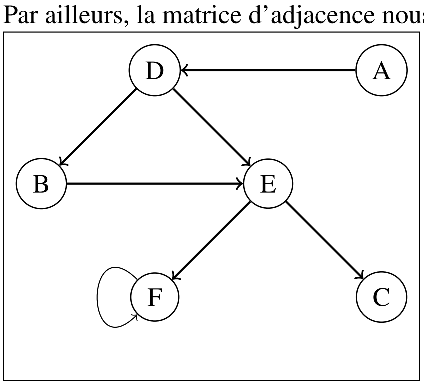
Soit la matrice M **suivante :**

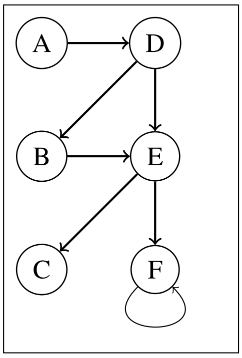
$$M = \begin{array}{cc} & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{array}{c} A\\ B\\ C\\ D\\ E\\ F \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Donnez la matrice M^4**. Expliquez comment vous obtenez cette matrice.**

m^n_{ij} contient le nombre de marches de longueur n partant du sommet i et allant vers le sommet j. c.f. Exercice 2 du TD1.

Par ailleurs, la matrice d'adjacence nous permet d'obtenir le graphe ci-dessous.



ce graphe peut aussi être écrit 

Recherchons sur ce graphe les marches possibles de longueur 4. On rappelle qu'à la différence des chemins, les marches peuvent repasser plus d'une fois par le même sommet.

marches partant de A ***de longueur*** 4 :

- A → D → E → C
- A → D → E → F
- A → D → B → E

Cela fait une marche de longueur 4 entre A et C, une entre A et F et une entre A et E

marches partant de B ***de longueur*** 4 : Il n'y a que B → E → F → F

Cela fait une marche de longueur 4 entre B et F

marches partant de C ***de longueur*** 4 : Il n'y en a pas car C n'a aucun voisin externe.

marches partant de D ***de longueur*** 4 :

- D → B → E → C
- D → B → E → F
- D → E → F → F

Cela fait une marche de longueur 4 entre D et C, et deux entre D et F

marches partant de E ***de longueur*** 4 : Il n'y a que E → F → F → F car F est le seul voisin externe de E

Cela fait une marche de longueur 4 entre E et F

marches partant de F ***de longueur*** 4 : Il n'y a que F → F → F → F car F est son seul voisin externe.

Cela fait seulement une marche de longueur 4 entre F et F

De là, on obtient $M^4 = \begin{array}{c c} & \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \end{array} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$

Exercice 6.II.5 (*Exercice 3 (Annales 2012-2013 : Contrôle Continu) - 9 points*)

Pour chaque algorithme, vous utiliserez l'ordre alphabétique lorsque vous aurez un choix à faire.

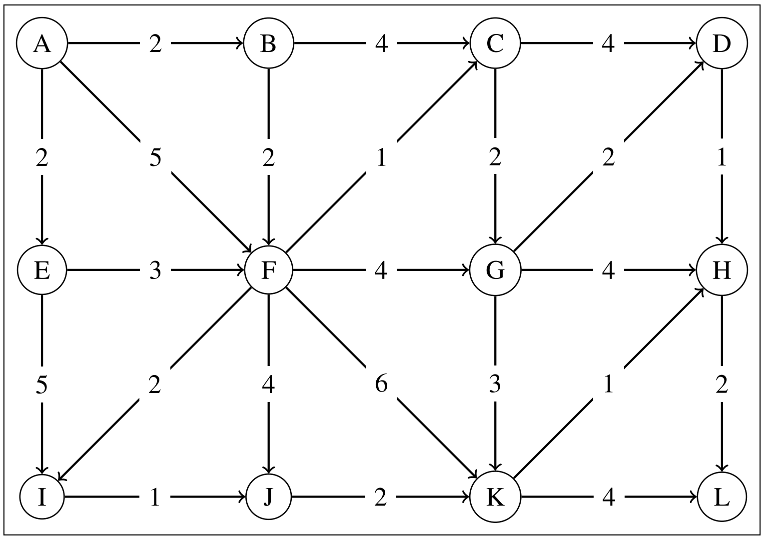


Figure 6.3

Question #1 : Donnez le chemin le plus court de A vers L. Dîtes quel algorithme vous utilisez et pourquoi. Indiquez les distances ez pères de tous les sommets.

