

G3, Contrôle Continu d'Éléments d'Algorithmique n° 2

Exercice 1. *Multiplication recursive terminale* [8pts].

La fonction (itérative) suivante multiplie deux entiers non-négatifs a, b :

```

fonction MULTITER( $a, b$ )
   $r \leftarrow 0$ 
  tant que  $b > 0$  faire
    si  $b \bmod 2 = 1$  alors
       $r \leftarrow r + a$ 
     $a \leftarrow 2 * a$ 
     $b \leftarrow b/2$ 
  retourne  $r$ 

```

(a) Exécutez à la main la fonction multIter pour les valeurs suivantes :

- i. $a = 4, b = 12$;
- ii. $a = 7, b = 8$;
- iii. $a = 8, b = 7$;
- iv. $a = 5, b = 0$.

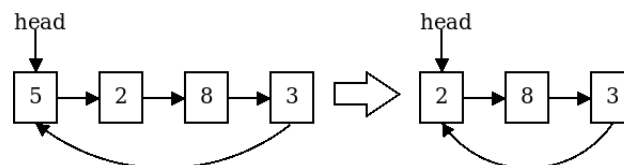
Montrez les valeurs intermédiaires des variables a, b, r pour chaque répétition de la boucle.

(b) Écrivez la version récursive terminale de la fonction multIter.

Exercice 2. *Listes chaînées circulaires* [4pts].

Une liste chaînée circulaire est une liste (simplement) chaînée dont l'élément qui suit le dernier est la tête (si la liste contient un seul élément, cet élément se suit lui-même).

Écrivez une fonction qui, étant donné une liste chaînée circulaire non-vide l , enlève la tête $l.head$ de la liste. La fonction renvoie alors une liste chaînée circulaire dont la tête est le deuxième élément de l (si cela existe) et le dernier élément connecte à la nouvelle tête (figure ci-dessous). Attention aux cas pathologiques !



Exercice 3. *Tableau circulement trié* [8pts].

Un tableau *circulement trié* est un tableau de n éléments distincts initialement triés par ordre croissant, puis décalés de façon cyclique vers la droite de p positions ($0 < p < n$). Ainsi, le minimum se trouve dans la position p du tableau, le maximum dans la position $p - 1$ et le premier élément est supérieur au dernier.

Par exemple, les tableaux $\{5, 6, 1, 2, 3\}$ et $\{2, 4, 7, 8, 1\}$ sont des tableaux circulement triés. Par contre, les tableaux $\{3, 4, 5, 1, 6\}$, $\{1, 3, 5, 6\}$ et $\{4, 3, 1, 8, 7\}$ n'en sont pas.

- (a) Écrivez un algorithme qui calcule le décalage p dans un tableau circulement trié. L'efficacité de votre algorithme sera prise en compte dans l'évaluation.
- (b) Quelle est la complexité de votre algorithme au pire des cas ? Expliquez.

Annexe

Algorithm 1 Recherche Dichotomique

Entrée : tableau T trié contenant des entiers, élément x

```
1:  $n \leftarrow T.\text{longueur}$ 
2: fonction RECHDICH( $T, x$ )
3:   retourne RECHDICHREC( $T, x, 0, n - 1$ )
4: fonction RECHDICHREC( $T, x, l, r$ )
5:   si  $r < l$  alors retourne non trouvé
6:    $mid \leftarrow (l + r)/2$ 
7:   si  $T[mid] = x$  alors retourne trouvé
8:   si  $T[mid] > x$  alors retourne RECHDICHREC( $T, x, l, mid - 1$ )
9:   sinon retourne RECHDICHREC( $T, x, mid + 1, r$ )
```

Algorithm 2 Listes chaînées, les structures

```
1: Structure Cellule :
2:   entier data
3:   Cellule next
4:
5: Structure Liste :
6:   Cellule head
```
