251 810 J.1443-C

ÉPREUVE B

L'épreuve comporte deux problèmes indépendants. Le premier problème étudie un algorithme de tri tandis que le deuxième problème s'intéresse aux monoïdes et aux mots.

PREMIER PROBLÈME

Le but du problème est d'étudier un algorithme de tri.

Un arbre enraciné est un arbre dont un sommet, la racine, est distingué des autres. Intuitivement, deux arbres enracinés sont égaux modulo un ordre sur les fils. On dessine habituellement les arbres enracinés avec la racine vers le haut.

Un arbre planaire est un arbre enraciné dont l'ensemble des fils de chaque sommet est ordonné. On dessine habituellement sur un même niveau tous les fils d'un sommet en allant de gauche à droite lorsqu'on décrit les fils du premier jusqu'au dernier. Ainsi, les deux arbres de la figure 1 sont différents si on les considère comme des arbres planaires mais ils sont égaux si on les considère comme des arbres enracinés.

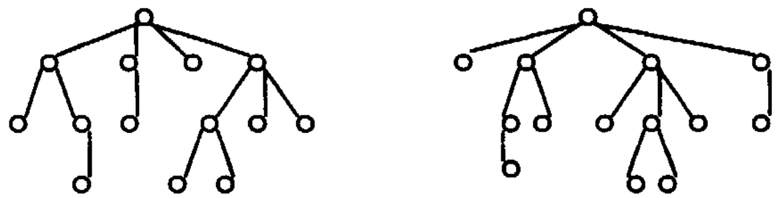
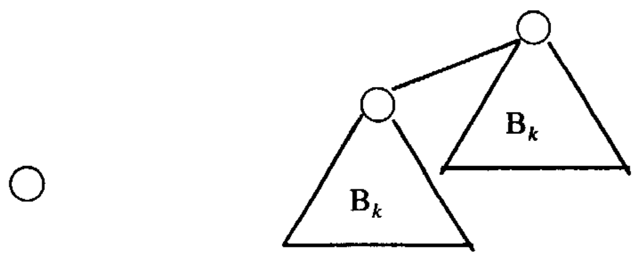


Figure 1

L'arbre binomial B_k ($k \geqslant 0$) est un arbre planaire défini récursivement de la façon suivante :

1. L'arbre binomial B_0 est formé d'un unique sommet, sa racine.
2. L'arbre binomial B_{k+1} est formé en rajoutant à la racine d'un premier arbre binomial B_k un fils à gauche qui est lui-même la racine d'un deuxième arbre binomial B_k ; la racine du tout étant la racine du premier arbre binomial B_k (cf. fig. 2).



B_0 B_{k+1}

Figure 2

Tournez la page S.V.P.

On rappelle que :

- **la taille d'un arbre planaire est le nombre de sommets de cet arbre ;**
- **la hauteur d'un arbre planaire est la distance maximale, distance comptée en nombre d'arêtes, entre la racine et une feuille de l'arbre ;**
- **le degré d'un sommet est le nombre de ses fils.**

1. Propriétés des arbres binomiaux.

1.1. Dessinez les arbres binomiaux B_k pour $k = 0, 1, \ldots, 5$.

1.2. Quelle est la taille de l'arbre binomial B_k ?

1.3. Quelle est la hauteur de l'arbre binomial B_k ?

1.4. Combien de sommets sont à une distance i de la racine dans l'arbre binomial B_k (distance comptée en nombre d'arêtes) ?

1.5. Quel est le degré de la racine de l'arbre binomial B_k ?

1.6. Que peut-on dire du sous-arbre planaire qui a pour racine le i-ième fils de la racine de l'arbre binomial B_k, les fils étant numérotés en ordre croissant de la gauche vers la droite, le premier ayant le numéro 0 ?