(solution page 115)

Question #2 : Peut-on voir ce graphe comme un graphe de précédence ? Pourquoi ? Si c'est le cas, donnez un ordre des sommets.

(solution page 115)

Question #3 : Donnez un arbre couvrant minimum du graphe non-orienté correspondant.

(solution page 115)

Solution de l'exercice 6.II.6

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

Pour chaque algorithme, vous utiliserez l'ordre alphabétique lorsque vous aurez un choix à faire.

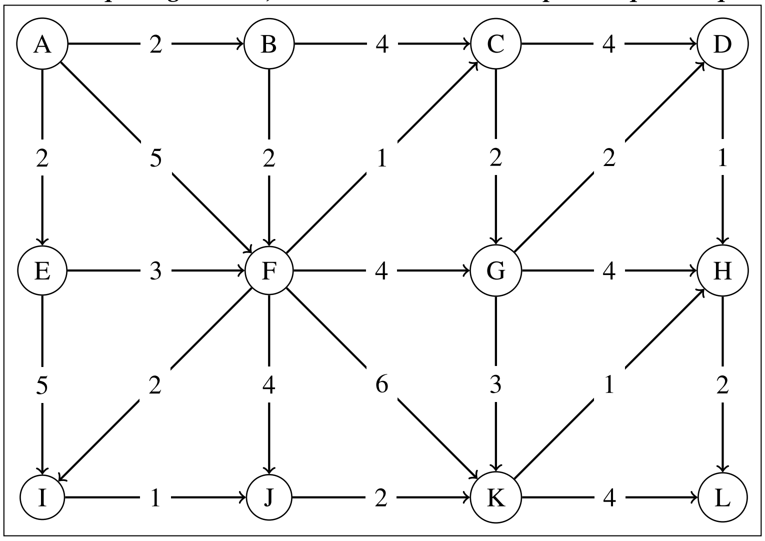


FIGURE 6.3

1. **Donnez le chemin le plus court de** A **vers** L**. Dîtes quel algorithme vous utilisez et pourquoi. Indiquez les distances ez pères de tous les sommets.**
   //TODO
2. **Peut-on voir ce graphe comme un graphe de précédence ? Pourquoi ? Si c'est le cas, donnez un ordre des sommets.**
   //TODO
3. **Donnez un arbre couvrant minimum du graphe non-orienté correspondant.**
   //TODO

Exercice 6.II.7 (*Exercice 4 (Annales 2012-2013 : Contrôle Continu) - 2 points bonus*)

Montrez que dans un groupe de personnes, il y a toujours ayant le même nombre d'amis présents. (solution page 115)

Solution de l'exercice 6.II.8

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

Montrez que dans un groupe de personnes, il y a toujours ayant le même nombre d'amis présents.

c.f. Exercice 4 du TD1

TD 7 : Annales 2012-2013 : Examen

Sommaire

7.I Énoncés

7.I.1 Exercice 1

(voir la version "exercice résolu" page 121)

