Université Paris Diderot – Paris 7
L3 Informatique

Algorithmique
Semaine du 22 octobre 2012

corrigé du TD n°5

1 Connexité

Exercice 1 $1 \Rightarrow 2$ Considérons une bipartition. On prend $x \in V_1$ et $y \in V_2$. Un chemin les relie d'après 1. Ce chemin part de V_1 et s'achève en V_2 donc au moins une arête travserve la bipartition.

$2 \Rightarrow 1$. Considérons deux sommets x et y. On veut construire un chemin les reliant. On regarde la bipartition $V_1 = \{x\}$ et $V_2 =$ le reste. D'après 2. au moins une arête traverse. Ses deux extrémités sont x et, disons, x_2. On construit une deuxième bipartition $V_1 = \{x, x_2\}$ et $V_2 =$ le reste. Une arête la traverse et nous donne un troisième sommet x_3. On construit ainsi une suite de bipartitions. Le graphe étant fini, au bout d'un moment on trouve $x_i = y$. Pour exhiber le chemin il suffit de regarder, comme dans un parcours, quel sommet x_i a fait rentrer x_j au temps j dans la partie V_1 et de dire que v_i est le père de v_j.

2 Degré

Exercice 2 Il suffit de remarquer que chaque arête uv compte deux fois dans la somme : une fois dans $deg(u)$ et une fois dans $deg(v)$. Un arc uv lui ne compte qu'une fois, dans $deg(u)$. Cette preuve devient une récurrence : on ajoute les arêtes (resp. arcs) une à une, à chaque fois m augmente de 1 et la somme de 2 (resp. 1).

Exercice 3

1. Vrai. Notons $I = \{v \mid d(v) \text{ impair }\}$ l'ensemble des sommets de degré impair et $P = S \setminus I$ l'ensemble des sommets de degré pair. On peut ainsi décomposer la somme des degrés des sommets d'un graphe :

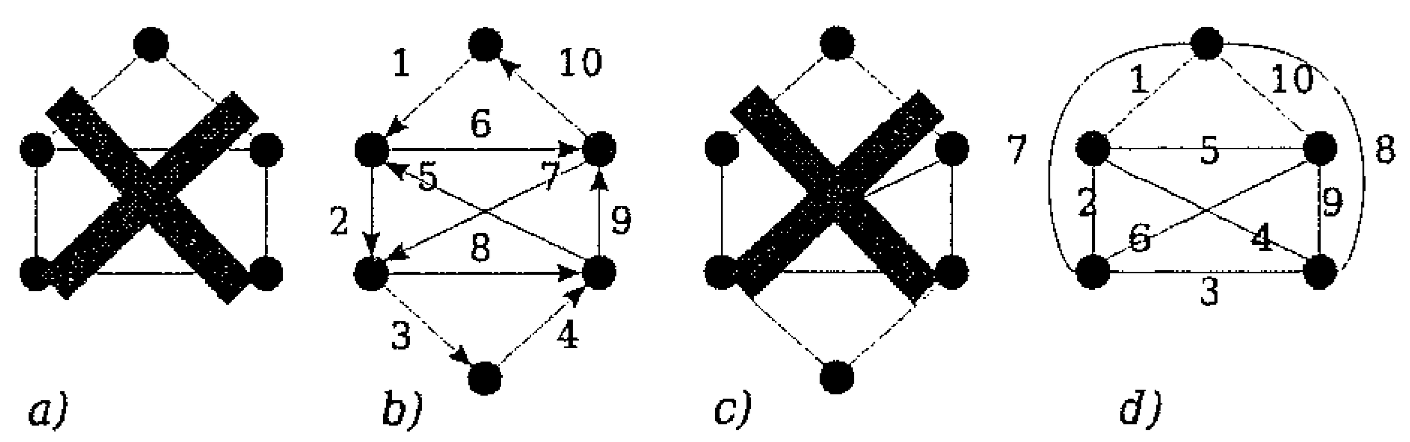
$$\sum_{v \in V} d(v) = \sum_{v \in I} d(v) + \sum_{v \in P} d(v)$$

Nous avons déjà montré que $\sum_{v \in V} d(v)$ est pair. Or $\sum_{v \in P} d(v)$ est pair : c'est la somme d'entiers tous pairs. Ainsi $\sum_{v \in I} d(v) = \sum_{v \in V} d(v) - \sum_{v \in P} d(v)$ est pair. Or la somme d'un nombre impair d'entiers impairs est toujours impaire : on en déduit que le nombre de sommets de degré impair doit être pair.

2. Faux. Un simple carré (cycle de 4 sommets) infirme l'énoncée.

3 Euler et les ponts de Königsberg

Exercice 4



a) b) c) d)

Exercice 5 Un cycle "rentre" autant de fois dans un sommet qu'il en "sort". Si le cycle est simple (emprunte chaque arête au plus une fois) et passe k fois par un sommet alors il touche $2k$ arêtes. Si il passe par toutes les arêtes du sommet (cas d'un cycle eulerien) alors son degré est $2k$, pair.

Exercice 6 Si eulerien, alors le long du cycle on peut atteindre tout sommet depuis tout autre, le graphe est donc connexe.

Exercice 7 Si la marche stoppe sur un sommet *qui n'est pas le sommet de départ* alors, si on est rentré k fois dans le sommet, on en est sorti $k-1$ fois (puisque là on ne peut plus en sortir). Comme on a vu toutes les arêtes son degré est $2k-1$, impair. Ce qui n'est pas possible. Donc on stoppe sur le sommet de départ ($k-1$ entrées pour $k-1$ sorties, c'est pair).

