Buhl Alexandre
Note finale: 13/20 (barème: 11.5/18)

+354/1/14+

Examen de THEG

Mars 2018, S6, ING1. Durée : 1 heure

— Aucun document ni appareil électronique autorisé.
— Noircir les cases au stylo (pas de crayon à papier) et sans déborder sur les voisines car la correction est automatisée.
— Certaines réponses incorrectes apportent des points négatifs. Dans le doute, s'abstenir.
— Marquez toutes les réponses correctes dans les questions marquées avec ♣.
— Lorsqu'une réponse numérique demande plusieurs chiffres, les chiffres sont lus de haut en bas.

Prénom, NOM

Alexandre BUHL

UID :
[ ]0 [ ]1 [x]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]8 [ ]9
[x]0 [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]8 [ ]9
[ ]0 [ ]1 [x]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]8 [ ]9
[ ]0 [ ]1 [x]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]8 [ ]9
[ ]0 [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [x]6 [ ]7 [ ]8 [ ]9

1 Rayon et diamètre

Soit $G = (V, E, w)$ un graphe connexe où chaque arête $e \in E$ est pondérée par une longueur $w(e) \geq 0$. Pour deux sommets v_1 et $v_2 \in V$, on note $d(v_1, v_2)$ la longueur du plus court chemin les reliant. L'excentricité du sommet $v \in V$, notée $exc(v)$, est sa distance au sommet le plus éloigné :

$$exc(v) = \max\{d(v, u) \mid u \in V\}.$$

Le rayon de G, noté $r(G)$, est la valeur de la plus petite excentricité, tandis que le diamètre $D(G)$ est la plus grande :

$$r(G) = \min\{exc(v) \mid v \in V\}$$
$$D(G) = \max\{exc(v) \mid v \in V\}$$

Question 1 ♣ Soit M la matrice des distances calculée en appliquant l'algorithme de Floyd-Warshall sur G.

- [x] $D(G)$ est la plus grande valeur de M.
- [ ] $r(G)$ est la plus petite valeur de M.
- [x] $D(G)$ s'obtient en calculant d'abord le minimum m_i pour chaque ligne i de M, puis en retournant le maximum de ces m_i.
- [ ] $r(G)$ s'obtient en calculant d'abord le maximum m_i pour chaque ligne i de M, puis en retournant le minimum de ces m_i.
- [ ] On a toujours $D(G) = 2 \times r(G)$.
- [x] On a toujours $r(G) \leq D(G) \leq 2 \times r(G)$.
- [ ] On a toujours $r(G) < D(G) < 2 \times r(G)$.
- [ ] $D(G) = r(G)$ si et seulement si G est complet.
- [ ] $D(G) = 2 \times r(G)$ si et seulement si G est complet.

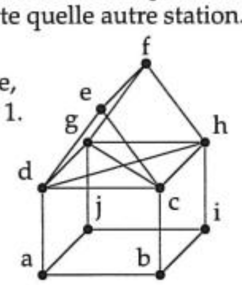
1.5/3

Question 2 ♣ On considère le graphe $G_1 = (V_1, E_1, w_1)$ où, V_1 représente les stations du métro parisien, E_1 relie les stations voisines sur une ligne de métro, et w_1 donne le temps (supposé constant) de parcours entre deux stations voisines. On néglige les coûts de correspondance.

- [x] $D(G_1)$ est un temps suffisant pour aller de l'importe quel endroit à n'importe quel autre.
- [ ] $D(G_1)$ est la durée de la plus longue balade que l'on puisse faire dans le métro sans passer deux fois au même endroit.
- [x] Il existe une ou plusieurs stations dont l'excentricité est égale à $D(G_1)$.
- [x] Il existe une ou plusieurs stations dont l'excentricité est égale à $r(G_1)$.
- [ ] L'excentricité d'une station donne le temps maximum pour rejoindre n'importe quelle autre station.
- [ ] L'excentricité d'une station donne le temps minimum pour rejoindre n'importe quelle autre station.

4.5/4

On considère le graphe G_2 ci-contre, où toutes les arêtes ont pour poids 1.



Question 3 Le rayon de G_2 est

[ ]0 [ ]1 [x]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]8 [ ]9

1/1

Question 4 Le diamètre de G_2 est

[ ]0 [ ]1 [ ]2 [x]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]8 [ ]9

1/1

2 Coloration Gloutone

Le *nombre chromatique* d'un graphe $G = (V, E)$ est le nombre de couleurs minimum nécessaire pour colorier les sommets du graphe de façon à ce que deux sommets voisins ne partagent pas la même couleur.

Question 5 Quel est le nombre chromatique du graphe G_2 de la question 3 ?