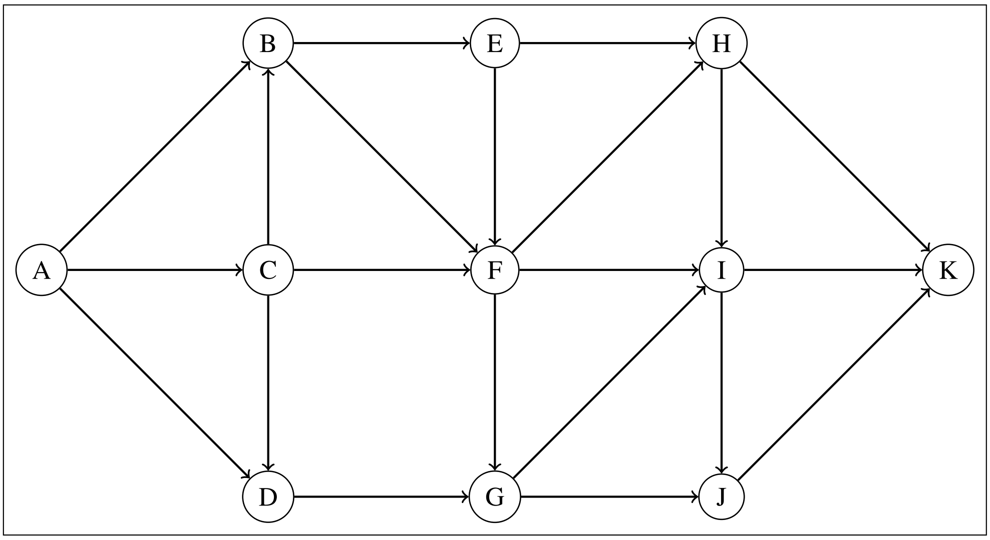


FIGURE7.4

Soit $\mathcal{G}$ le réseau de transport définit par le graphe 7.4 ci-dessus. Les sommets A et K sont respectivement la source et le puit du réseau. La capacité de tous les arc est de 5.
On définit le flot f par les valeurs suivantes :
$f(A,B) = 5$, $f(A,C) = 4$, $f(A,D) = 1$, $f(B,E) = 5$, $f(C,D) = 2$, $f(C,F) = 2$, $f(D,G) = 3$, $f(E,H) = 2$, $f(E,F) = 3$, $f(F,G) = 5$, $f(G,I) = 5$, $f(G,J) = 3$, $f(H,I) = 2$, $f(I,K) = 5$, $f(J,K) = 5$

7.I.1.a Question 1

Dessiner le flot f sur le graphe et donner la valeur du flot.

7.I.1.b Question 2

Soit $X = \{F, D, G\}$ et $Y = \{B, E, C\}$. Donner la capacité de X vers Y. Donner le flot de X vers Y.

7.I.1.c Question 3

Donner le réseau résiduel de $\mathcal{G}$ induit par le flot f.

7.I.1.d Question 4

A partir du flot f utiliser la méthode de Ford-Fulkerson pour obtenir le flot maximal du réseau $\mathcal{G}$.

7.I.2 Exercice 2

(voir la version "exercice résolu" page 124)

Soit le graphe 7.6 suivant :

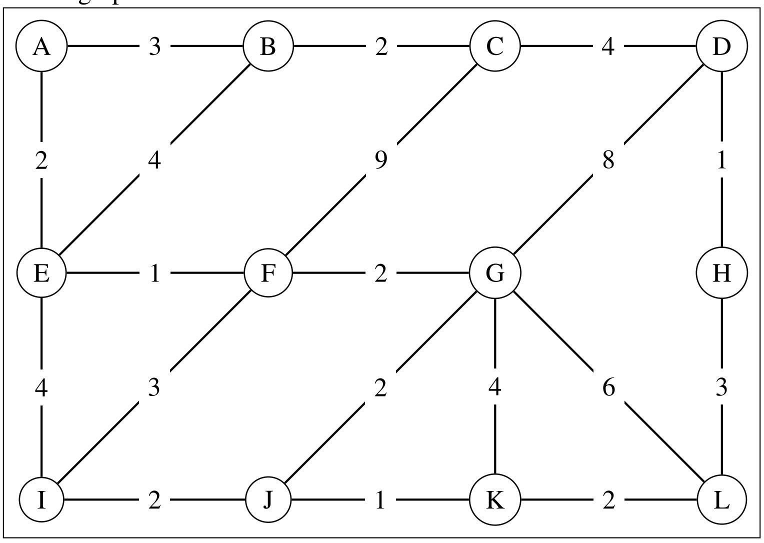


FIGURE7.6

7.I.2.a Question 1

Est-ce que ce graphe est eulérien ? Justifier.

7.I.2.b Question 2

Donner l'arbre couvrant minimum du graphe. Vous utilisez l'algorithme de Prim.

7.I.3 Exercice 3

(voir la version "exercice résolu" page 125)

Tâche	Durée	Tâche Précédente
F	B	3
C	A H B	4
G	R	3
E	G S R	3
R	-	3
S	D F	2
D	R	2
H	B	4
A	D F R	5
B	-	4

TABLE7.3

Soit le problème d'ordonnancement représenté par le tableau 7.3 ci-dessus.

7.I.3.a Question 1

Calculez le rang de chaque tache. Dessinez le graphe représentant le projet.

7.I.3.b Question 2

Donner les dates au plus tard, au plus tôt et les marges de chaque tache. Quelle sera la durée minimum du projet ?

