Mardi 28 mai 2013 G. Aupy / L. Martinet / E. Thierry

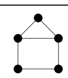
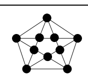
Algorithmique 2 - Examen (3 heures)

Identifiant : mettre ici un signe distinctif (dessin, nombre, ...) et reporter ce signe sur la copie anonyme pour les associer.

Consignes : *Aucun document autorisé.* L'évaluation de votre copie tiendra compte de la présentation. Pour les analyses de complexité, on évaluera la complexité en temps dans le pire cas. Par défaut les graphes sont sans boucles, ni arêtes/arcs multiples. Pour toutes les questions algorithmiques, on supposera que les graphes ou arbres en entrée sont donnés par leurs listes d'adjacence. Le nombre de sommets (resp. arêtes ou arcs) d'un graphe sera noté n (resp. m).

Exercice 1. QCM (11 pts)

Mode d'emploi : Pour chacune des questions, entourer la (ou les) réponse(s) justes (il en a toujours au moins une de juste). Barême par question : +1/3 si la (ou les) réponse(s) justes sont toutes entourées, -1/3 si une réponse incorrecte est entourée, 0 sinon. Pour les questions de complexité, on attend des réponses correspondant aux connaissances vues en cours/TD (par exemple ne pas s'autoriser l'hypothèse $P = NP$). Si plusieurs complexités incomparables sont solutions, les entourer toutes. Et si plusieurs sont comparables, entourer la meilleure. Dans les questions sur les graphes, pour simplifier l'écriture des complexités, on suppose $m = \Omega(n)$.

	QUESTION	REPONSE
1	La proposition suivante est-elle vraie : *"si S est un ensemble indépendant de $G = (V, E)$, alors $V-S$ est une clique de G"*?	OUI NON
2	La proposition suivante est-elle vraie : *"S est un ensemble indépendant de $G = (V, E)$ si et seulement si $V-S$ est un ensemble couvrant les arêtes de G"*?	OUI NON
3	Dans une classe de 9 élèves, chaque élève envoie des cartes de St Valentin à 2 autres élèves de son choix. Est-il possible que chaque élève recoive des cartes des 2 élèves à qui il a écrit ?	OUI NON
4	Même question en remplaçant 2 par 3 ?	OUI NON
5	Lesquels de ces graphes admettent un cycle eulérien ?	  K_{2n}
6	Quel est le nombre minimum d'arêtes d'un graphe connexe à n sommets ne contenant pas de sous-graphe induit isomorphe à ?	$\lfloor \frac{n}{2} \rfloor$ $n-1$ $\lfloor \frac{n^2}{4} \rfloor$ $\frac{n(n-1)}{2}$
7	Avec quelle complexité sait-on détecter si un graphe non orienté est connexe ?	$\mathcal{O}(n)$ $\mathcal{O}(m)$ $\mathcal{O}(n^2)$ $\mathcal{O}(n^3)$
8	Avec quelle complexité sait-on détecter si un graphe orienté contient un cycle ?	$\mathcal{O}(n)$ $\mathcal{O}(m)$ $\mathcal{O}(n^2)$ NP-complet
9	Avec quelle complexité sait-on détecter si un graphe non orienté contient un cycle ?	$\mathcal{O}(n)$ $\mathcal{O}(m)$ $\mathcal{O}(n^2)$ NP-complet
10	Avec quelle complexité sait-on détecter si un graphe non orienté contient un cycle qui passe par tous les sommets une fois et une seule ?	$\mathcal{O}(n)$ $\mathcal{O}(m)$ $\mathcal{O}(n^2)$ NP-complet
11	Avec quelle complexité sait-on détecter si un graphe non orienté contient un cycle qui passe par toutes les arêtes une fois et une seule ?	$\mathcal{O}(n)$ $\mathcal{O}(m)$ $\mathcal{O}(n^2)$ NP-complet
12	Avec quelle complexité sait-on calculer un arbre couvrant de poids minimum dans un graphe non orienté connexe pondéré ?	$\mathcal{O}(m)$ $\mathcal{O}(m \log n)$ $\mathcal{O}(n^2)$
13	Idem quand toutes les arêtes ont des poids égaux à 1 ?	$\mathcal{O}(m)$ $\mathcal{O}(m \log n)$ $\mathcal{O}(n^2)$

