USTL - Licence ST-A 2ème année 2006-2007

Algorithmes et Programmation Impérative 2

Examen de janvier 2007

durée 2h - documents de cours autorisés - calculatrices non autorisées.

Exercice 1.

```
function f(n : CARDINAL) : CARDINAL;
var
  s,i : CARDINAL;
begin
  if n=0 then
    f := 1
  else begin
    s := 0;
    for i := 0 to n-1 do
      s := s + f(i)*f(n-1-i);
    f := s;
  end {if};
end {f};
```

Question 1. Donnez les valeurs calculées par cette fonction pour n compris entre 0 et 4.
Question 2. Écrivez la relation de récurrence qui lie la valeur de $f(n+1)$ aux valeurs $f(k)$ avec $k \leq n$.
Question 3.
Q 1.3–1. Montrez que le nombre $c(n)$ de multiplications d'entiers effectuées pour le calcul de $f(n)$ vérifie les équations

$$\begin{aligned} c(0) &= 0 \\ \forall n \geq 1 \quad c(n) &= n + 2\sum_{i=0}^{n-1} c(i). \end{aligned}$$

Q 1.3–2. En déduire que pour tout entier n on a

$$c(n+1) = 3c(n) + 1.$$

Q 1.3–3. En déduire que pour tout entier $c(n) = \frac{3^n-1}{2}$.
Question 4. En utilisant un tableau pour mémoriser les valeurs de $f(n)$ proposez un algorithme quadratique pour calculer $f(n)$.

Exercice 2. *Purger une pile*

On veut dans cet exercice, réaliser une procédure qui purge une pile (d'entiers) passée en paramètre, de toutes les occurrences d'un entier lui aussi passé en paramètre. Dans l'exemple illustré à la figure 1, on peut voir une pile avant et après la purge de toutes les occurrences des 1.

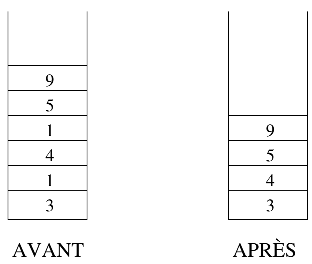

FIG. 1 – Une pile avant et après purge de tous les 1

Question 1. Quelle structure de données pensez-vous utiliser pour accomplir cette tâche ? Expliquez votre choix.
Question 2. Réalisez la procédure `purger`.

Exercice 3. *Permutation circulaire d'une liste*

On appelle *permutation circulaire* d'une liste l la liste l' obtenue en plaçant le dernier élément de l en tête.

l	l'
()	()
(1)	(1)
(1,2)	(2,1)
(1,2,3)	(3,1,2)
(1,2,3,4)	(4,1,2,3)

TAB. 1 – Exemples de listes et leur permutations circulaires

Dans cet exercice, vous allez proposez deux versions différentes de l'obtention d'une permutation circulaire d'une liste.

Question 1. *1ère version* La liste l' est une liste entièrement constituée de nouvelles cellules (cf figure 2). En particulier, la liste d'origine l est inchangée.

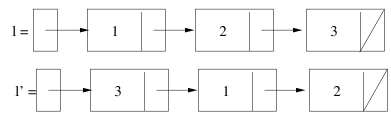

FIG. 2 – Permutation circulaire :1ère version

Faîtes une fonction `permute` qui réalise cette version.

Question 2. *2ème version* Aucune nouvelle cellule n'est créée, et la liste l est modifiée.

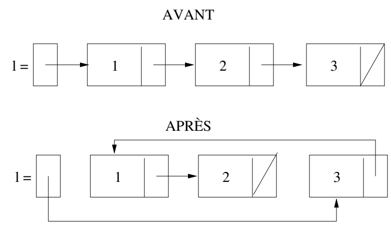

FIG. 3 – Permutation circulaire :2ème version

Faîtes une procédure `permuter` qui réalise cette version.

Exercice 4. *Construction dichotomique d'arbres*

À partir d'un tableau indéxé de 1 à N, il est possible de construire un arbre binaire de taille n dont

- la racine est l'élément du milieu du tableau ;
- le sous-arbre gauche est obtenu en appliquant la même construction sur la partie du tableau qui précède l'élément du milieu ;
- et le sous-arbre droit est obtenu à partir de la partie qui suit l'élément du milieu.

La figure 4 montre un tableau et l'arbre binaire ainsi construit.

On supposera définis les types et fonction qui suivent :

```
const
  N = ...; // le nombre d'éléments d'un tableau
type
  T_ELEMENT = ...;
  T_INDICE  = 1..N;
  T_INDICE_ETENDU = 0..N+1;
  TABLEAU = array[T_INDICE] of T_ELEMENT;

// milieu(a,b) = entier au milieu de
// l'intervalle ⟦a,b⟧
function milieu(a,b : T_INDICE_ETENDU) : T_INDICE_ETENDU;
begin
  milieu := (a+b) div 2;
end {milieu};
```

Question 1. Réalisez une fonction `tableauEnArbre` qui transforme un tableau en un arbre binaire selon cette méthode dichotomique.

Question 2. Un arbre binaire est dit *équilibré en hauteur* si

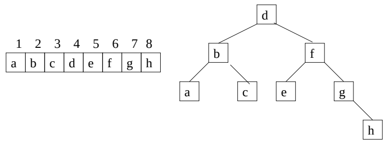

FIG. 4 – Un tableau et l'arbre associé

- il est vide ;
- ou bien si ses deux sous-arbres sont équilibrés en hauteur et la différence de leur hauteur n'excède pas 1.

Les arbres produits par `tableauEnArbre` sont-ils équilibrés en hauteur ?

Question 3. Un arbre binaire est dit *équilibré en taille* si

- il est vide ;
- ou bien si ses deux sous-arbres sont équilibrés en taille et la différence de leur taille n'excède pas 1.

Les arbres produits par `tableauEnArbre` sont-ils équilibrés en taille ?

Exercice 5. *Parking*

Achille possède un parking en centre ville capable d'accueillir un nombre M de voitures, chaque place étant numérotée de 0 à $M-1$. Mais n'aimant pas les automates, et soucieux d'accueillir le mieux possible les automobilistes qui viennent chez lui garer leur voiture, il décide d'attribuer personnellement à chaque arrivée un numéro de place en fonction de certaines caractéristiques de la voiture à garer.

Question 1. Achille hésite sur les propriétés de la voiture à prendre en considération :

- sa couleur ;
- sa marque ;
- ou son immatriculation.

Aidez-le à choisir la propriété la mieux adaptée pour répartir au mieux les voitures dans son parking.

Question 2. À partir de la propriété caractéristique choisie pour attribuer un numéro de place, donnez une méthode (programmable) de calcul de ce numéro.

Question 3. Il peut arriver que la méthode de calcul du numéro de place attribue le même numéro à deux voitures différentes. Il est alors possible qu'un automobiliste trouve occupée la place qui lui a été attribuée. Quelle consigne Achille peut-il donner à cet automobiliste pour trouver une place ?