7.I.4 Exercice 4

(voir la version "exercice résolu" page 126)

On modélise l'état d'un patient atteint d'un certain type cancer du poumon. Le cancer évolue à travers 3 stades notés S_1, S_2 et S_3 et commence en S_1. Chaque année, l'état du cancer peut évoluer.
En S_1, la probabilité de rester en S_1 est de $0,3$, la probabilité de guérison est de $0,2$ et la probabilité de passage en S_2 est de $0,5$.
En S_2, la probabilité de rester en S_2 est de $0,2$, la probabilité de guérison est de $0,1$ et la probabilité de passage en S_3 est de $0,7$.
En S_3, la probabilité de rester en S_3 est de $0,1$, la probabilité de guérison est de $0,1$ et la probabilité de décès $0,8$.

7.I.4.a Question 1

Modéliser l'état du cancer à l'aide d'une chaîne de Markov. Donner sa matrice de transition P et les états absorbants.

7.I.4.b Question 2

Donner la matrice de transition puissance 2 : P^2. Expliquez sa signification. Soit N la matrice fondamentale de P :

$$N = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{pmatrix} 0,49 & 0,89 & 0,69 \\ 0 & 1,25 & 0,97 \\ 0 & 0 & 1,11 \end{pmatrix} \end{array}$$

7.I.4.c Question 3

Donner le temps moyen qu'un patient reste malade (jusqu'à la guérison ou le décès). Expliquez votre réponse.

7.I.4.d Question 4

Quelles sont les probabilités qu'un patient venant de développer ce type de cancer soit guéri ou décède ? Expliquez votre réponse.

7.I.5 Exercice 5

(voir la version "exercice résolu" page 127)

Un groupe de personnes est tel que :
- chaque personne est membre d'exactement deux associations,
- chaque association comprend exactement trois membres,
- deux associations quelconques ont toujours exactement un membre en commun.

  Combien y a-t-il de personnes ? d'associations ? Justifier votre réponse à l'aide d'un graphe.

7.II Exercices résolus

Les solutions présentées ci-après ne sont en rien officielles ni validées par un professeur, il ne s'agit que de propositions.

Exercice 7.II.1 (*Exercice 1 (Annales 2012-2013 : Examen) - 5 points*)

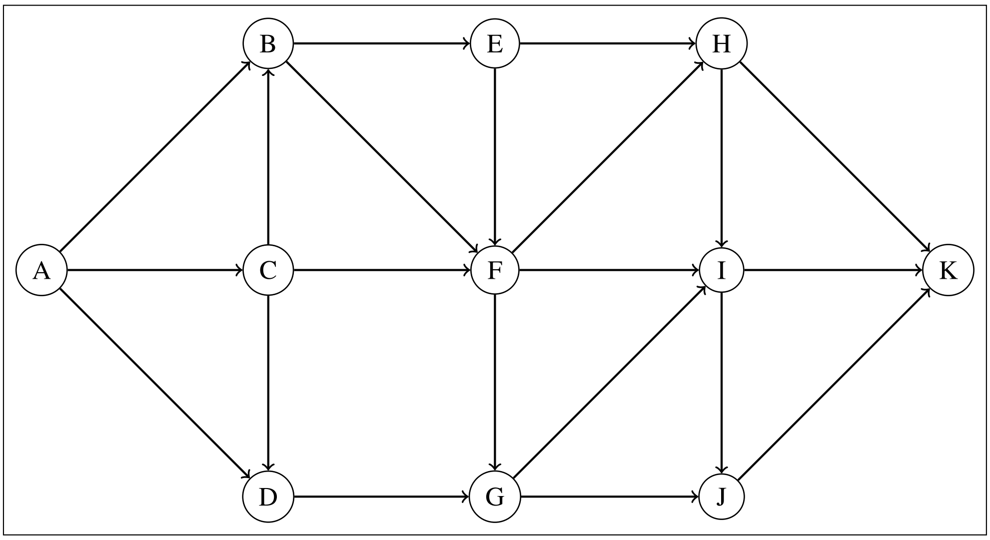


Figure7.4

Soit $\mathcal{G}$ le réseau de transport définit par le graphe 7.4 ci-dessus. Les sommets A et K sont respectivement la source et le puit du réseau. La capacité de tous les arc est de 5.
On définit le flot f par les valeurs suivantes :
$f(A, B) = 5$, $f(A, C) = 4$, $f(A, D) = 1$, $f(B, E) = 5$, $f(C, D) = 2$, $f(C, F) = 2$, $f(D, G) = 3$, $f(E, H) = 2$, $f(E, F) = 3$, $f(F, G) = 5$, $f(G, I) = 5$, $f(G, J) = 3$, $f(H, I) = 2$, $f(I, K) = 5$, $f(J, K) = 5$

Question #1 : Dessiner le flot f sur le graphe et donner la valeur du flot.