	QUESTION	REPONSE
14	Avec quelle complexité sait-on calculer les distances à partir d'une source dans un graphe orienté dont les arcs ont des poids positifs ?	$\mathcal{O}(m)$ $\mathcal{O}(m\log n)$ $\mathcal{O}(n^2)$
15	Idem quand tous les arcs ont des poids égaux à 1 ?	$\mathcal{O}(m)$ $\mathcal{O}(m\log n)$ $\mathcal{O}(n^2)$
16	Avec quelle complexité sait-on calculer les distances à partir d'une source dans un graphe orienté sans cycle de poids strictement négatif ?	$\mathcal{O}(m)$ $\mathcal{O}(n^2)$ $\mathcal{O}(mn)$
17	Avec quelle complexité sait-on calculer les distances pour tous les couples de sommets dans un graphe orienté dont les arcs ont des poids positifs ?	$\mathcal{O}(m)$ $\mathcal{O}(n^2)$ $\mathcal{O}(n^3)$ $\mathcal{O}(n^4)$
18	Avec quelle complexité sait-on vérifier si un graphe est biparti ?	$\mathcal{O}(m)$ $\mathcal{O}(n^2)$ $\mathcal{O}(mn)$
19	Tout arbre admet au moins un couplage parfait ?	OUI NON
20	Etant donné un graphe biparti, avec quelle complexité sait-on calculer un couplage de card max ?	$\mathcal{O}(m)$ $\mathcal{O}(nm)$ $\mathcal{O}(nm^2)$ NP-dur
21	Quelle est la meilleure complexité avec laquelle on sait calculer le cardinal maximum d'un indépendant pour un graphe non orienté quelconque ?	$\mathcal{O}(n)$ $\mathcal{O}(n^2)$ NP-dur
22	Quelle est la meilleure complexité avec laquelle on sait calculer le cardinal maximum d'un indépendant pour un arbre ?	$\mathcal{O}(n)$ $\mathcal{O}(n^2)$ NP-dur
23	Un mini-gala réunit 5 filles et 5 garçons. Sachant que chaque fille a déjà fait connaissance de exactement 3 des 5 garçons et chaque garçon a déjà fait connaissance de exactement 3 des 5 filles, est-il toujours possible de répartir les 10 en 5 couples (fille,garçon) disjoints tel que chaque couple réunit deux personnes se connaissant déjà ?	OUI NON
24	Dans un réseau de flot à capacités dans $\mathbb{Q}_+$, avec quelle complexité sait-on calculer un flot maximum ?	$\mathcal{O}(m)$ $\mathcal{O}(nm)$ $\mathcal{O}(nm^2)$ NP-dur
25	Dans un réseau de flot, introduire des capacités sur les sommets et des capacités inférieures et supérieures sur les arcs ne change pas la complexité en $\mathcal{O}$ avec laquelle on sait calculer un flot maximum.	OUI NON
26	Etant donné un problème d'optimisation où l'espace des solutions valides est convexe et la fonction de coût est aussi convexe, un minimum local est-il toujours un minimum global ?	OUI NON
27	Soit un mot $u \in \Sigma^n$ où Σ alphabet fini fixé, avec quelle complexité sait-on calculer $bord(u)$ le plus long préfixe propre de u aussi suffixe de u ?	$\mathcal{O}(n)$ $\mathcal{O}(n\log n)$ $\mathcal{O}(n^2)$
28	Soient deux mots $u \in \Sigma^n$ et $v \in \Sigma^m$ où Σ alphabet fini fixé, avec quelle complexité sait-on trouver toutes les occurences de u dans v ?	$\mathcal{O}(m+n)$ $\mathcal{O}(m\log n)$ $\mathcal{O}(mn)$
29	Avec un tas binaire contenant n éléments, avec quelle complexité sait-on gérer l'opération d'insertion d'un nouvel élément ?	$\mathcal{O}(1)$ $\mathcal{O}(\log n)$ $\mathcal{O}(n)$
30	Les files de priorités sont des structures de données conçues pour traiter les opérations de fusion d'ensembles ?	OUI NON
31	Partant d'une partition en singletons d'un ensemble à n éléments auxquels on peut accèder en temps constant, avec quelle complexité sait-on traiter un total de u opérations d'union de parties et f requètes de recherche de la partie contenant un élément ?	$\mathcal{O}(f+u)$ $\mathcal{O}(f+u\log u)$ $\mathcal{O}(fu)$ $\mathcal{O}(u+f\log u)$
32	Avec quelle complexité sait-on trier lexicographiquement n mots sur un alphabet fini fixé, de longueurs respectives $\ell_1,\ldots,\ell_n$?	$\mathcal{O}(n+\max_i \ell_i)$ $\mathcal{O}(\sum_i \ell_i)$ $\mathcal{O}(n\sum_i \ell_i)$
33	Avec quelle complexité sait-on prétraiter un arbre enraciné à n sommets, pour répondre ensuite en temps $\mathcal{O}(1)$ aux requêtes de premier ancètre commun (LCA) ?	$\mathcal{O}(n)$ $\mathcal{O}(n^2)$ $\mathcal{O}(\exp(n))$

