République Algérienne démocratique et populaire
Centre universitaire de Mila
Institut des Sciences et Technologies

Domaine : MI – 2 ème année informatique
Année : 2023-2024
Module: Algorithmique et structure de données 3

Corrigé contrôle

Exercice 1 : (**5.5 points**)

1. Ecrire une procédure ***supprimer_ième*** (var F : File, i : entier) : **(3 points)**

```
Procédure supprimer_ième (var F : File, i : entier)
P1: Pile
Début
    Si i ≤ Taille (P) alors 0.25
0.25  Initialiser (F1) ;
        Tantque ! est_vide (F)  faire 0,25
      0,5  Si i≠1 alors
                Emfiler (F1, tête (F )) ; 0,5
             Finsi
             Défiler (F) ; 0,25
             i ← i-1 ; 0,25
        Fin Tantque
        F←F1 ; 0,5
    Sinon
        Ecrire ("erreur") ; 0,25
    Finsi
Fin
```

```
Procédure supprimer_ième (var F : File, i : entier)  (2.5 points)
P1: Pile
Début
    Si i ≤ Taille (F) alors
0,25
        J← Taille (F) ; 0.5
        Tantque j>0  faire 0.5
             Si i≠1 alors
                Enfiler (F, tête (F )) ; 0.5
             Finsi
             Defiler (F) ; 0,25
             i ← i-1 ;
             J← j-1 ; 0,25
        Fin Tantque
    Sinon
        Ecrire ("erreur") ; 0.25
    Finsi
Fin
```

Exercice 2 : (8 points)

1. **Fonction est_valide1 (S : chaine de caractère)** (3 points)

```
Fonction est_valide1 (S : chaine de caractère)

i: entier ; P : Pile ;

Début
i ←0 ; P← Nil;
Tantque S[i] ≠ '\0' faire                                   0.25
    Si S[i] = '(' alors                                     0.25
        Empiler (P, '('); 0.5
    Sinon
        Si S[i] = ')' alors 0.25
          Si ! est_vide (p) alors
              Dépiler (P); 0.5
          Sinon
              Retourne Faux; 0.5
          Fin si
        Finsi
    Finsi
    i← i+1;
Fin Tantque
Si est_vide (P) alors
    Retourne vrai ; 0.5
Sinon
    Retourne faux ; 0.25
Fisi
Fin
```

1. **Fonction est_valide2 (L : liste) (2.5 points)**

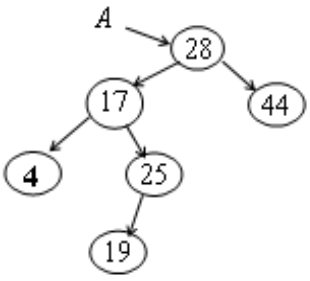
```
Fonction est_valide (L : liste) : booléen
Courant : Liste ; P : Pile ;
Début
Courant ←L ; P← Nil;
Tantque courant ≠ Nil faire
    Si courant ->ele = '(' ou courant ->ele = '{' alors 0.5
        Empiler (P, courant ->ele );
    Sinon
      Si (courant ->ele = ')' ou courant ->ele = '}') alors 0.5
        Si est_vide (p) alors
            Retourne faux ; 0.5
        Sinon                                                                    0.5
            Si (courant ->ele = ')' et sommet (p) = '(' ) ou ( courant ->ele = '}' et sommet (p) = '{') ) alors
                dépiler (P) ;
            sinon                                                 0.25
                Retourne Faux;                                    0.25
            Fin si
    Finsi
    Courant ← courant -> suivant ;
Fin Tantque
Si est_vide (P) alors
    Retourne vrai ;
Sinon
    Retourne faux ;
Finsi
Fin
```

2. Fonction supprimer (L : Liste) : Liste (2)

```
Fonction supprimer (L : Liste) : Liste
Début
Si L= Nil alors                                          0.75
        Retourne Nil ;
Sinon
        Si premier (L) ='{' ou premier (L) = '}' alors   0.5
                Retourne supprimer (reste (L)) ;         0.5
        Sion
                L-> suivant ← supprimer (reste (L)) ;    0.75
                Retourne L ;
        Finsi
Finsi
```

Exercice 3: (9.25 points)

1. Quelle est le nombre maximal de nœuds parcourus de la recherche d'un élément :
   - Dans un arbre binaire simple: La taille de l'arbre **0.75**
   - Dans un arbre binaire de recherche : 1 + Hauteur de l'arbre **0.75**
2. La complexité de l'opération de recherche d'un élément dans un arbre binaire de recherche équilibré est O (log (N)). **0.75**



3. Un exemple d'arbres binaires de recherche non équilibrés. **0. 75**

   La différence entre la hauteur de sous arbre gauche de nœud 28 et la hauteur de sous arbre droite de nœud 28 est supérieur à 1. **0.75**

4. Fonction nb_nœudsparcourus1: **(3.25 points)**

```
Fonction nb_nœudsparcours1 (A : arbre, x : entier) : entier
Début
Si a=Nil alors
        Return 0 ;                                       0.75
Sinon
        Si x= contenu (A) alors
                Return 1 ;                               0.75
        Sinon
                Si x<contenu (a) alors 0.25
                        Retourne 1 + nb_nœudsparcourus1 (filsG(A), x) ;   0.75
                Sinon
                        Retourne 1+ nb_nœudsparcourus1 (filsD(a), x) ;    0.75
                Finsi
    Finsi
Finsi
Fin
```

5. Fonction ***nb_nœudsparcourus2(*** **2.25 points)**

Fonction ***nb_nœudsparcourus*** (A : arbre, x : entier, var b : booléen) : entier **0.25**

Début

Si a=Nil alors

Return 0 ; **0.25**

Sinon

Si x= contenu (A) alors

B←vrai ; **0.5**

Return 1 ; **0.25**

Sinon

B←Faux ; **0.25**

Nb ←1+ ***nb_nœudsparcours*** (filsG(A), x, B) ; **0.25**

Si B = Faux alors **0.25**

Nb←Nb+ ***nb_nœudsparcours*** (filsD(a), x) ; **0.25**

Finsi

Finsi

Finsi

Fin