Exercice 8 Il suffit de dessiner un graphe à 5 sommets : deux triangles ayant un côté en commun, et de faire une marche qui bloque dans le triangle de gauche.

Exercice 9

Eulerien(G : graphe à n sommets, r : sommet de départ);
début
  C_1 = marche(G,r)
  si *C_1 n'est pas un cycle (sommet fin $\neq r$)* **alors**
    STOP : le graphe n'est pas eulerien (sommet fin a degré impair)
  si *il reste des arêtes hors de C_1* **alors**
    Soit $G' = G - C_1$ (on enlève aussi les sommets déconnectés, de degré 0)
    si *il n'existe aucun sommet de G' apparaissant dans C_1* **alors**
      STOP : le graphe n'est pas eulerien (pas connexe)
    Soit s un sommet de G' apparaissant dans C_1
    C_2 = Eulerien(G', s)
    C = Fusionner(C_1, C_2, s)
  sinon
    $C = C_1$
  retourner (C)
fin
Marche(G : graphe à n sommets, r : sommet de départ)
début
  C=liste vide
  ... voisin s tel que rs non marquée **faire**

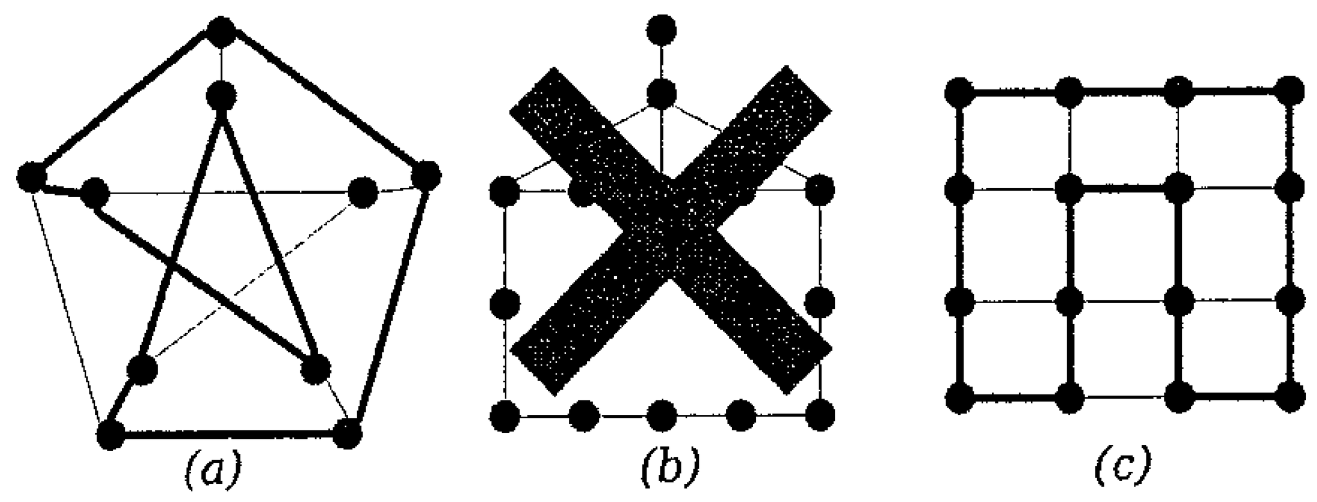
2. **il retourne un cycle.** Si C_1 n'est pas un cycle il stoppe, mais dans ce cas par l'exercice 6 le sommet d'arêt a degré impair. Donc C_1 esy un cycle ; et la fusion de deux cycles (C_1 et C_2) ayant un sommet en commun (y) est un cycle.
3. **Ce cycle est simple** car une marche passe au plus une fois par arête et ensuite les arêtes sont enlevées en cas d'appel récursif
4. **il contient toute les arêtes.** En effet si une arête est "oubliée" à la fin (pas dans C, donc ni dans C_1 ni dans C_2), c'est qu'elle est dans G' lors de tous les appels. G' n'est donc jamais vide. Or l'algorithme a terminé (et renvoyé C) sans erreur, donc G' est vide, contradiction.
5. On a donc un cycle eulerien

Cette preuve est constructive.

4 La variante de Sir William Rowan Hamilton

Exercice 11 Comme le cycle passe une et une seule fois par tous les sommets, la taille du cycle est n.

Exercice 12



- le graphe (a) n'est pas hamiltonien, mais il existe des cycles à $n-1$ sommets.
- le graphe (b) n'est pas hamiltonien à cause du sommet tout en haut. En effet ce dernier a un degré égal à 1, et tout cycle passant par ce sommet passera deux fois par le sommet juste en dessous.
- le graphe (c) est hamiltonien comme le montre le cycle sur le dessin ci-dessus.

Exercice 13

La recherche de cycle hamiltonien est un problème connu pour être NP-complet. Plus simplement cela veut dire que l'on ne sait pas à l'heure actuelle s'il existe un algorithme polynomial qui teste si le graphe est hamiltonien. Une façon simple (mais pas la plus efficace !) de trouver un cycle hamiltonien consiste à examiner toutes les permutations σ des sommets, et à tester ensuite si la permutation σ correspond à un cycle. Or il existe $n!$ permutations possibles sur n sommets. Donc en utilisant la célèbre formule de Stirling :

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n (1 + \Theta\left(\frac{1}{n}\right))$$