

Rattrapage d'algorithmique 1

08h-09h30

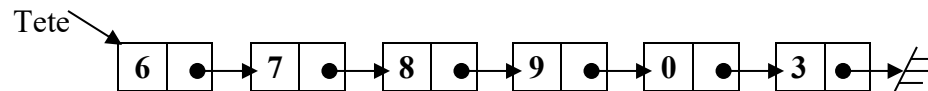
A1, A2

Exercice 1 (7 pts: 1 + 2 + 2 + 2)

On souhaite représenter les nombres entiers par des listes linéaires chaînées de caractères où chaque chiffre d'un nombre est rangé dans un maillon de la liste sous forme d'un caractère:

Exemple :

Le nombre "678903" est représentée par la liste suivante :



Il est demandé d'écrire en langage algorithmique vu dans le cours les fonctions suivantes :

1. Fonction **Pair** (L :Pointeur(TMaillon)) : Booleen
Qui retourne Vrai si le nombre représenté par L est pair, et Faux sinon.
2. Fonction **Mult100**(L: Pointeur(TMaillon)) : Pointeur(TMaillon) ;
Qui retourne une nouvelle liste représentant le nombre représenté par L multiplié par 100.
3. Fonction **Div100**(L:Pointeur(TMaillon)):Pointeur(TMaillon);
Qui retourne une nouvelle liste représentant le nombre représenté par L divisé par 100.
4. Fonction **Mod100**(L: Pointeur(TMaillon)):Pointeur(TMaillon);
Qui retourne une nouvelle liste représentant le reste de division du nombre représenté par L par 100.

On donne les fonctions:

CarVersEnt(C :caractère) : donne la valeur entière du caractère C

CarVersEnt('8') = 8

EntVersCar(I :Entier) : donne le caractère correspondant au chiffre I

EntVersCar(8) = '8'

Exercice 2 (8 pts : 1.5 + 2 + 2 + 2.5)

Soit les valeurs des clés triés selon l'ordre d'arrivée :

14, 23, 4, 9, 17, 11, 28, 16, 3, 7

- Dessiner l'arbre binaire de recherche obtenu en insérant ces clés dans l'ordre d'arrivée

- Dessiner l'arbre après la suppression des clés : 11, 17, 23, 14

Ecrire les fonctions suivantes :

1. Fonction **NbrePairs** (Racine: Pointeur(TNoeud)):entier ;
Qui retourne le nombre de clés paires dans l'arbre.
2. Fonction **SommeInterne** (Racine: Pointeur(TNoeud)):entier ;
Qui retourne la somme des clés non feuilles dans l'arbre.

Exercice 3 (5 pts : 3 + 1 + 1)

- Dessiner l'arbre représentant le tas construit après l'insertion dans l'ordre des clés suivants : 14, 23, 4, 9, 17, 11, 28, 16, 3, 7

- Donner le tas statique correspondant

- Donner le tas (arbre) après un retrait

Bon courage

Fortigé type

Exercice 1 (7 pts: 1 + 2 + 2 + 2)

1. Fonction **Pair** (L :Pointeur(TMaillon)) : Booleen
Var P : Pointeur(TMaillon) ;
Debut
P ← L ;
Si P=Nil alors Pair ← Vrai
Sinon
TQ Suivant(P) ≠ Nil Faire
P ← Suivant(P) ;
FTQ ;
Si CarVersEnt(Valeur(P)) mod 2 = 0 alors Pair ← Vrai
Sinon
Pair ← Faux ;
FSi
FSi
Fin ;
2. Fonction **Mult100**(L: Pointeur(TMaillon)) : Pointeur(TMaillon) ;
Var P,P1,Q,Tete : Pointeur(TMaillon) ;
Debut
P ← L ; Tete ← Nil ;
TQ P ≠ Nil Faire
Allouer (P1) ; Aff_Val(P1, Valeur(P)) ; Aff_Adr(P1,Nil) ;
Si Tete=Nil alors Tete ← P1
Sinon Aff_Adr(Q,P1)
FSi ;
Q ← P1 ; P ← Suivant(P) ;
FTQ ;
Si Tete ≠ Nil alors
Allouer(P) ; Aff_Val(P,'0') ; Aff_Adr(Q,P) ; Q ← P ;
Allouer(P) ; Aff_Val(P,'0') ; Aff_Adr(Q,P) ; Aff_Adr(P, Nil) ;
FSi
Mult100 ← Tete ;
Fin ;