1.7. Quel est le degré maximum d'un sommet dans un arbre binomial de taille n ?

2. Les suites binomiales.

On note $\mathbb{N}$ l'ensemble des nombres entiers positifs ou nuls et $\mathbb{R}$ l'ensemble des nombres réels.

Une *suite binomiale* S_n est une suite d'arbres binomiaux telle que :

1° la suite des tailles des arbres qui la composent est strictement croissante,

2° la somme des tailles des arbres qui la composent est égale à n.

Pour les questions suivantes, on considère que la taille d'une suite binomiale S_n est n.

En fait, une suite binomiale S_n est une structure de données qui contient les éléments d'un ensemble totalement ordonné de taille n (tous les éléments sont différents). Pour ce problème, nous allons supposer qu'il s'agit d'un ensemble de nombres entiers. Chaque sommet de la suite binomiale S_n contient un des entiers de l'ensemble (*i.e.* est étiqueté par un entier de l'ensemble).

Les racines des arbres constituant S_n sont chaînées entre elles. Les arbres de S_n sont représentés en machine en utilisant la représentation fils aîné-frère droit. Plus précisément chaque sommet est une cellule comportant :

- un pointeur vers son premier fils (le plus à gauche) ; ce pointeur sera NIL si le sommet est une feuille ;
- un pointeur vers son frère droit (le sommet ayant le même père et qui est immédiatement à sa droite) ; ce pointeur sera NIL, si le sommet est le fils le plus à droite de son propre père ;
- un pointeur qui, pour une racine, pointe vers la racine suivante de la liste ; ce pointeur sera à NIL si le sommet n'est pas une racine ou s'il est la dernière racine de la suite. (*Remarque :* on peut se passer de ce pointeur en utilisant le pointeur frère droit pour les racines) ;
- un champ donnant le degré du sommet ;
- un champ contenant l'entier associé à ce sommet.

Une suite binomiale sera désignée par un pointeur sur la racine du premier arbre binomial de la suite.

De plus on respectera à tout instant la règle suivante : l'entier associé à un sommet est strictement supérieur à l'entier associé à son père.

On définit la complexité d'un algorithme A sur une donnée d de taille n (notée $c_A(d)$) comme le nombre de comparaisons (*i.e.* instructions « if » portant sur les éléments de l'ensemble contenu dans la suite binomiale) effectuées par l'algorithme A lors de son exécution à partir de d. La complexité de l'algorithme A est alors la fonction $C_A(n)$ définie par :

$$C_A(n) = \max\{\, c_A(d);\ d \text{ de taille } n \,\}.$$

Nous appellerons ***ordre de grandeur asymptotique*** **d'une complexité $C_A(n)$ une fonction [illegible] que $\exists\, k_1 \in \mathbf{R},\ \exists\, k_2 \in \mathbf{R},\ \exists\, n_0 \in \mathbf{N}$ tel que $\forall n \geq n_0,\ k_1 \cdot f(n) \leq C_A(n) \leq k_2 \cdot f(n)$ [illegible]**

Les algorithmes seront écrits dans le langage Pascal ou le langage Fortran. Dans ce dernier cas, les pointeurs seront des indices dans des tableaux.

2.1. Expliquez comment, à chaque entier n, on peut faire correspondre une et une seule suite binomiale S_n.

2.2. Donnez l'algorithme de la procédure sb_minimum(*s*,*x*) qui renvoie dans *x* un pointeur [illegible] contenant le plus petit entier associé dans la suite binomiale S_n pointée par *s*.

2.3. Calculez un ordre de grandeur asymptotique de la complexité de l'algorithme que vous avez construit pour sb_minimum(*s*,*x*).

2.4. Donnez l'algorithme de la procédure sb_union(*s*,*s1*,*s2*) qui renvoie dans *s* un pointeur sur la suite qui est l'union des deux suites pointées par *s1* et *s2*.

2.5. Calculez un ordre de grandeur asymptotique de la complexité de l'algorithme que vous avez construit pour sb_union(*s*,*s1*,*s2*), *n* étant la taille de *s* (la somme des tailles de *s1* et *s2*).

