

IN325T2, Algorithmes et structures de données
Licence d'informatique
durée 2h

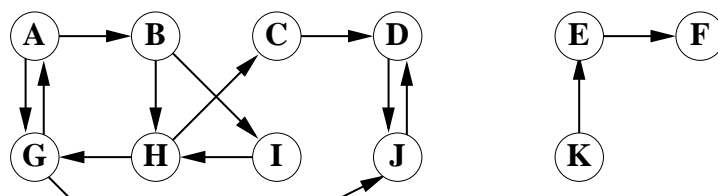
Les notes de cours et TD sont autorisées.

Les deux parties A et B sont à rendre sur des copies séparées.

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

1 Partie A

1.1 Exercice 1.



Question 1.1. On considère le parcours en profondeur du graphe orienté ci-dessus (on suppose que les successeurs sont traités dans l'ordre alphabétique). Donner la suite des sommets parcourus dans l'ordre de première visite, lorsque l'on part du sommet A.

Question 1.2. Dessiner la forêt couvrante associée au parcours effectué. Donner aussi la représentation de la forêt couvrante sous forme d'arbre fils-frère.

Question 1.3. Ecrire en pseudo-code une fonction qui effectue le parcours en profondeur d'un graphe quelconque \mathcal{G} et qui simultanément construit, sous sa **représentation fils-frère**, la forêt couvrante associée au parcours.

1.2 Exercice 2.

On considère un ensemble S d'entiers distincts et un intervalle $[c, d]$. On représente l'ensemble S par un arbre binaire de recherche (ABR). On veut déterminer les éléments de S dans l'intervalle $[c, d]$.

Question 1.4. Ecrire en pseudo-code une fonction Recherche-Intervalle (A ABR, c entier, d entier) qui affiche tous les éléments x de l'ABR A tels que $c \leq x \leq d$.

On souhaite généraliser ce problème à la dimension 2. A savoir, étant donné un ensemble T de points du plan et un rectangle R , on veut déterminer les points de T qui sont dans le rectangle R . Pour simplifier, on se limite à des rectangles dont les côtés sont parallèles aux axes.

Pour représenter l'ensemble T de points du plan, on utilise une version généralisée d'ABR : un ABR-Point. Un ABR-Point est un arbre binaire tel que chaque étiquette est un point (x, y) et où, lors d'une recherche ou d'une insertion, la discrimination se fait au niveau 0 (i.e. à la racine) selon l'abscisse x , au niveau 1 selon l'ordonnée y , et de façon générale : aux niveaux pairs selon l'abscisse x et aux niveaux impairs selon l'ordonnée y . Plus précisément : sur un noeud de niveau pair, on prend à gauche si le point à insérer a une abscisse inférieure à l'abscisse de ce noeud et sinon à droite ; sur un noeud de niveau impair, on prend à gauche si le point à insérer a une ordonnée inférieure à l'ordonnée de ce noeud et sinon à droite.

Question 1.5. Dessiner l'ABR-Point obtenu par l'insertion successive des points $(5, 3)$, $(1, 4)$, $(6, 8)$, $(7, 5)$, $(2, 5)$, $(3, 2)$, $(4, 1)$.

On représentera les points par des enregistrements à deux champs : *abs* et *ord*. Et un noeud de l'ABR-Point sera décrit par un enregistrement à trois champs : *val* de type point et *gauche* et *droit* deux pointeurs sur les fils gauche et droit.

Question 1.6. Ecrire en pseudo-code une fonction *Insérer* qui ajoute un point P dans un ABR-Point A .

Question 1.7. Ecrire une fonction *Recherche-Rectangle* qui affiche tous les points P de l'ABR-Point A contenu dans le rectangle dont le coin inférieur gauche est C et le coin supérieur droit est D .

