

Examen de graphes - I3 - 2 heures - sans documents

Le barème est donné à titre indicatif.

1. Graphes sans circuits et plus courts chemins (≈ 8 Pts)

Soit $G = (E, \vec{\Gamma})$ un graphe (orienté) sans circuit (GSC). On rappelle que, pour tout sommet x dans E , le *rang de x* , noté $r(x)$, est le nombre d'arcs d'un chemin élémentaire maximal (*i.e.*, comportant le maximum d'arcs) se terminant en x .

Soit $i \in E$, on suppose que le sommet i est une racine de G .

A (0.5 Pt) Rappeler la définition d'une racine et d'une source.

B (1.5 Pts) Démontrez que dans un GSC, une racine est nécessairement une source. Donnez un contre-exemple montrant que l'inverse n'est pas vrai. Quel est le rang d'une source ?

Soit une application ℓ de $\vec{\Gamma}$ dans $\mathbb{R}$ qui, à chaque arc $u \in \vec{\Gamma}$, fait correspondre un réel appelé *longueur de l'arc u* . On se propose de calculer, pour tout sommet $x \in E$, la longueur $\pi(x)$ d'un plus court chemin de i à x dans G .

C (0.5 Pt) En observant que tout chemin de i à x passe nécessairement par l'un des prédécesseurs de x , et qu'un chemin extrait d'un plus court chemin est lui aussi un plus court chemin, donnez une relation liant $\pi(x)$ aux valeurs $\pi(y)$ pour les prédécesseurs y de x .

D (2.5 Pts) Démontrez que dans un GSC, pour tout sommet x et pour tout prédécesseur y de x , on a $r(x) > r(y)$. Donnez un contre-exemple montrant que cette proposition n'est pas vraie dans le cas d'un graphe comportant un circuit.

On en déduit que dans le cas d'un GSC, on peut calculer les valeurs de π en appliquant la relation de la question **C** pour tous les sommets de G pris par ordre de rang croissant.

E (1 Pt) Quel algorithme permet (indirectement) de calculer les rangs des sommets d'un GSC ? Rappelez brièvement son principe (il n'est pas demandé de pseudo-code). Quel est sa complexité ?

F (2 Pts) On suppose qu'on a calculé les rangs des sommets de G et qu'on a trié les sommets par ordre de rang croissant : on a $E = \{x_1, \dots, x_n\}$ avec $r(x_1) \leq \dots \leq r(x_n)$. Donnez, en pseudo-code, un algorithme linéaire en temps de calcul permettant de calculer les valeurs $\pi(x)$ pour tous les sommets x .

2. Ordonnancement (≈ 7 Pts)

On se place dans la perspective de l'aide à la gestion de grands projets, comme la construction d'une fusée ou d'un bâtiment, la réalisation d'un logiciel complexe, etc.

Soit un *objectif* à atteindre dont la réalisation suppose l'exécution de *tâches* soumises à des *contraintes* de précédence, du type : il faut que les tâches M, N, O, P soit terminées pour que la tâche Q puisse commencer.

Pour chaque tâche, on a une estimation de la durée. Le problème est de déterminer l'ordre et les dates d'exécution des tâches afin de minimiser le temps global du projet, ainsi que d'aider à la surveillance des tâches et l'allocation des ressources.

On modélise le projet par un graphe $G = (E, \vec{\Gamma})$, dont les sommets représentent des tâches. Il existe un arc $v = (t_1, t_2)$ dans $\vec{\Gamma}$ si la tâche t_1 doit être terminée pour que la tâche t_2 puisse commencer. A chaque arc $v = (t_1, t_2)$, une application ℓ fait correspondre la durée estimée de la tâche t_1 . On ajoute de plus à E deux tâches fictives α et ω de durées nulles, représentant respectivement le début et la fin du projet.

A (0.5 Pt) Quelle propriété doit posséder le graphe $G = (E, \vec{\Gamma})$ représentant un projet pour que ce projet soit réalisable ? Justifiez votre réponse.