1/1 ☐0 ☐1 ☐2 ☐3 ☒4 ☐5 ☐6 ☐7 ☐8 ☐9

L'algorithme suivant calcule un coloriage des sommets du graphe. Les sommets sont parcourus dans un ordre donné en argument, et coloriés par la première couleur disponible et non-utilisée par les voisins déjà coloriés. Pour un sommet $x \in V$, on note $adj(x)$ l'ensemble des sommets voisins : $adj(x) = \{y \mid (x, y) \in E\}$, qu'on suppose stockés sous la forme d'une liste d'adjacence.

```
GreedyColor(G = (V,E), σ)
Entrée : un graphe G, un ordre sur les sommets σ
         (σ est une permutation de V)
Sortie : un tableau de couleurs C indicé par les sommets
       // marquer toutes les couleurs comme disponibles
 1     for each c ∈ {1,...,|V|}
 2         Avail[c] ← 1
       // initialement les sommets ne sont pas coloriés
 3     for each x ∈ V
 4         C[x] ← 0
 5     for each x in σ :
           // repérage des couleurs voisines
 6         for each y ∈ adj(x) :
 7             Avail[C[y]] ← 0
           // recherche de la première couleur libre
 8         i ← 1
 9         while Avail[i] = 0
10             i ← i + 1
           // affectation de la couleur trouvée
11         C[x] ← i
           // remise à disposition des couleurs voisines
12         for each y ∈ adj(x) :
13             Avail[C[y]] ← 1
14     return C
```

Dans cet algorithme, les couleurs sont désignées par des numéros de 1 à $|V|$. La valeur $C[x]$ donne la couleur du sommet x, ou 0 s'il n'est pas colorié. Le tableau *Avail* est utilisé pour repérer les couleurs utilisées par les sommets voisins, afin de pouvoir trouver la première couleur inutilisée (notez que *Avail*[0] peut changer de valeur aux lignes 7 et 13, mais ne sera jamais lu, la boucle de la ligne 9 commençant à l'indice 1).

Question 6 Combien de fois la ligne 7 est-elle exécutée précisément ?

☐ $\lfloor\frac{\lvert V\rvert}{2}\rfloor$	☐ $\lvert V\rvert+1$	☐ $\lceil\frac{\lvert E\rvert}{2}\rceil$	☐ $\lvert E\rvert^2$
☐ $\lceil\frac{\lvert V\rvert}{2}\rceil$	☐ $\lvert V\rvert^2$	☐ $\lvert E\rvert-1$	☒ $2\times\lvert E\rvert$
☐ $\lvert V\rvert-1$	☐ $2\times\lvert V\rvert$	☐ $\lvert E\rvert$	☐ $\lvert V\rvert\cdot\lvert E\rvert$
☐ $\lvert V\rvert$	☐ $\lfloor\frac{\lvert E\rvert}{2}\rfloor$	☐ $\lvert E\rvert+1$	☐ $\lvert V\rvert+\lvert E\rvert$

0/2

Question 7 Sachant que le nombre total d'exécution de la ligne 10 ne peut dépasser $|E|$, donnez la complexité de l'algorithme GreedyColor ? (Attention, le graphe peut ne pas être connexe.)

☐ $O(\lvert V\rvert^2)$	☐ $O(\lvert V\rvert+\lvert E\rvert)$	☐ $\Theta(\lvert E\rvert^2)$
☐ $\Theta(\lvert E\rvert)$	☐ $\Theta(\lvert V\rvert)$	☒ $\Theta(\lvert V\rvert+\lvert E\rvert)$
☐ $O(\lvert E\rvert)$	☐ $\Theta(\lvert V\rvert^2)$	☐ $O(\lvert V\rvert\cdot\lvert E\rvert)$
☐ $\Theta(\lvert V\rvert\cdot\lvert E\rvert)$	☐ $O(\lvert V\rvert)$	☐ $O(\lvert E\rvert^2)$

0/2

3 Divers

Question 8

On considère le graphe non-orienté dont voici la matrice d'adjacence. Combien possède-t-il d'arbres couvrants différents ?

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

☐0 ☐1 ☒2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☒4 ☐5 ☐6 ☐7 ☐8 ☐9 0/2

Question 9 L'algorithme de Bellman-Ford pour la recherche de plus court chemin...

☐ ... fonctionne uniquement sur des graphes non pondérés.

☒ ... fonctionne dans des graphes pondérés uniquement s'il n'y a pas de cycle de somme négative. 2/2

☐ ... fonctionne dans des graphes pondérés uniquement si les poids sont tous positifs.

Question 10 ♣ L'algorithme d'Edmonds vu en cours permet :

☒ De calculer un couplage parfait.

☒ De calculer un couplage maximum. 0.5/2

☒ De calculer un couplage maximal.