(solution page 124)

Question #2 : Soit $X = \{F, D, G\}$ et $Y = \{B, E, C\}$. Donner la capacité de X vers Y. Donner le flot de X vers Y.

(solution page 124)

Question #3 : Donner le réseau résiduel de $\mathcal{G}$ induit par le flot f.

(solution page 124)

Question #4 : A partir du flot f utiliser la méthode de Ford-Fulkerson pour obtenir le flot maximal du réseau $\mathcal{G}$.

(solution page 124)

Solution de l'exercice 7.II.2

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

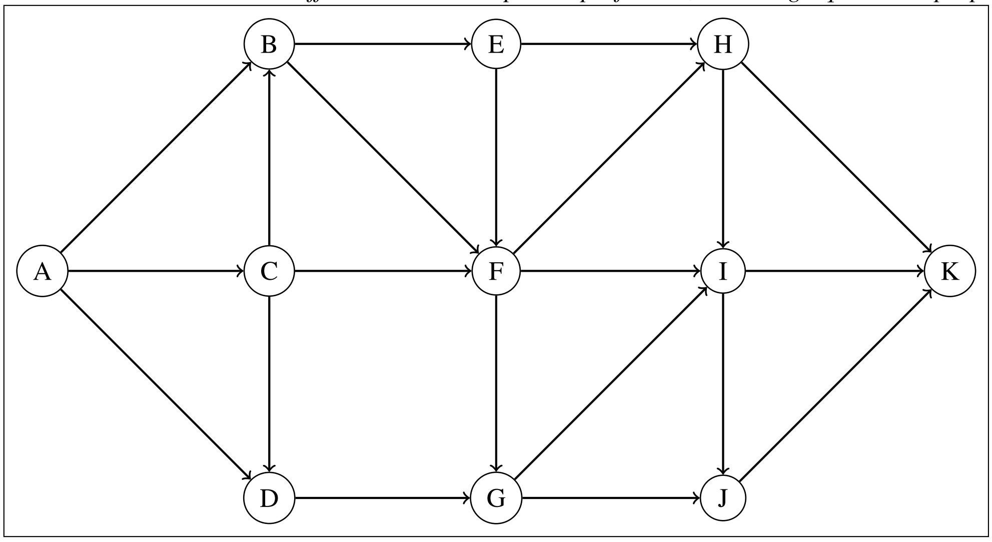


FIGURE7.4

Soit $\mathcal{G}$ le réseau de transport définit par le graphe 7.4 ci-dessus. Les sommets A **et** K **sont respectivement la source et le puit du réseau. La capacité de tous les arc est de** 5**.**
On définit le flot f **par les valeurs suivantes :**
$f(A,B) = 5$, $f(A,C) = 4$, $f(A,D) = 1$, $f(B,E) = 5$, $f(C,D) = 2$, $f(C,F) = 2$, $f(D,G) = 3$, $f(E,H) = 2$, $f(E,F) = 3$, $f(F,G) = 5$, $f(G,I) = 5$, $f(G,J) = 3$, $f(H,I) = 2$, $f(I,K) = 5$, $f(J,K) = 5$

1. **Dessiner le flot** f **sur le graphe et donner la valeur du flot.**

2. **Soit** $X = \{F, D, G\}$ **et** $Y = \{B, E, C\}$**. Donner la capacité de** X **vers** Y**. Donner le flot de** X **vers** Y**.**

3. **Donner le réseau résiduel de $\mathcal{G}$ induit par le flot** f**.**

4. **A partir du flot** f **utiliser la méthode de Ford-Fulkerson pour obtenir le flot maximal du réseau $\mathcal{G}$.**

Exercice 7.II.3 (*Exercice 2 (Annales 2012-2013 : Examen) - 4 points*)

Soit le graphe 7.6 suivant :

