Énoncé des travaux dirigés

L3 Informatique

Algorithmique avancée

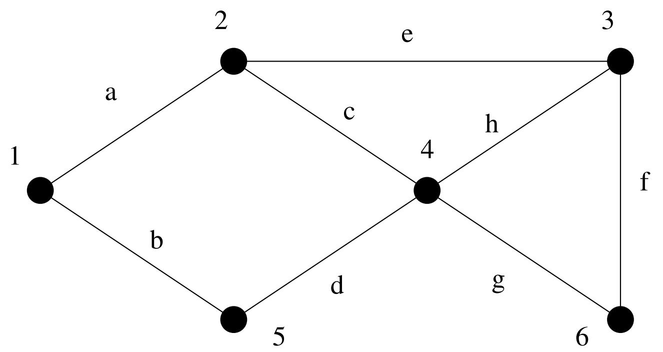
etienne.birmele@parisdescartes.fr

Automne 2016

Université Paris Descartes

TD1 - Graphes

Exercice 1.1. On considère le graphe de la figure suivante



Ecrire la matrice d'adjacence de ce graphe ainsi que son codage en liste d'arêtes et en listes d'adjacences.

Exercice 1.2. On considère un graphe G simple et non dirigé.

1. Soit A la matrice d'adjacence de G. Que valent les sommes des coefficients par lignes et par colonnes ?
2. Soit M la matrice d'incidence de G, c'est-à-dire une matrice de taille $|V(G)| \times |E(G)|$ telle que $m_{ue} = 1$ si l'arête e est incidente au sommet u et $m_{ue} = 0$ sinon. Ecrire la matrice d'incidence du graphe de l'exercice précédent.
3. Que valent les sommes des coefficients par lignes et par colonnes de M ?
4. Soit D la matrice diagonale de taille $n(G) \times n(G)$ dont le coefficient de la ligne correspondant au sommet u vaut le degré de u.
   Démontrer que $M^t M = A + D$.

Exercice 1.3. Montrer que tout groupe d'au moins deux personnes contient toujours au moins deux individus ayant le même nombre d'amis.

Exercice 1.4. Soit $(d_1, \dots, d_n)$ la suite des degrés d'un graphe non-dirigé G. On note $\delta(G)$ et $\Delta(G)$ respectivement le plus petit et le plus grand de ces degrés.

1. Montrer que $\sum_{i=1}^{n} d_i = 2m(G)$.
2. En déduire que $\delta(G) \leq \frac{2m(G)}{n(G)} \leq \Delta(G)$.

Exercice 1.5. Le but cet exercice est de montrer que deux plus long chemins dans un graphe non-orienté connexe G ont forcément un sommet commun.

1. Supposons par l'absurde que G contient deux plus long chemins P et Q qui sont disjoints. Soit $\mathcal{H}$ l'ensemble des chemins de G reliant un sommet de P et un sommet de Q.
   Justifier que $\mathcal{H}$ est non-vide puis qu'il contient un chemin H dont les sommets différents des extrémités n'appartiennent ni à P ni à Q.

2. En déduire que P et Q ont forcément un sommet commun.
3. Cette propriété est-elle encore vraie pour les graphes non connexes ? pour les graphes orientés faiblement connexes ?

Exercice 1.6. Un graphe non-orienté est 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. On note $n(A)$ et $n(B)$ le nombre de sommets de A et B.

1. Quel est le nombre maximal d'arêtes d'un graphe biparti simple ? Comment définiriez-vous la densité dans un problème ne concernant les graphes bipartis ?
2. Montrer que si G est biparti, tout cycle de G est forcément de longueur paire.
3. On suppose maintenant qu'un graphe G a tous ses cycles de longueur paire. Montrer que pour, toute paire de sommets u et v, tous les chemins entre u et v ont la même parité.

   Indication : On raisonnera par l'absurde et on considérera la paire des sommets (u, v) et la paire de chemins (P, Q) entre u et v tels que la somme des longueurs de P et Q est minimale.
4. Montrer que le graphe G de la question précédente est biparti. En déduire qu'un graphe admet une structure bipartie si et seulement si tous ses cycles sont de longueur paire.
5. Proposer un algorithme basé sur un BFS ou un DFS qui détecte si un graphe est biparti ou non.

Exercice 1.7. On considère un graphe non orienté G.

