Numéro d'anonymat :

M1 Informatique.
UE MOGPL.

Examen du 18 janvier 2019. Durée : 2 heures

Une feuille recto-verso est autorisée, tout autre document est interdit.
Téléphones portables éteints et rangés dans vos sacs.
Le barème est indicatif et est susceptible d'être modifié.

Exercice 1 (8 points)

On considère dans cet exercice un graphe orienté où chaque arc (u, v) a une longueur notée $\ell(u, v)$. Nous cherchons dans ce graphe un chemin de s à t dont la longueur **moyenne** des arcs est minimale. Par exemple, si un chemin a 3 arcs de longueur 2, 6 et 4, la longueur moyenne de ses arcs est $\frac{2+6+4}{3} = 4$.

On se place dans un premier temps dans le cas d'un graphe sans circuit, et l'on cherche à établir un algorithme de programmation dynamique.

Question 1 (1.5/8) — Expliquez en terme de principe d'optimalité pourquoi l'algorithme de Bellman (directement adapté à ce critère de longueur moyenne) ne permet pas de résoudre le problème.

Question 2 (3/8) — On note $\lambda(j, v)$ la longueur minimale d'un chemin de s à v comportant exactement j arcs ($\lambda(j, v) = \infty$ si un tel chemin n'existe pas). Explicitez un algorithme de programmation dynamique basé sur le calcul des $\lambda(j, v)$ permettant de trouver la longueur moyenne optimale (longueur moyenne des arcs d'un chemin de s à t de longueur moyenne minimale). On n'omettra pas de préciser l'initialisation, ainsi que la manière de calculer la valeur optimale.

Donnez la complexité de l'algorithme.

Question 3 (2/8) — Appliquez l'algorithme sur le graphe G_0 de la figure ci-après. On donnera simplement la valeur des $\lambda(j, v)$ (sous forme de tableau), la valeur (longueur moyenne) optimale, ainsi qu'un chemin optimal.

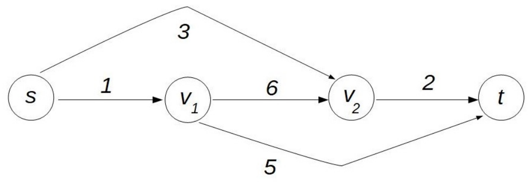


Figure 1 – Graphe G_0

Question 4 (1.5/8) — On considère dans cette question non plus des graphes sans circuit mais des graphes généraux, et l'on s'intéresse à l'existence d'une solution optimale. Donnez un exemple d'un graphe où les longueurs des arcs sont toutes positives mais qui ne possède pas de chemin (fini) de longueur moyenne minimale.

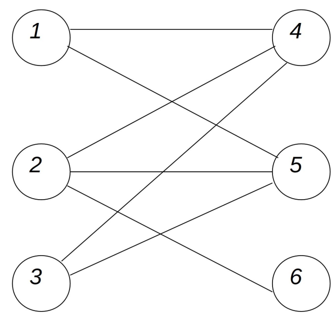
Exercice 2 (8 points)

Rappel : Un couplage d'un graphe est un ensemble d'arêtes deux-à-deux non adjacentes. Un couplage est dit parfait si chaque sommet du graphe est extrémité d'une arête du couplage. Par exemple, sur le graphe ci-dessous :

— L'ensemble d'arêtes $\{(1,4),(2,5)\}$ est un couplage ;

— L'ensemble d'arêtes $\{(1,4),(1,5)\}$ n'en est pas un ;

— L'ensemble d'arêtes $\{(1,5),(2,6),(3,4)\}$ est un couplage parfait.



Considérons un graphe biparti $G = (V, E)$, où $V = L \cup R$ ($L \cap R = \emptyset$) et toute arête a une extrémité dans L et l'autre dans R. Nous supposons dans tout l'exercice que $|L| = |R|$. Le but de l'exerice est de montrer à l'aide de propriétés sur les flots/coupes le théorème suivant :

Théorème : G a un couplage parfait si et seulement si pour tout $A \subset L$, $|N(A)| \geq |A|$

où $N(A)$ désigne l'ensemble des voisins des sommets de A ($N(\{1,3\}) = \{4,5\}$ dans l'exemple précédent).

Question 1 (4/8) — Exemple introductif

On rappelle qu'on associe classiquement à un graphe biparti G le réseau de transport H construit à partir de G de la manière suivante :

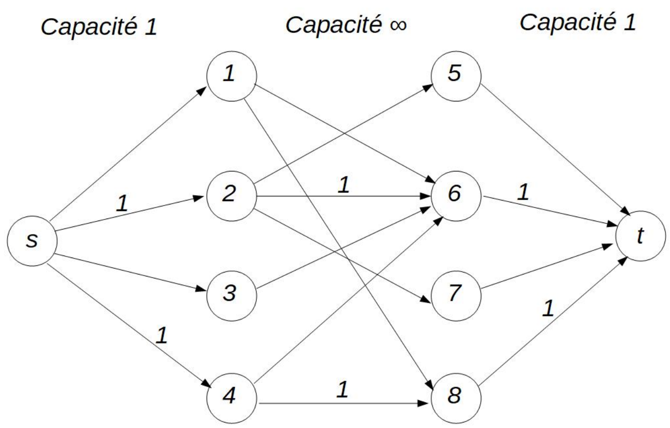
— On ajoute deux sommets s et t ; il y a un arc de capacité 1 de s à chaque sommet de L, un arc de capacité 1 de chaque sommet de R à t.