2.6. En utilisant les procédures précédentes, donnez l'algorithme de la procédure sf_tri(*n*,*v1*,*v2*) qui renvoie dans un vecteur *v2* les éléments triés par ordre croissant d'un vecteur *v1* de taille *n*.

2.7. Donnez un ordre de grandeur asymptotique de la complexité de cet algorithme de tri.

DEUXIÈME PROBLÈME

Le but de ce problème, constitué de trois parties, est d'étudier les propriétés de base des monoïdes et des mots. Toutes les définitions utiles sont données, en général, dans chaque partie ; la troisième partie fait exception car elle utilise des définitions sur les mots finis, déjà données dans la deuxième partie. Il y a de plus une certaine logique dans l'ordre des parties, ce qui fait qu'on conseille de lire le texte en entier avant de commencer. Les questions marquées d'une étoile demandent un peu plus de réflexion.

1. Monoïdes et monoïdes libres.

Un *monoïde* est un ensemble M muni d'une loi de composition interne, notée multiplicativement, qui est associative et possède un élément neutre noté 1_M (ou 1, s'il n'y a pas de confusion). Un *morphisme* d'un monoïde M dans un monoïde M′ est une application f de M dans M′ telle que :

$$f(m \cdot n) = f(m) \cdot f(n) \qquad \text{pour tous } m, n \in \mathrm{M} \quad \text{et} \quad f(1_{\mathrm{M}}) = 1_{\mathrm{M}'}.$$

On définit le *produit* de deux parties A et B de M par $\mathrm{A} \cdot \mathrm{B} = \{x \in \mathrm{M} / x = a \cdot b \text{ avec } a \in \mathrm{A} \text{ et } b \in \mathrm{B}\}$ et la *différence* par $\mathrm{A} - \mathrm{B} = \{x \in \mathrm{A} / x \notin \mathrm{B}\}$. Pour une partie A de M, on notera A^* l'ensemble des produits (finis) d'éléments de A auquel on ajoute l'élément neutre 1 ; plus formellement, on définit A^* par :

$$\mathrm{A}^* = \{x \in \mathrm{M} / \text{il existe } n,\ n \geq 1, \text{ éléments } a_i \in \mathrm{A} \text{ tel que } x = a_1 \ldots a_n\} \cup \{1\}.$$

Une partie A de M est un *ensemble générateur* de M si $\mathrm{M} = \mathrm{A}^*$. Un monoïde M est *libre* s'il existe une partie A de M telle que $\mathrm{M} = \mathrm{A}^*$ et si la décomposition d'un élément $m \in \mathrm{M}$ en éléments de A est unique ; A est alors appelé une *base* de M. Une partie N d'un monoïde M est un *sous-monoïde* de M si le produit de deux éléments de N est toujours un éléments de N ($\mathrm{N} \subseteq \mathrm{N} \cdot \mathrm{N}$) et si $1_{\mathrm{M}} \in \mathrm{N}$.

1.1. Montrez que l'élément neutre d'un monoïde est unique.

*1.2. Donnez un exemple d'une partie N d'un monoïde M telle que N soit un monoïde sans être un sous-monoïde de M. On suppose que la loi de N est la même que celle de M.

1.3. Montrez que l'intersection de sous-monoïdes est un sous-monoïde.

1.4. Soit $f : \mathrm{M} \longrightarrow \mathrm{N}$ un morphisme du monoïde M dans le monoïde N. Montrez que $f(\mathrm{M})$ est un sous-monoïde de N.

1.5. Soit G un groupe, N un monoïde, $f : \mathrm{G} \longrightarrow \mathrm{N}$ un morphisme de monoïde. Montrez que $f(\mathrm{G})$ est un sous-groupe de N.

Tournez la page S.V.P.