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

1. Soit M la matrice d'adjacence de G. Montrer par récurrence sur k que le coefficient (u, v) de M^k corrrespond au nombre de marches de longueur k entre u et v.
2. Montrer que le coefficient (u, v) de $(I + M)^k$ est non nul si et seulement si il existe $k' \leq k$ tel que le coefficient (u, v) de $M^{k'}$ est non nul.
3. Montrer que l'existence d'une marche entre u et v implique l'existence d'un chemin entre u et v.
4. En déduire que G est connexe si et seulement si tous les coefficients de $(I + M)^{n(G)}$ est non nul.
5. Quel est complexité de l'algorithme consistant à calculer cette matrice pour vérifier la connexité de G ? Implémenteriez-vous cet algorithme ?

TD2 : Arbres couvrants

Exercice 2.1. Une *forêt* est une réunion disjointe d'arbres. Proposer à partir du parcours en profondeur un procédé permettant de déterminer une forêt couvrante pour tout graphe G.

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.

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

Exercice 2.2. 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* de v tout sommet u tel qu'il existe un chemin orienté de u à v.

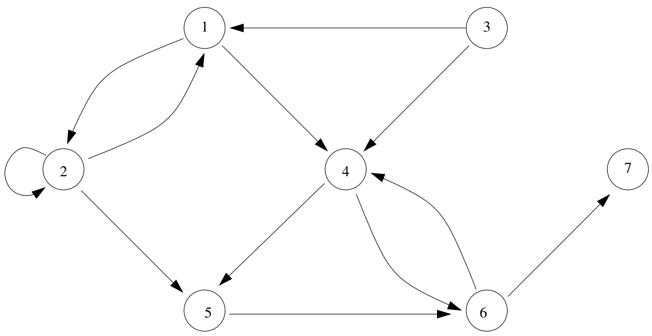


FIGURE 2.1 –

1. On considère le graphe de la figure 2.1. Déterminer l'ensemble des descendants et des ascendants du sommet 4.
2. Proposer deux façons de déterminer les ancêtres et les descendants d'un sommet. Quelle est la complexité de ces algorithmes ?

Exercice 2.3. 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 de la figure 2.2 comporte-t-il un ou des sommet(s) séparateur(s), et si oui le(s)quel(s) ?
2. 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 ?

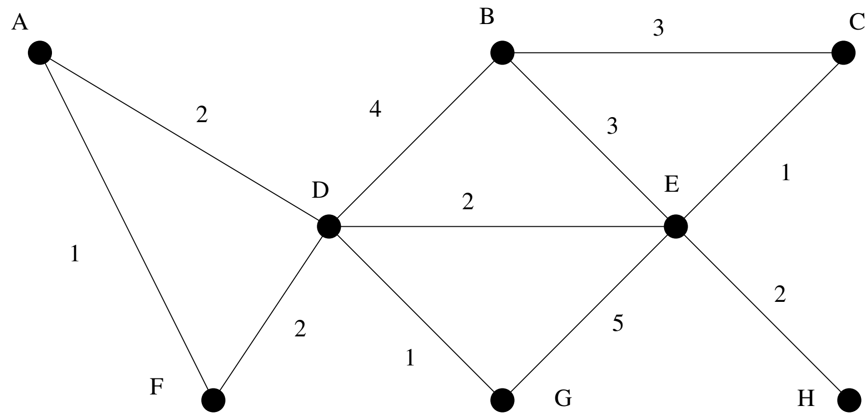


FIGURE 2.2 –

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. 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 en le nombre d'arêtes.

Exercice 2.4. On considère des graphes non-orienté, non-pondérés et connexes.

1. Soit v un sommet et T l'arbre obtenu en appliquant un parcours en largeur depuis v. Montrer que pour tout sommet w, le chemin de T reliant v à w est un plus court chemin entre v et w dans G. En déduire un algorithme permettant de déterminer le sommet le plus éloigné d'un sommet v.
2. En déduire un algorithme pour calculer le diamètre d'un graphe non pondéré. Quelle est sa complexité ?
3. Pour des très grands graphes, une complexité non linéaire peut être rédhibitoire. On considère l'algorithme consistant à

   Etape 1 Choisir un sommet v quelconque, lancer un BFS enraciné en v et choisir x comme le sommet le plus éloigné de v.

   Etape 2 Lancer un BFS enraciné en x et choisir y comme le sommet le plus éloigné de x.

   Etape 3 Retourner la distance x à y.

   Justifier le fait que cet algorithme est linéaire.

   Construire un contre-exemple montrant qu'il n'est pas exact.