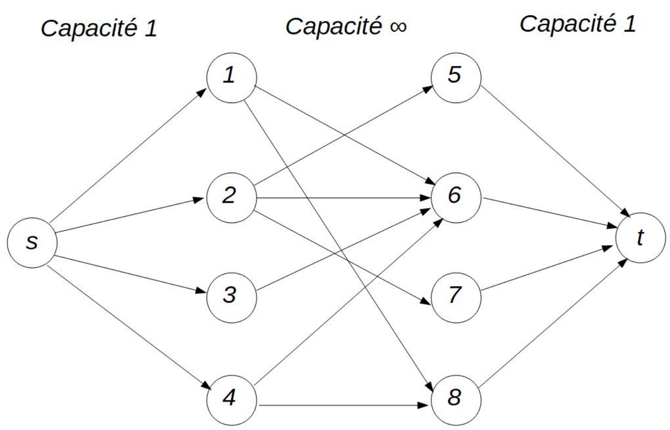
— Chaque arête (i,j) de G ($i \in L$, $j \in R$) correspond à un arc (i,j) de capacité infinie dans H.

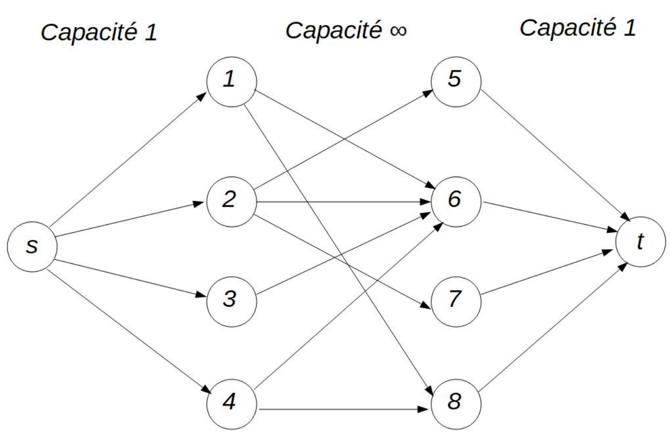
1. Calculez un flot maximum dans le réseau H_1 de la figure ci-après en appliquant l'algorithme de Ford et Fulkerson à partir du flot donné sur la figure (les arcs sans marque ont un flux 0). Vous préciserez à chaque étape la chaîne augmentante trouvée. Utilisez des/les copies du réseau pour le déroulement de l'algorithme (il peut y en avoir plus que nécessaire).

   En déduire une coupe de capacité minimum.

Réseau H_1


Capacité 1
Capacité ∞
Capacité 1
s
1
2
3
4
5
6
7
8
t
1
1
1
1
1
1


Capacité 1
Capacité ∞
Capacité 1
s
1
2
3
4
5
6
7
8
t


Capacité 1
Capacité ∞
Capacité 1
s
1
2
3
4
5
6
7
8
t

2. En déduire que le graphe G_1 ci-dessous n'admet pas de couplage parfait.

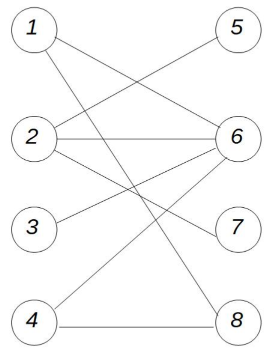


Figure 2 – Graphe G_1

3. Donner un ensemble $A \subset L$ ($L = \{1, 2, 3, 4\}$) tel que $|A| > |N(A)|$.

Les questions 2 et 3 ont pour but de montrer le théorème pré-cité.

Question 2 (1/8) — Soit M un couplage parfait d'un graphe biparti, et soit $A \subset L$. Expliquez (brièvement) pourquoi $|N(A)| \geq |A|$.

Question 3 (3/8) — Le but de cette question est de montrer la réciproque : *si pour tout* $A \subset L$ $|N(A)| \geq |A|$, *alors* G *admet un couplage parfait.*

Nous supposons donc dans cette question que G est un graphe biparti (avec $|L| = |R|$) tel que pour tout $A \subset L$ $|N(A)| \geq |A|$.

1. Considérons une coupe (S, T) ($s \in S$, $t \in T$) du réseau H associé à G (cf la construction rappelée au début de la question 1), et notons $A = S \cap L$ et $B = S \cap R$. Montrez que la capacité de cette coupe est au moins $|L|$. On pourra dans un premier temps montrer que la coupe est de capacité au moins $|L| - |A| + |B|$.

2. En déduire que G a un couplage parfait.

Exercice 3 (4 points)

On considère un problème d'affectation de n tâches $t_1, \dots, t_n$ à n agents $a_1, \dots, a_n$ (chaque tâche doit être affectée à un et un seul agent, chaque agent doit avoir une et une seule tâche à effectuer). Le coût d'affecter la tâche t_i à l'agent a_j est $c_{i,j}$.

Dans le problème de l'affectation classique (vu en cours), la valeur de la solution consistant à affecter la tâche i à l'agent $\sigma(i)$ est la somme des coûts $\sum_{i=1}^{n} c_{i,\sigma(i)}$. On cherche à minimiser cette valeur.

Dans le problème de l'affectation *bottleneck*, la valeur de la solution consistant à affecter la tâche i à l'agent $\sigma(i)$ est le coût maximum $\max_{i=1}^{n} c_{i,\sigma(i)}$. On cherche également à minimiser cette valeur.

Question 1 (1.5/4) — Donnez un exemple avec $n = 2$ où les solutions optimales des deux problèmes (*bottleneck* et classique) sont différentes.

Question 2 (2.5/4) — Etant donné un nombre M, expliquez comment déterminer s'il existe une affectation *bottleneck* de valeur au plus M. Montrez alors comment résoudre efficacement le problème de l'affectation *bottleneck* en résolvant un certain nombre (que l'on précisera) de fois un problème de flot maximum ou d'affectation classique.