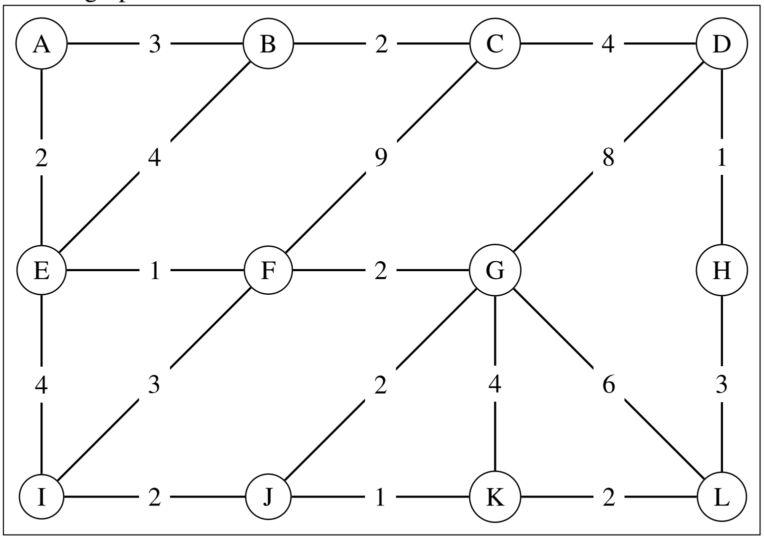


Figure7.6

Question #1 : Est-ce que ce graphe est eulérien ? Justifier.

(solution page 125)

Question #2 : Donner l'arbre couvrant minimum du graphe. Vous utilisez l'algorithme de Prim.

(solution page 125)

Solution de l'exercice 7.II.4

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.
Soit le graphe 7.6 suivant :

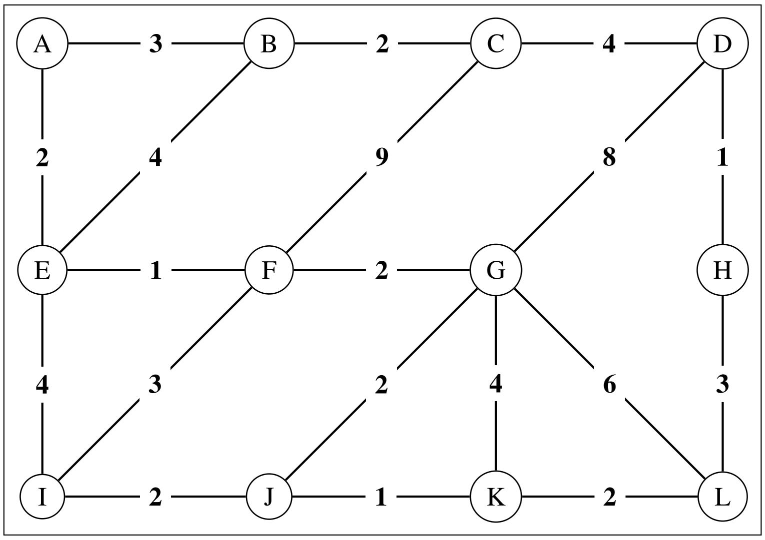


FIGURE7.6

1. **Est-ce que ce graphe est eulérien ? Justifier.**

2. **Donner l'arbre couvrant minimum du graphe. Vous utilisez l'algorithme de Prim.**

Exercice 7.II.5 (***Exercice 3 (Annales 2012-2013 : Examen) - 4 points***)

Tâche	Durée	Tâche Précédente
F	B	3
C	A H B	4
G	R	3
E	G S R	3
R	-	3
S	D F	2
D	R	2
H	B	4
A	D F R	5
B	-	4

TABLE7.3

Soit le problème d'ordonnancement représenté par le tableau 7.3 ci-dessus.

Question #1 : Calculez le rang de chaque tache. Dessinez le graphe représentant le projet.

(solution page 126)

Question #2 : Donner les dates au plus tard, au plus tôt et les marges de chaque tache. Quelle sera la durée minimum du projet ?

(solution page 126)

Solution de l'exercice 7.II.6

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

Tâche	Durée	Tâche Précédente
F	**B**	**3**
C	**A H B**	**4**
G	**R**	**3**
E	**G S R**	**3**
R	**-**	**3**
S	**D F**	**2**
D	**R**	**2**
H	**B**	**4**
A	**D F R**	**5**
B	**-**	**4**

TABLE7.3

Soit le problème d'ordonnancement représenté par le tableau 7.3 ci-dessus.