   Montrer qu'il s'agit d'une 2-approximation, c'est-à-dire que $d(x,y) \leq diam(G) \leq 2d(x,y)$. On pourra pour cela comparer $d(x,y)$ et $diam(G)$ à $d(v,x)$.

Exercice 2.5. 1. Décrire 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. Reprendre la question précédente 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.

Exercice 2.6. Décrire un algorithme qui, étant donné un sommet v dans un graphe orienté, détermine un plus court cycle passant par v.

Indication : Lancer deux parcours en largeur.

Exercice 2.7. Déterminer un arbre couvrant de poids minimal pour le graphe de la figure 2.3

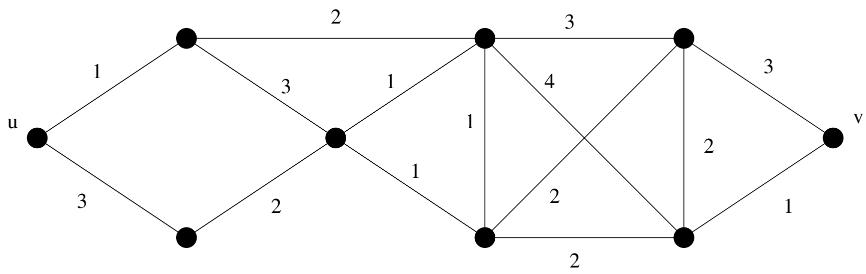


Figure 2.3 –

Exercice 2.8. On considère un graphe valué G et un ensemble U de sommets. Proposer un algorithme qui détermine un arbre couvrant de poids minimal parmi ceux tels que tous les sommets de U sont des feuilles.

Démontrer sa validité et préciser sa complexité.

Exercice 2.9. Appliquer l'algorithme de Dijkstra au graphe de la figure 1.2 pour déterminer le chemin de poids minimal entre u et v.

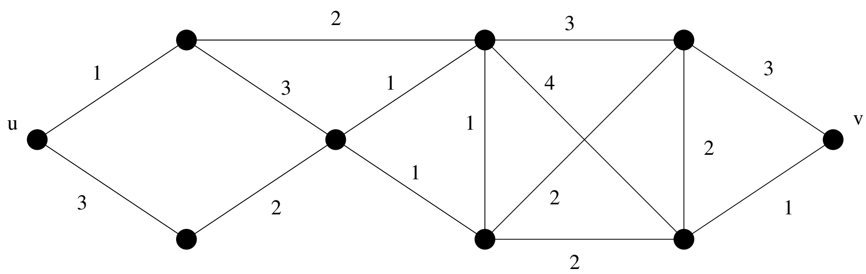


Figure 2.4 – .

Exercice 2.10. A 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 ?

TD3. Parcours de graphes : cycles

Exercice 3.1. Les graphes de la figure 2.4 contiennent-ils un chemin eulérien, un cycle eulérien, un chemin hamiltonien, un cycle hamiltonien ?

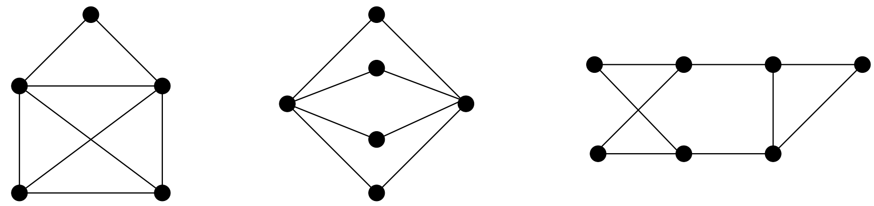


FIGURE 3.1 –

Exercice 3.2. Résoudre le problème du postier sur le graphe de la figure 3.2

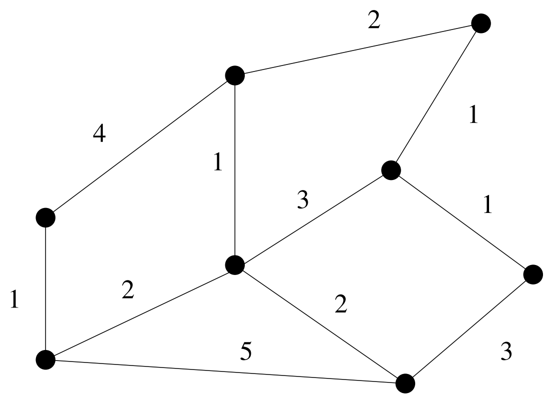


FIGURE 3.2 –

Exercice 3.3. Pour résoudre le problème du voyageur de commerce, on propose l'heuristique suivante :

Etape 1 construire un arbre couvrant de poids minimal ;

Etape 2 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 l'optimal.