1.6. Montrez qu'un morphisme de monoïde $f : A^* \rightarrow N$ est entièrement déterminé par sa donnée sur A. Réciproquement, montrez que si $f : A \rightarrow N$ est une application alors elle s'étend de façon unique en un morphisme $f' : A^* \rightarrow N$.

1.7. Montrez que si M est un sous-monoïde de A^* alors M admet un unique ensemble de générateurs minimal pour l'inclusion.

Indication : considérez $X = (M - \{1_M\}) - (M - \{1_M\})^2$.

1.8. Montrez que si M est un sous-monoïde libre de A^* alors son unique base est X, avec $X = (M - \{1_M\}) - (M - \{1_M\})^2$.

1.9. En gardant les notations précédentes, montrez que X^* n'est pas toujours un monoïde libre.

Indication : considérez $X = (\{ab, ba, a\})^*$.

1.10. Montrez que la base d'un monoïde libre est unique.

2. Mots.

Soit A un ensemble fini et non vide appelé *alphabet* dont les éléments sont appelés des *lettres*. On appelle *mot* sur un alphabet A une suite finie d'éléments de A :

$$(a_1, a_2, \ldots, a_n), \; a_i \in A.$$

La *concaténation* est une opération binaire définie par :

$$(a_1, a_2, \ldots, a_n) \cdot (b_1, b_2, \ldots, b_p) = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_p).$$

La concaténation est associative et permet d'écrire un mot sous la forme suivante :

$$a_1 a_2 \ldots a_n$$

au lieu de $(a_1) \cdot (a_2) \ldots (a_n)$ en identifiant une lettre $a \in A$ avec la suite (a).

La suite vide, appelée *mot vide*, est l'élément neutre pour la concaténation et est notée 1. On a donc, pour tout mot w :

$$1 \cdot w = w \cdot 1 = w.$$

L'ensemble des mots sur A est notée A^*. On notera A^+ l'ensemble des mots non vides sur A : $A^+ = A^* - \{1\}$. On notera x^n le mot $x \ldots x$ ($n \geqslant 0$ fois) et on l'appellera x à la puissance n ; par définition, $x^0 = 1$. La *longueur* d'un mot $w = a_1 a_2 \ldots a_n$ est le nombre n de lettres composant w et est notée $|w|$; la longueur du mot vide est 0 : $|1| = 0$. On a alors $|u \cdot v| = |u| + |v|$.

Un mot $f \in A^*$ est un *facteur* d'un mot $w \in A^*$ s'il existe deux mots $x, y \in A^*$ tels que $w = xfy$; f est un *facteur gauche* de w si $w = fy$ (on écrira $f \leqslant w$) ; f est un *facteur gauche propre* de w si $f \leqslant w$, $f \neq 1$ et $f \neq w$.

Deux mots u, v de A^+ sont *conjugués* s'il existe deux mots x, y dans A^* tels que $u = xy$ et $v = yx$.

2.1. Soit A un alphabet. Montrez que A^* est un monoïde libre.

2.2. Montrez que la relation « être facteur gauche » entre mots de A^* est une relation d'ordre.

*2.3. Montrez que si x, y, z, t sont des mots tels que $xy = zt$ alors il existe un mot u tel que :

ou bien $(xu = z$ et $y = ut)$;

ou bien $(x = zu$ et $uy = t)$.

2.4. Déduire du résultat précédent que deux facteurs gauches f, g d'un mot w sont toujours comparables pour l'ordre « être facteur gauche ».

*2.5. Montrez que, si u et v sont deux mots de A^+, alors les trois conditions suivantes sont équivalentes :

(1) $uv = vu$;

(2) u et v sont puissances d'un même mot ;

(3) il existe $n, p > 0$ tels que $u^n = v^p$.