1. **Calculez le rang de chaque tache. Dessinez le graphe représentant le projet.**

2. **Donner les dates au plus tard, au plus tôt et les marges de chaque tache. Quelle sera la durée minimum du projet ?**

Exercice 7.II.7 (*Exercice 4 (Annales 2012-2013 : Examen) - 5 points*)

On modélise l'état d'un patient atteint d'un certain type cancer du poumon. Le cancer évolue à travers 3 stades notés S_1, S_2 et S_3 et commence en S_1. Chaque année, l'état du cancer peut évoluer.
En S_1, la probabilité de rester en S_1 est de $0,3$, la probabilité de guérison est de $0,2$ et la probabilité de passage en S_2 est de $0,5$.
En S_2, la probabilité de rester en S_2 est de $0,2$, la probabilité de guérison est de $0,1$ et la probabilité de passage en S_3 est de $0,7$.
En S_3, la probabilité de rester en S_3 est de $0,1$, la probabilité de guérison est de $0,1$ et la probabilité de décès $0,8$.

Question #1 : Modéliser l'état du cancer à l'aide d'une chaîne de Markov. Donner sa matrice de transition P et les états absorbants.

(solution page 127)

Question #2 : Donner la matrice de transition puissance 2 : P^2. Expliquez sa signification.

(solution page 127)

Partie #a :

Soit N la matrice fondamentale de P :

$$N = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{pmatrix} 0,49 & 0,89 & 0,69 \\ 0 & 1,25 & 0,97 \\ 0 & 0 & 1,11 \end{pmatrix} \end{array}$$

Question #1 : Donner le temps moyen qu'un patient reste malade (jusqu'à la guérison ou le décès). Expliquez votre réponse.

(solution page 127)

Question #2 : Quelles sont les probabilités qu'un patient venant de développer ce type de cancer soit guéri ou décède ? Expliquez votre réponse.

(solution page 127)

Solution de l'exercice 7.II.8

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

On modélise l'état d'un patient atteint d'un certain type cancer du poumon. Le cancer évolue à travers 3 stades notés S_1, S_2 et S_3 et commence en S_1. Chaque année, l'état du cancer peut évoluer.

En S_1, la probabilité de rester en S_1 est de $0,3$, la probabilité de guérison est de $0,2$ et la probabilité de passage en S_2 est de $0,5$.

En S_2, la probabilité de rester en S_2 est de $0,2$, la probabilité de guérison est de $0,1$ et la probabilité de passage en S_3 est de $0,7$.

En S_3, la probabilité de rester en S_3 est de $0,1$, la probabilité de guérison est de $0,1$ et la probabilité de décès $0,8$.

1. **Modéliser l'état du cancer à l'aide d'une chaîne de Markov. Donner sa matrice de transition P et les états absorbants.**

2. **Donner la matrice de transition puissance 2 : P^2. Expliquez sa signification.**

Soit N la matrice fondamentale de P :

$$N = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{pmatrix} 0,49 & 0,89 & 0,69 \\ 0 & 1,25 & 0,97 \\ 0 & 0 & 1,11 \end{pmatrix} \end{array}$$

1. **Donner le temps moyen qu'un patient reste malade (jusqu'à la guérison ou le décès). Expliquez votre réponse.**

2. **Quelles sont les probabilités qu'un patient venant de développer ce type de cancer soit guéri ou décède ? Expliquez votre réponse.**

Exercice 7.II.9 (*Exercice 5 (Annales 2012-2013 : Examen) - 2 points*)

Un groupe de personnes est tel que :
– chaque personne est membre d'exactement deux associations,
– chaque association comprend exactement trois membres,
– deux associations quelconques ont toujours exactement un membre en commun.

Combien y a-t-il de personnes ? d'associations ? Justifier votre réponse à l'aide d'un graphe. (solution page 127)

Solution de l'exercice 7.II.10

Cette solution n'est en rien officielle ni validée par un professeur, il ne s'agit que d'une proposition.

Un groupe de personnes est tel que :
– chaque personne est membre d'exactement deux associations,
– chaque association comprend exactement trois membres,
– deux associations quelconques ont toujours exactement un membre en commun.

Combien y a-t-il de personnes ? d'associations ? Justifier votre réponse à l'aide d'un graphe.