TD4

Exercice 4.1. Déterminer tous les flots entiers du graphe suivant. En déduire un flot maximal.

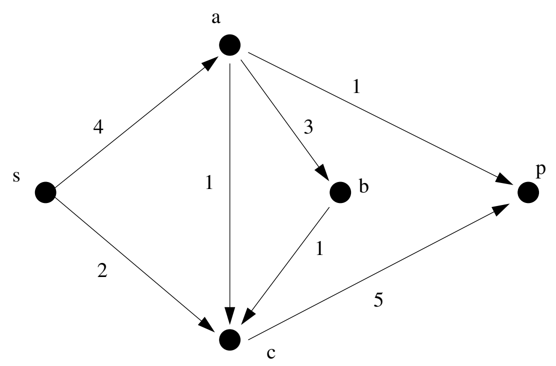


FIGURE 4.1 –

Déterminer toutes les coupes. En déduire une coupe minimale.

L'énumération de toutes les possibilités semble-t-elle une approche raisonnable pour de grands graphes ?

Exercice 4.2. Les produits des usines u_1 et u_2 doivent être acheminés vers les ports p_1, p_2 et p_3. Le nombre de tonnes pouvant être acheminés quotidiennement sont indiqués sur les routes de la figure 3.2.

1. Quelle est le tonnage maximal de produits pouvant être acheminée ?
2. Des crédits d'infrastructure sont disponibles. Sur quelles routes faut-il investir pour augmenter la capacité d'acheminement ?

Exercice 4.3. On considère un graphe orienté valué par des capacités, ayant un ensemble de sources S et de puits P. A chaque source $s_i \in S$ correspond une capacité d'émission $\sigma(s_i)$ et à chaque puits p correspond une demande de réception $\delta(p_i)$.

Un flot est dit *admissible* si,

$$\forall s_i \in S, \quad f^+(s_i) - f^-(s_i) \leq \sigma(s_i)$$

et

$$\forall p_i \in P, \quad f^-(p_i) - f^+(p_i) \geq \delta(p_i)$$

1. On construit un graphe H obtenu en ajoutant
   — une source s_0 et, pour tout s_i, l'arête (s_0, s_i) avec une capacité $\sigma(s_i)$;
   — un puits p_0 et, pour tout p_i, l'arête (p_i, p_0) avec une capacité $\delta(p_i)$.
   Construire le graphe H correspondant au graphe de la figure 3.2 avec $\sigma(u_1) = 11$, $\sigma(u_2) = 11$, $\delta(p_1) = 6$, $\delta(p_2) = 8$ et $\delta(p_3) = 7$.

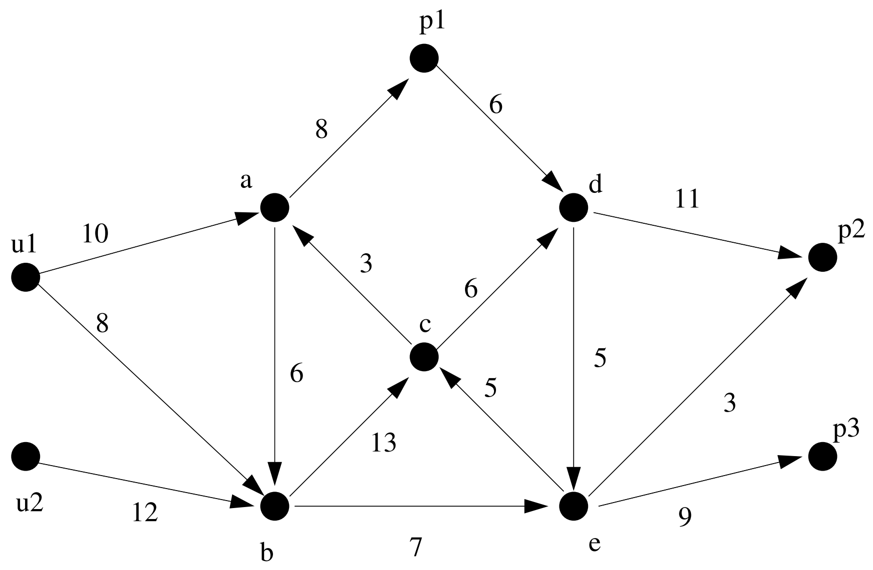


FIGURE 4.2 –

2. Montrer qu'un flot admissible existe dans G si et seulement si il existe un flot dans H qui sature toutes les arêtes de la forme (p_i, p_0).
3. Montrer qu'un tel flot est forcément maximal dans H.
4. Existe-t-il un flot admissible dans le graphe de la figure 3.2 pour les valeurs d'émission et de réception données à la question 1 ?