2 Partie B (A rendre sur une copie séparée. Noter le numéro de table)

2.1 Hypothèses et objectifs

On se place dans un réseau de sites communicants par messages, et on suppose que les lignes de communications sont fiables. On suppose de plus que les sites ont des identités deux à deux distinctes et, pour simplifier, on supposera que ces identités sont les nombres de 1 à N , N étant bien entendu le nombre de sites du réseau. Le réseau est de plus supposé connexe (contrairement à celui de la partie A-1 du présent examen).

Chaque site est susceptible de lancer, s'il le désire, un algorithme réparti ayant pour but de créer sur l'ensemble du réseau un arbre couvrant dont il sera la racine. Les différents sites du réseau devront mémoriser leur appartenance à cet arbre, l'identité de l'initiateur étant présente dans cette mémorisation, afin de ne pas mélanger les appartenances à plusieurs arborescences couvrantes dans le cas où plusieurs sites du réseau auront décidé de lancer un tel algorithme.

Question 2.1. Précisez quel(s) type(s) d'algorithme(s) on peut utiliser pour créer de tels arbres couvrants, et ce que devront comporter les messages relatifs à cet ou ces algorithme(s) pour distinguer les différents initiateurs possibles.

2.2 Structures de données

Plus précisément, l'appartenance d'un site à un arbre couvrant sera mémorisée par une structure, de type *memo-arbre*, composée de plusieurs champs, élément d'un tableau indexé par les identités des sites du réseau; ainsi, sur un site quelconque, la structure *arbre[n]* mémorisera l'ensemble des informations correspondants à l'arbre couvrant dont le site n est la racine.

En plus des informations indispensables relatives à la mémorisation de cet arbre, on désire mémoriser dans une chaîne de caractères la position du site dans l'arbre, de la manière suivante :

- le site initiateur (racine) utilise simplement sa propre identité. Exemple : 1
- les fils immédiats de l'initiateur ajoutent à l'identité de l'initiateur un suffixe -1 ou -2, ...etc selon leur position (1er fils ou 2ème fils, ...etc). ceci donne donc des chaînes de caractères telles que 1-1, 1-2, ...etc.
- Ainsi de suite, par exemple le 2ème fils du 3ème fils du 1er fils du site 4 sera identifié par la chaîne de caractères : 4-1-3-2.

La notion de premier fils, deuxième fils, ... sera simplement liée à une numérotation par chaque site de ses lignes de communication. Ainsi chaque site qui envoie un message vers un voisin pourra y inclure, sous une forme ou une autre, le numéro de ligne qui le relie à ce voisin. Les fils d'un site ne seront d'ailleurs pas forcément numérotés de manière contiguë, car certaines lignes ne serviront pas dans les relations Père fils.

Question 2.2. *Définir les champs de la structure de type memo-arbre, en précisant leur type, leur rôle et leurs valeurs initiales. Il n'est pas interdit d'ajouter aux champs indispensables un ou des champs auxiliaires qui serviront dans l'algorithme de parcours, voir la section Algorithme.*

Question 2.3. *En supposant que l'algorithme ait permis de mettre à jour tous les champs concernés, quelles informations peut-on en déduire sur la profondeur de l'arbre? Que faudrait-il faire pour que l'initiateur connaisse cette profondeur? (répondez en une simple phrase donnant le principe, pas avec un algorithme complet).*

2.3 Algorithme

Question 2.4. *Ecrire un algorithme réparti permettant à tout site de lancer la création de l'arbre couvrant dont il sera racine, à tout site du réseau de recevoir les messages appropriés et de mémoriser, ainsi qu'il a été défini plus haut, son appartenance et sa position dans l'arbre construit.*

N.B. : Le choix de l'algorithme précis à utiliser est laissé à votre appréciation, mais il faut au moins que chaque site connaisse à la fin son père, ses fils, ainsi que sa position mémorisée au moyen de la chaîne de caractères définie plus haut. On ne demande pas que l'initiateur soit averti de la terminaison, ni connaisse la profondeur de l'arbre comme demandé à la section précédente.

Si vous n'avez pas le temps d'écrire complètement l'algorithme, essayez d'indiquer vos idées plutôt que des morceaux de (pseudo)code incomplets.