Examen Final LO21 – A 09
Durée 2 heures
Document non autorisé

Exercice 1

On considère la liste de tous les arrêts de bus d'un réseau de transport en commun.
Une ligne de bus est une liste d'arrêt. Chaque arrêt est caractérisé par son nom et la distance qui le sépare de l'arrêt précédent de la ligne.

1) Étant donnée une ligne de bus, écrire l'algorithme récursif du sous programme permettant de calculer la distance totale de toute cette ligne.
2) On considère l'ensemble du réseau de bus comme une liste de lignes de bus. Chaque élément de cette liste est caractérisé par un numéro de la ligne et une liste d'arrêts.
    a) Écrire l'algorithme du sous programme récursif qui donne le numéro de la ligne la plus courte (la ligne ayant la plus petite distance totale), sans utiliser de structure temporaire pour stocker la longueur des lignes.
    b) Donner en langage C la déclaration complète du type réseau de bus.

Exercice 2

Considérons le vocabulaire V = {+,=,a} et la grammaire G= ({S, M}, {+,=,a}, →, S) définie sur V avec les règles suivantes:

1. S → aSa
2. S → a + Ma
3. M → aMa
4. M → a = a

1) Monter en appliquant les règles de la grammaire (par dérivation) que le mot aa + aaaa = aaaaaa (qu'on peut noter $a^2 + a^4 = a^6$) appartient au langage L(G) (Pour chaque étape de la dérivation, veuillez préciser le numéro de la règle utilisée).
2) Donner la forme générale des mots appartenant à L(G) (langage défini par la grammaire G).

Exercice 3: Conjecture de Syracuse

Considérons la fonction de Collatz f : N → N définie comme suit:

- Si n est pair, f(n)= n/2.
- Si n est impair, f(n)= 3*n + 1.

Étant donné un entier positif C et la suite Un définie comme suit :

- Pour n=0, $U_0 = C$
- Pour n> 0, $U_n = f(U_{n-1})$

La conjecture affirme que, pour tout C > 0, il existe un indice n tel que la suite atteigne 1.
Exemples :
En partant de *C* = 6, la suite est la suivante: 6, 3, 10, 5, 16, 8, 4, 2, 1.
En partant de *C* = 11, la suite est la suivante: 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.
1) Écrire l'algorithme du sous-programme nommé *Collatz* qui calcule la valeur de la fonction f pour un entier donné.
2) Étant donné un entier positif C, écrire l'algorithme récursif du sous-programme nommé *Syracuse* qui construit la liste des valeurs successives de la suite U_n ($u_0, u_1, \ldots, 1$), jusqu'à ce que la suite atteigne la valeur 1 pour la première fois.
Par exemple, pour C=6, *Syracuse* construit la liste 6, 3, 10, 5, 16, 8, 4, 2, 1.

3) Écrire l'algorithme récursif d'un sous-programme nommé *stoppingTime* qui calcule le nombre minimal d'itérations nécessaires à la fonction de Collatz pour atteindre 1 pour la première fois. Par xemple, pour C=6, *stoppingTime* renvoie la valeur 9.

Exercice 4

On souhaite représenter les résultats d'un tournois de tennis à l'aide d'un arbre binaire. Chaque nœud de l'arbre contient le nom du vainqueur (chaîne de caractères), l'identifiant ou numéro du match (un entier) entre les joueurs se trouvant dans ces nœuds fils. Pour les nœuds feuilles, le nom du vainqueur correspond au nom d'un joueur et l'identifiant du match est –1.

Pour comparer des chaînes de caractères, vous pouvez utiliser (sans la définir) l'opération EgaleCh (ch1,ch2) qui donne vraie si les deux chaînes ch1 et ch2 sont égales, faux sinon.

1) Écrire l'algorithme récursif du sous programme construisant une liste de tous les identifiants des matchs gagnés par le vainqueur du tournois. La liste retournée doit être triée par ordre des matchs joués.