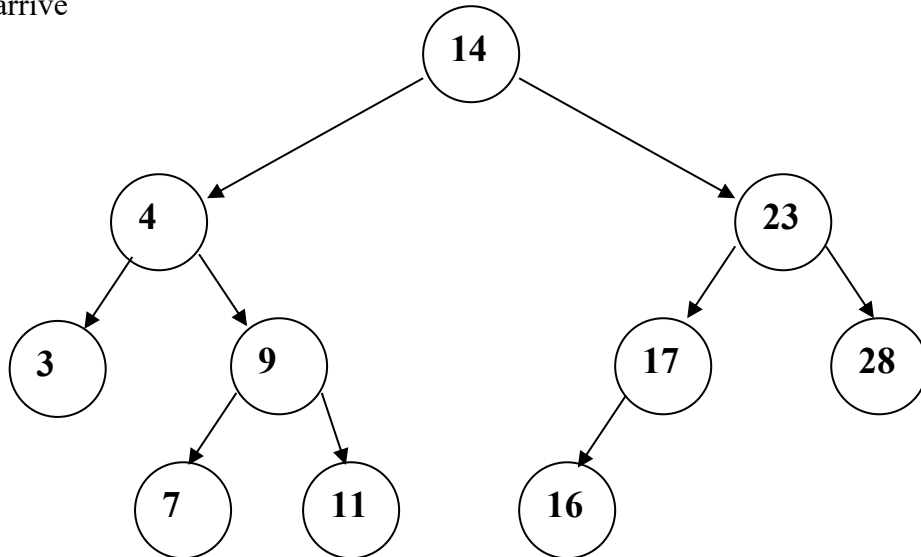
3. Fonction **Div100**(L:Pointeur(TMaillon)):Pointeur(TMaillon);
Var P,P1, Q,Tete : Pointeur(TMaillon) ;
Debut
P ← L ; Tete ← Nil ;
TQ (P ≠ Nil et Suivant(P) ≠ Nil et Suiavnt(Suivant(P)) ≠ Nil) Faire
Allouer (P1) ; Aff_Val(P1, Valeur(P)) ; Aff_Adr(P1,Nil) ;
Si Tete=Nil alors Tete ← P1
Sinon Aff_Adr(Q,P1)
FSi ;
Q ← P1 ; P ← Suivant(P) ;
FTQ ;
Div100 ← Tete ;
Fin ;
4. Fonction **Mod100**(L: Pointeur(TMaillon)):Pointeur(TMaillon);
Var P,P1, Q,Tete : Pointeur(TMaillon) ;
Debut
P ← L ; Tete ← Nil ;
TQ (P ≠ Nil et Suivant(P) ≠ Nil et Suiavnt(Suivant(P)) ≠ Nil) Faire
P ← Suivant(P) ;
FTQ ;
Tete ← Nil ;
TQ P ≠ Nil Faire
Si Val(P) ≠ '0' ou Q ≠ Nil alors
Allouer (P1) ; Aff_Val(P1, Valeur(P)) ; Aff_Adr(P1,Nil) ;
Si Tete=Nil alors Tete ← P1
Sinon Aff_Adr(Q,P1)
FSi ;
Q ← P1 ;
FSi
P ← Suivant(P) ;
FTQ ;
Mod100 ← Tete ;
Fin ;

Exercice 2 (8 pts : 1.5 + 2 + 2 + 2.5)