Exercice 4.4. A la suite d'une catastrophe naturelle, n personnes $I_1, \dots, I_n$ doivent être amenées dans k hôpitaux $H_1, \dots, H_k$. Chaque personne ne peut être amenée que dans les hôpitaux situés à moins d'un demi-heure de route (on part du principe que cette liste est connue pour chaque individu). De plus, on veut répartir les malades équitablement, à savoir en affecter au plus $\lceil \frac{n}{k} \rceil$ dans chaque hôpital. On considère un graphe G formé des sommets $I_1, \dots, I_n$ et $H_1, \dots, H_k$, avec une arête allant de I_i à H_j si la personne i peut être affectée à l'hôpital j. On lui ajoute un sommet s avec des arêtes de s vers chacun des I_i et un sommet p avec une arête de chacun des H_j vers p.

1. Dessiner un graphe correspondant à une instance du problème pour dix individus et trois hôpitaux.
2. On suppose dans un premier temps les capacités des arêtes incidentes à p infinies alors que toute les autres arêtes sont de capacité 1. A quoi correspond un flot entier de valeur n dans ce graphe ?
3. Comment affecter des capacités pour qu'une répartition équitable (au sens expliquée plus haut) des patients existe si et seulement si un flot suffisamment important entre s et p existe.
4. En déduire un algorithme répondant à la question des autorités sanitaires. Quelle est sa complexité ?

Exercice 4.5. Soit G un réseau de source 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.

1. Construire un graphe de flot maximal n tel qu'il soit possible de construire un flot maximal en un seul chemin augmentant ou en n tels chemins.

Un tel exemple montre qu'il est en pratique important de rechercher des chemins augmentant les plus efficaces possibles. Pour cela, on considère l'algorithme suivant, dit de flot maximal par échelonnement :

1 $C \leftarrow max_e c(e)$;
2 Initialiser le flot f à 0 ;
3 $K \leftarrow 2^{\lfloor log_2 C \rfloor}$;
4. tant que $K \geq 1$ faire
5   tant que il existe un chemin non saturé P de capacité résiduelle $\geq K$
6   augmenter le flot f le long de P ;
7     $K \leftarrow K/2$;
8 retourner f

2. Montrer qu'une coupe de G a une capacité au plus égale à Cm.
3. Pour un nombre K donné, montrer qu'un chemin non-saturé améliorant la capacité d'une valeur au moins égale à K peut être trouvé en temps $\mathcal{O}(m)$, si un tel chemin existe.
4. Montrer que l'algorithme par échelonnement calcule bien un flot entier maximum. L'appliquer au graphe de la figure 4.2.
5. Montrer que la capacité d'une coupe minimale du graphe résiduel vaut au plus $2Km$ à chaque exécution de la ligne 4 (Le graphe résiduel est l'instance dans laquelle on ne considère comme capacité de l'arête que la capacité non encore utilisée).
6. Montrer que la ligne 6 est exécutée $\mathcal{O}(m)$ fois pour chaque valeur de K.
7. En déduire une borne de complexité pour cet algorithme.

Travaux dirigés 5

TD5

Exercice 5.1. Pour chacune des matrices suivantes,

1. vérifier qu'il s'agit bien d'une matrice de transition d'une chaîne de Markov
2. représenter graphiquement la chaîne
3. déterminer l'ensemble des états récurrents et des états transients
4. déterminer l'ensemble des mesures invariantes.

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

Exercice 5.2. On considère la propagation d'une 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.
2. En déduire un moyen de calculer la probabilité que l'information soit encore vraie pour un nombre de pas tendant vers l'infini.
   Vers quoi tend cette valeur quand le nombre de relais tend vers l'infini ?

Exercice 5.3. On considère un graphe non-orienté connexe ayant un circuit de longueur p et un circuit de longueur l, p et l premiers entre eux. On considère une marche aléatoire équiprobable sur ce graphe (quand elle est sur un sommet de degré v, elle prend chaque arête possible avec probabilité $\frac{1}{d(v)}$).

1. Justifier l'unicité de la mesure invariante μ pour cette chaîne de Markov.
2. Montrer que pour tout état i,

$$\mu(i) = \sum_{j \in N(i)} \frac{1}{d(j)} \mu(j)$$

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