Exercice 2. **Des algorithmes (4 pts)**

1. Ecrire un algorithme de complexité linéaire qui prend en entrée un graphe orienté $G = (V, E)$ donné par ses listes d'adjacences $N^+(x)$, $x \in V$, et qui détecte s'il est fortement connexe. Justifier brièvement sa correction et sa complexité.

Consigne : pour cette question, si vous souhaitez utiliser des algorithmes de cours/TD, vous devez les réécrire complétement, et non pas juste les invoquer.

2. Soit $G = (V, E)$ un graphe orienté sur $V = \{1, \ldots, n\}$ avec arcs pondérés dans $\mathbb{Q}$, stockés dans une matrice W de taille $n \times n$: $W[i, j]$ vaut le poids de l'arc ij s'il existe, ou $+\infty$ sinon. On suppose qu'il n'existe pas de cycle de poids strictement négatif.

Ecrire un algorithme en $\mathcal{O}(n^3)$ qui calcule les distances (pondérées) pour tous les couples de sommets (i, j) (par convention cette distance vaudra $+\infty$ s'il n'existe pas de chemin de i à j).

Consigne : pour cette question, si vous souhaitez utiliser des algorithmes de cours/TD, vous devez les réécrire complétement, et non pas juste les invoquer.

3. Un ensemble de p laboratoires envoie des chercheurs à une conférence ; le i-ème laboratoire envoie m_i chercheurs. En tant qu'organisateur de la conférence vous avez prévu q groupes de travail qui se dérouleront simultanément ; le j-ème groupe pourra accueillir au plus n_j participants. Vous voulez répartir les chercheurs entre les différents groupes pour que les chercheurs d'un même laboratoire se retrouvent dans des groupes tous différents. Par contre, les groupes ne doivent pas forcément être tous remplis jusqu'à leur capacité maximum.

Pouvez-vous déterminer en temps polynomial si une telle répartition est possible ? Si oui, préciser la complexité que vous proposez.

Consigne : si vous utilisez des algorithmes vus en cours/TD, vous n'avez pas besoin de les re-détailler dans cette question.

