Interrogation TD
Sujet B

10/04/2007, Licence 1 MASS semestre 2

Question 1

Ecrire l'algorithme de recherche dichotomique dans un tableau d'entiers de taille n ordonné par ordre dÃĭcroissant.

Algorithme rechercheDichotomique(x : entier, t : tableau d'entier, n : entier) : booleen
début
variable a,b,c : entier
  $a \leftarrow 0$
  $b \leftarrow n-1$
  $c \leftarrow (a+b)/2$
  tant que $a \leq b$ et $t[c] \neq x$ **faire**
    si $t[c] < x$ **alors**
      $a \leftarrow c+1$
    sinon
      $b \leftarrow c-1$
    fin si
    $c \leftarrow (a+b)/2$
  fin tant que
  retourner $a \leq b$
fin

Barème :

- entete correcte : 0.5 point
- initalisation : 0.5 point
- boucle correcte : 1 point
- test correct : 2 points

Question 2

Dans cet exercice, seul les tableaux d'entiers non ordonnés de taille n sont considérés.

a - Ecrire un algorithme qui calcule le produit des nombres contenus dans un tableau d'entiers entre les indices i en $i-5$.

Algorithme produit5(i : entier, t : tableau d'entier, n : entier) : entier
début
variable s, j : entier
  $s \leftarrow 1$
  pour j **de** 0 **à** 5 **faire**
    $s \leftarrow s * t[i-j]$
  fin pour
  retourner s

fin

Barème :

- entete correcte : 1 point
- initalisation : 0.5 point
- boucle correcte : 1 point
- accumulation correcte : 0.5 point

b - Ecrire un algorithme qui recherche le plus petit produit de 5 nombres consécutifs dans le tableau. Le résultat de l'algorithme sera ce plus petit produit.

Algorithme petitProduit5(t : tableau d'entier, n : entier) : entier
début
variable min, s, i : entier
  $min \leftarrow$ produit5(5, t, n)
  pour i **de** 6 **à** $n-1$ **faire**
    $s \leftarrow$ produit5(i, t, n)
    si $s < min$ **alors**
      min $\leftarrow$ s
    fin si
  fin pour
  retourner min
fin

Barème :

- entete correcte : 1 point
- initalisation : 1 point
- boucle correcte : 1 point
- test correct : 1 point

Question 3

a - Ecrire un algorithme récursif *construction* qui prend en paramètre un entier positif n et qui retourne une liste de taille n qui commence par 0 si n est pair et qui commence par 1 si n est impair.
Par exemple, construction(5) retourne la liste $(1, 0, 1, 0, 1)$ et construction(6) retourne la liste $(0, 1, 0, 1, 0, 1)$

Algorithme construction(n : entier) : liste
début
  si $n = 0$ **alors**
    retourner listeVide()
  sinon
    si $modulo(n, 2) = 0$ **alors**
      retourner listeCons(0, construction($n-1$))
    sinon
      retourner listeCons(1, construction($n-1$))
    fin si
  fin si
fin

Barème :

- entete correcte : 1 point
- test de base correct : 1 point
- appel récursif : 1 point
- utilisation des primitives correctes : 1 point

b - Ecrire un algorithme qui retourne *Vrai* si et seulement si la liste donnée en paramètre ne contient que des 1.

```
Algorithme  uniforme1(l : liste) : booleen
début
  si listeEstVide?(l) alors
    retourner Vrai
  sinon
    retourner (listeTete(l)=1) et uniforme1(listeQueue(l))
  fin si
fin
```

Barème :

- entete correcte : 0.5 point
- test de base correct : 0.5 point
- appel récursif : 1 point
- utilisation des primitives correctes : 1 point

c - Donner un exemple d'exécution des algorithmes précédents.

Bareme :
1 point pour chaque exemple correct