2.6. **Trouvez une condition nécessaire et suffisante sur les mots u, v et w pour qu'on ait $u^2v^2 = w^2$**

2.7. **Trouvez une condition nécessaire et suffisante sur les mots u, v et w pour qu'on ait u [illegible]**

2.8. **Généralisez ces deux derniers résultats.**

2.9. **Montrez que la relation de conjugaison est une relation d'équivalence.**

***2.10.** **Montrez que si u, v, w sont trois mots de A^+, alors $uv = vw$ si et seulement s'il existe deux mots x, y et un entier $k \geqslant 0$ tels que : $u = xy$, $v = (xy)^k x$, $w = yx$.**

2.11. **En déduire que deux mots u et w sont conjugués si et seulement s'il existe un mot v tel que $uw = vw$.**

3. Mots infinis.

Un *mot infini* w sur l'alphabet A est une suite infinie d'éléments de A :

$$w = (a_1, a_2, \ldots, a_n, \ldots),\ a_i \in A$$

que l'on notera :

$$w = a_1 a_2 \ldots a_n \ldots$$

Le *facteur gauche* de longueur n d'un mot infini w est le mot $w[n] = a_1 a_2 \ldots a_n$. On peut définir un mot infini comme la limite d'une suite infinie strictement croissante de mots (pour l'ordre « être facteur gauche »). En effet, soit $w_1, w_2, \ldots, w_n, \ldots$ une suite infinie strictement croissante de mots dans A^* telle que chaque w_n est un facteur gauche propre de w_{n+1}. On définit alors le mot infini w par :

$$\text{quel que soit } n,\ n \geqslant 1, \quad w[|w_n|] = w_n.$$

Ce mot infini est aussi noté :

$$w = \lim w_n.$$

On précisera, à partir de maintenant, lorsqu'on parlera de mot s'il s'agit de mot fini ou de mot infini.

On dit qu'un mot fini u non vide a *deux occurrences* dans le mot w s'il existe quatre mots finis x, y, x', y' dans A^* tels que :

$$w = xuy = x'uy' \qquad \text{avec } x \neq x'.$$

Supposons par exemple $|x| < |x'|$. On dira que les deux occurrences de u *se chevauchent* lorsque $|x'| < |xu|$; le mot u sera appelé un *facteur chevauchant.*

Un *carré* est un mot fini de la forme uu où u est un mot fini non vide. Un mot fini ou infini *contient un carré* si un de ses facteurs est un carré ; sinon le mot est dit *sans carré.*

3.1. Montrez qu'un mot fini w contient deux occurrences, qui se chevauchent, d'un mot $u \neq 1$ si et seulement si w contient un facteur de la forme $avava$ où a est une lettre et v un mot.

Dans la suite du problème, on pose $A = \{a,b\}$ et $f : A^* \longrightarrow A^*$ le morphisme défini par :

$$f(a) = ab \qquad \text{et} \qquad f(b) = ba.$$

3.2. Montrez qu'il y a exactement six mots sans carré dans A^+ et énumérez-les.

3.3. Montrez que $f^{n+1}(a) = f^n(a) f^n(b)$ et que $f^n(a)$ est un facteur gauche propre de $f^{n+1}(a)$. La limite de la suite $f^n(a)$ existe donc. On pose $t = \lim f^n(a)$.

3.4. Montrez que le mot infini t contient des carrés.

On va montrer que t ne contient pas de facteur chevauchant.

3.5. Soit $X = \{ab, ba\}$. Montrez que si $x \in X^*$ alors $axa \notin X^*$ et $bxb \notin X^*$.

*3.6. Montrez que si $w \in A^+$ n'a pas de facteur chevauchant alors $f(w)$ non plus.

3.7. Montrez que le mot infini t ne contient pas de facteur chevauchant.

*3.8. Construire un mot infini sans carré sur l'alphabet $A = \{a, b, c\}$.

*3.9. Montrez que l'existence d'une infinité de mots sans carré est équivalente à l'existence d'un mot infini sans carré (rappelons que A est fini).