

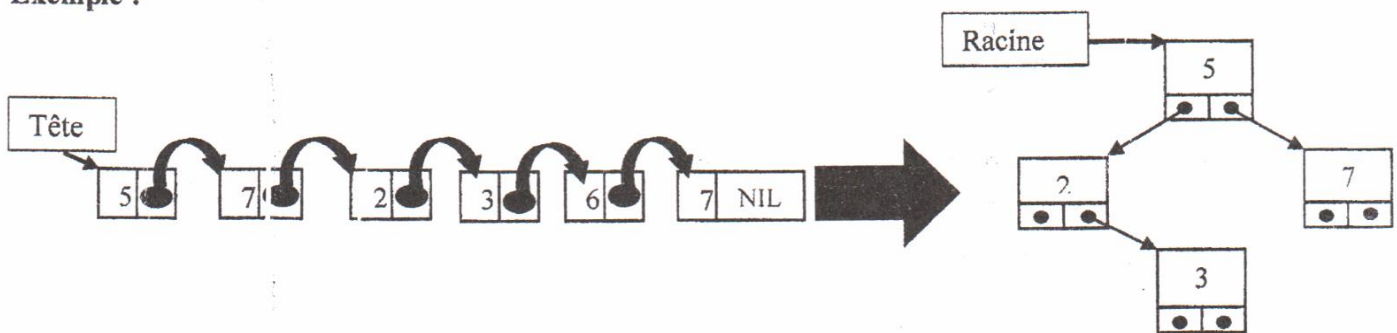
Examen de rattrapage d'Algorithmique et Structures de données avancées

2^{ème} année licence
(01 h 30 min)

« La propreté de la feuille de réponses est exigée et notée (1 point) »

L'objectif de cet examen consiste à manipuler les nombres premiers d'une liste de valeurs entières. Autrement dit, nous voulons extraire les nombres premiers d'une liste linéaire chaînée, tout en ignorant les valeurs répétées.

Exemple :



Pour ce faire, nous allons diviser le travail en quatre parties :

Partie 1 : (Remplir la liste) (2 points)

1. Ecrire, en langage algorithmique, la procédure qui permet de créer une liste à partir des valeurs entières fournies par l'utilisateur. (2 pts)

Procédure CréerListe (Var Tête : Pointeur (Maillon), N : entier)

Partie 2 : (Création d'un arbre binaire de recherche (ABR)) (8 points)

Dans cette partie, nous allons créer un arbre binaire de recherche à partir d'une liste linéaire chaînée.

1. Ecrire, en langage algorithmique, une fonction booléenne qui vérifie si N est un nombre premier ou non. (3 pts)

Fonction Premier (N : Entier) : Booléen

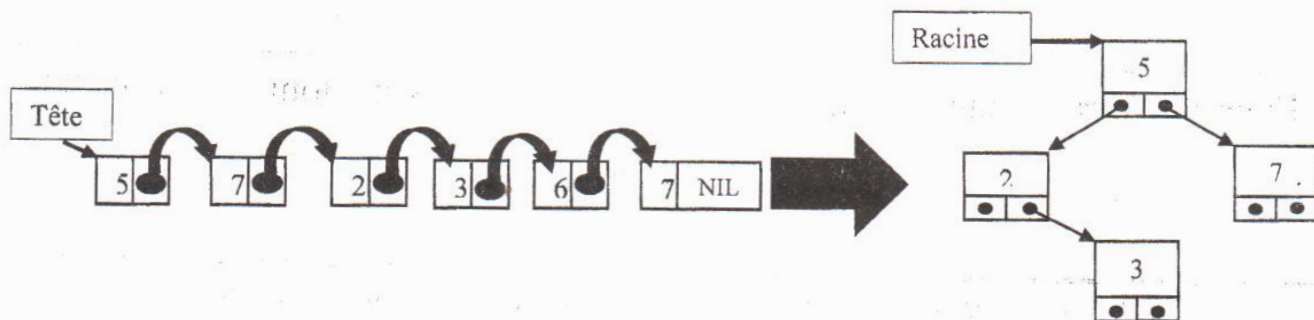
2. Ecrire, en langage algorithmique, la procédure d'insertion d'une valeur dans un arbre binaire de recherche (2 pts).

Procédure InsérerABR (Var Racine : Pointeur (Nœud), Val : entier)

3. Ecrire, en langage algorithmique, la procédure qui permet de créer un arbre binaire de recherche à partir de nombres premiers d'une liste linéaire chaînée (3pts).

Procédure CréerABR-Premier (Var Racine:Pointeur (Nœud), Tête : Pointeur (Maillon))

Exemple # :



$\text{MinArbrePremier}(\text{Racine})=2$
 $\text{TailleArbrePremier}(\text{Racine})=3$
 $\text{FeuilleArbrePremier}(\text{Racine})=2$

Partie 3 : (Manipulation de l'ABR de nombres premiers) (6 points)

1. Ecrire, en langage algorithmique, une fonction qui permet de déterminer le minimum des valeurs de l'arbre précédent (ABR des nombres premiers) (2 points). (Exemple #)

Fonction *MinArbrePremier* (*Racine:Pointeur(Noeud)*) : Entier

2. Ecrire, en langage algorithmique, une fonction récursive qui permet de déterminer le nombre des nœuds de l'ABR précédent (ABR des nombres premiers). Le nombre de nœuds est appelé aussi la taille de l'arbre (2 points). (Exemple #)

Fonction *TailleArbrePremier* (*Racine:Pointeur(Noeud)*) : Entier

3. Ecrire, en langage algorithmique, une fonction récursive qui permet de déterminer le nombre de feuilles de l'ABR précédent (ABR des nombres premiers). Une feuille est un nœud sans fils (FG=Nil et FD=Nil) (2 points). (Exemple #)

Fonction *FeuilleArbrePremier* (*Racine:Pointeur(Noeud)*) : Entier

Partie 4 : (Programme principal) (3 points)

En utilisant les modules des parties précédentes, écrire le programme principal (Sans la partie déclaration de variables, fonctions ...etc.) qui permet de :

- 1) Créer une liste Linéaire chaînée à partir des valeurs fournies par l'utilisateur.
- 2) A partir des nombres premiers de cette liste, créer un arbre binaire de recherche (ABR).
- 3) Afficher le nombre des nombres premiers.
- 4) Afficher le plus petit des nombres premiers (minimum).

Bonne chance