B (1 Pt) On cherche à déterminer, pour une tâche quelconque t , à quelle date cette tâche pourra débiter au plus tôt, compte tenu des contraintes de précédence qui la lient aux autres tâches. On note $d(t)$ cette *date au plus tôt pour la tâche t* , en prenant pour convention $d(\alpha) = 0$. Comment exprimer cette date au plus tôt en termes de longueurs de chemins dans le graphe G ? Justifiez votre réponse. **Note : il est conseillé de traiter en parallèle la question E qui suit pour illustrer les définitions.**

C (0.5 Pt) Comment exprimer la durée minimum du projet dans son ensemble ?

D (1 Pt) Si une tâche t démarre en réalité à une date $d' > d(t)$, cela ne retarde pas forcément la date de fin du projet. Par contre, pour toute tâche t il existe une date $D(t)$, nommée *date au plus tard pour t* , telle que si t démarre après $D(t)$ alors la fin du projet est retardée d'autant. Comment exprimer cette date au plus tard en termes de longueurs de chemins dans le graphe G ? Justifiez votre réponse.

E (3 Pts) Application : dessinez le graphe associé au projet dont les données sont décrites ci-dessous, et pour chaque tâche indiquez les dates au plus tôt et au plus tard. Quelles sont les tâches critiques (*i.e.*, les tâches t telles que $d(t) = D(t)$) ? Comment le chef de projet peut-il exploiter cette information ?

F (1 Pt) Quel(s) algorithme(s) peut-on employer pour calculer automatiquement les dates au plus tôt et au plus tard ? Justifiez votre réponse. Complexité ?

Tâche	Durée	Précédences
1	10	
2	5	
3	7	
4	1	1
5	4	1,2
6	8	2,3
7	10	4,5,6
8	9	6

3. Cocktail (≈ 3 Pts)

Voici une affirmation : “dans toute réunion mondaine, il y a au moins deux personnes ayant le même nombre d’amis présents” (on suppose la relation d’amitié symétrique).

A (0.5 Pt) Dans le cadre des graphes (finis), à quelle propriété correspond cette affirmation ?

B (2.5 Pts) Cette propriété est elle fausse ou vraie ? Donnez une preuve de ce que vous avancez.

4. Arborescence de recherche (≈ 3 Pts)

Soit D un ensemble appelé *domaine*, muni d’une relation d’ordre total. Soit $X \subset D$ et $n \in D$. La question que l’on se pose est de savoir si n appartient à l’ensemble X . Dans la suite, on considèrera un domaine $D \subset \mathbb{N}$ et l’ordre habituel sur les entiers naturels.

A (0.5 Pt) Quelle est la complexité du test $n \in X$ si X est représenté sous forme d’une liste (non triée) ? Quelle est la complexité de ce test si X est représenté sous forme d’un tableau de booléens ? Quelle est la taille du tableau en question ?

Nous allons représenter X sous forme d’une arborescence binaire, c’est-à-dire que le nombre de successeurs de tout sommet est compris entre 0 et 2.

On impose de plus les conditions suivantes :

- l’arborescence est ordonnée (on distingue pour chaque sommet son successeur “droit” et son successeur “gauche”).
- pour tout sommet x , toute valeur située dans la sous-arborescence gauche de x est inférieure à la valeur associée à x , et toute valeur située dans la sous-arborescence droite de x est supérieure à la valeur associée à x .

Une arborescence binaire respectant les conditions ci-dessus est appelée **arborescence de recherche**.

Pour une recherche efficace, il faut de plus que l'arborescence soit équilibrée, c'est-à-dire que pour chaque sommet les tailles des arborescences gauche et droite soient comparables.

B (1 Pt) Dessinez une arborescence de recherche équilibrée représentant l'ensemble $X = \{9, 7, 10, 13, 14, 11, 4\}$.

C (1.5 Pts) Proposez un algorithme permettant de tester la présence d'une valeur donnée v dans une arborescence de recherche.