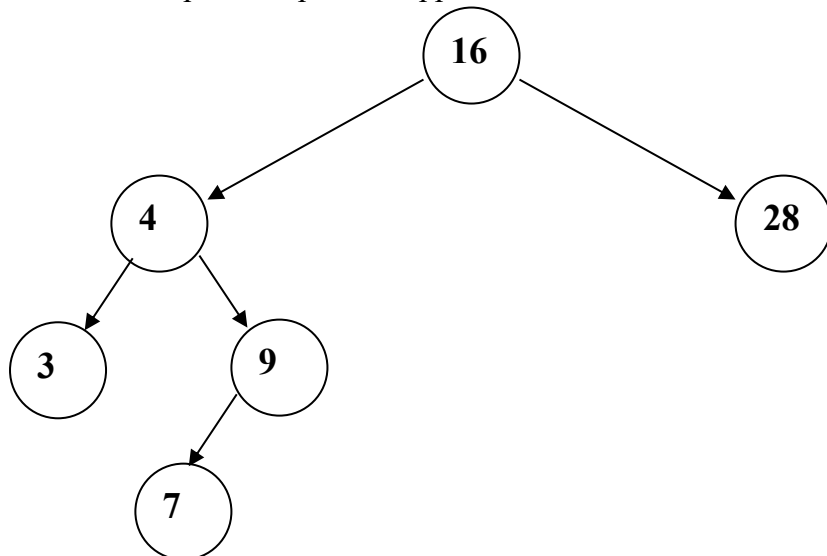
Soit les valeurs des clés triées selon l'ordre d'arrivée :

14, 23, 4, 9, 17, 11, 28, 16, 3, 7

- Dessiner l'arbre binaire de recherche obtenu en insérant ces clés dans l'ordre d'arrivée



- Dessiner l'arbre après la suppression des clés : 11, 17, 23, 14



Ecrire les fonctions suivantes :

1. Fonction **NbrePairs** (Racine: Pointeur(TNoeud)):entier ;

Debut

Si Racine = Nil alors NbrePairs \leftarrow 0

Sinon

NbrePairs \leftarrow ((Clé(Racine) + 1) mod 2) + NbrePairs(FG(Racine)) +
NbrePairs(FD(Racine)) ;

FSi ;

Fin ;

2. Fonction **SommeInterne** (Racine: Pointeur(TNoeud)):entier ;

Debut

Si Racine = Nil ou (FG(Racine)=Nil et FD(Racine)=Nil) alors
SommeInterne \leftarrow 0

Sinon

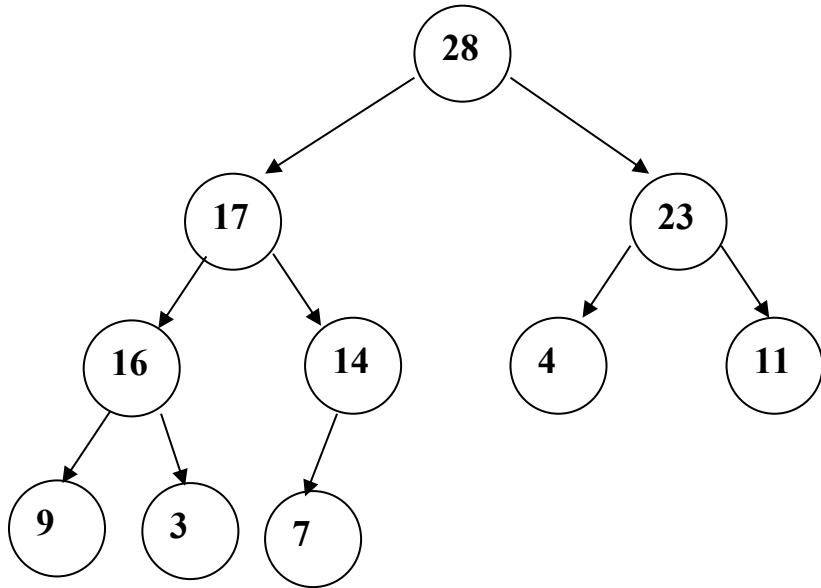
SommeInterne \leftarrow (Clé(Racine) + SommeInterne(FG(Racine)) +
SommeInterne(FD(Racine)) ;

FSi ;

Fin ;

Exercice 3 (5 pts : 3 + 1 + 1)

- Dessiner l'arbre représentant le tas construit après l'insertion dans l'ordre des clés suivants : 14, 23, 4, 9, 17, 11, 28, 16, 3, 7



- Donner le tas statique correspondant

28	17	23	16	14	4	11	9	3	7
----	----	----	----	----	---	----	---	---	---

- Donner le tas (arbre) après un retrait