Exercice 3. **Des théorèmes (3 pts)**

1. Enoncer (sans le démontrer) le théorème de Hall sur les couplages dans les graphes bipartis.

2. Un étudiant dispose d'un algorithme en temps linéaire qui prend en entrée un mot M et calcule pour tout $1 \leq i \leq |M|$, $\beta[i] = |bord(M[1..i])|$ où $bord(M[1..i])$ est le bord max du préfixe de longueur i de M.

Soient P et T deux mots sur un alphabet fixé Σ, un exercice demande un algorithme linéaire pour trouver toutes les occurences de P comme facteur de T. L'étudiant propose de calculer β pour le mot PT (concaténation de P et T), puis d'annoncer "nouvelle occurence de P" pour tous les indices i tels que $\beta[i] = |P|$ et $2|P| \leq i \leq |P| + |T|$. A-t-il raison ? Si oui, démontrer la correction de son algo. Si non, donner un contre-exemple et proposer une modification simple qui corrige sa solution.

3. Démontrer le théorème de Ford-Fulkerson, à savoir l'équivalence des trois assertions suivantes pour un flot f dans un réseau $R = (V, E, c)$ de source s et puits t :

(1) f est un flot maximum,

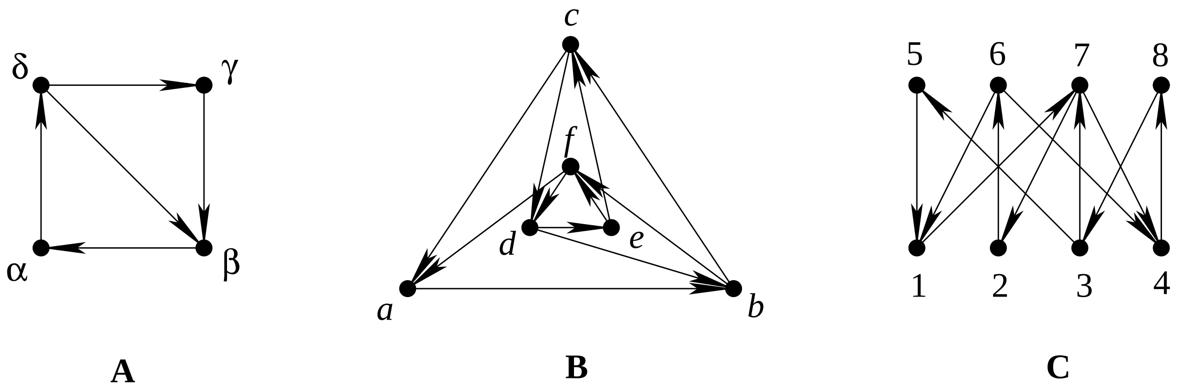
(2) il n'existe pas de chemin améliorant,

(3) il existe une coupe $(X, \overline{X})$ telle que $|f| = c(X, \overline{X})$.

Exercice 4. **Période d'un graphe (2 pts)**

La *période* $per(G)$ d'un graphe orienté $G = (V, E)$ *fortement connexe* est le PGCD des longueurs de tous ses cycles. Ce paramètre joue un rôle important par exemple en théorie des automates, en probabilités et dans l'étude de systèmes à événements discrets.

1. Donner la période de chacun des trois graphes fortement connexes suivants (en argumentant votre réponse) :



2. L'algorithme suivant est proposé pour calculer la période d'un graphe orienté $G = (V, E)$ que l'on suppose fortement connexe :

1. Construire un arbre T, sous-graphe de G, enraciné depuis un sommet s, couvrant tous les sommets.
2. Pour tout arc $xy \in E \setminus T$, de x vers y de profondeurs respectives p_x et p_y dans T, associer l'entier $\delta_{xy} = p_y - p_x - 1$.
3. Retourner $\text{pgcd}\{\delta_{xy} \mid xy \in E \setminus T\}$.

Démontrer que cet algorithme est correct. Expliquer comment l'implémenter efficacement et analyser la complexité obtenue.