

Contrôle continu du jeudi 18 octobre 2021 (Correction)

Durée : 1 heure

Aucun document. Aucune machine. Barème indicatif. Tous les exercices sont indépendants.

Exercice 1 : (8 points)

On considère un tableau contenant n cartes, où chaque carte est faite de deux informations, une couleur qui est un entier compris entre 1 et 4 et une valeur qui est un entier entre 1 et 13. Par exemple : $T = [[4,1],[1,10],[3,1],[4,13],[1,9],[1,8],[2,3]]$

- Que fait l'algorithme suivant lorsqu'on l'appelle avec un tableau de cartes? Executer l'algorithme sur le tableau T donné un exemple, en écrivant le tableau résultat et tous les tableaux intermédiaires pour i de 1 à 6 dans la table ci dessous.

```
public static void triCartesT(int [][]T){
    int n=T.length;
    int [] tmp;
    for(int i=1; i< n; i++){
        int j=i;
        while (j>0 && T[j][1]<T[j-1][1]){
            tmp=T[j-1];
            T[j-1]=T[j];
            T[j]=tmp;
            j--;
        }
    }
}
```

Itérations	Tableau
i=1	
i=2	
i=3	
i=4	
i=5	
i=6	

▷ on obtient : $T = [[4,1],[3,1],[2,3],[1,8],[1,9],[1,10],[4,13]]$.

- Supposons d'avoir un tableau dont les couleurs sont ordonnés dans l'ordre croissant, par exemple $W = [[1,9],[2,9],[3,8],[3,2],[4,9],[4,7],[4,8]]$. Est-ce que, en appliquant l'algorithme précédent, le tableau obtenu est tel que à valeur égales, les couleurs sont ordonnés dans l'ordre croissant? Expliquez pourquoi.
- On voudrait maintenant adapter le tri précédent afin de trier les cartes selon leur couleur et, à couleur fixée, selon la valeur. Sur l'exemple précédent on doit obtenir $[[1,8],[1,9],[1,10],[2,3],[3,1],[4,1],[4,13]]$. Pour adapter le tri précédent nous allons juste remplacer le test $T[j][1]<T[j-1][1]$ dans la boucle **while** par un appel `mineur(T[j],T[j-1])` à une fonction `boolean mineur(int []tab, int []tab2)`. Écrivez le corps de la fonction `mineur(T[j],T[j-1])` dans l'espace suivant :

▷ **Réponse :**

```

public static boolean mineur(int []tab, int []tab2){
    if(tab[0]<tab2[0])
        return true;
    else if (tab[0]>tab2[0])
        return false;
    else if (tab[1]<=tab2[1])
        return true;
    else return false;
}

public static void triCartesTBis(int [][]T){
    int n=T.length;
    int [] tmp;
    for(int i=1; i< n; i++){
        int j=i;
        while (j>0 && mineur(T[j],T[j-1])){
            tmp=T[j-1];
            T[j-1]=T[j];
            T[j]=tmp;
            j--;
        }
    }
}

```

Exercice 2 : (6 Points)

Soit T un tableau d'entiers triés, écrire un algorithme dichotomique récursif terminal `boolean` `DichoRecT(int [] T, int j, ...)` (les arguments de `DichoRec` sont à compléter par vous) qui permet de vérifier si un entier j est présent dans un tableau T de taille n et tel que, à chaque appel récursif, imprime le nombre d'appel récursif jusqu'à cet appel. La fonction renverra `true` ou `false` selon si l'entier j est présent, ou pas, dans le tableau T . Ecrivez l'instruction qui permettrait de tester le fonctionnement de `DichoRecT`.

▷ *Réponse :*

```

public static boolean DichoRecT(int []tab, int j, int l, int r, int depth){
    print("profondeur de recursion: "+depth);
    if r == l then return false;
    int i = tab[(l+r)/2]
    if i == j then return true;
    if i > j then return DichoRecT(tab, j, l, (l+r)/2, depth+1);
    return DichoRecT(tab, j, (l+r)/2, r, depth+1);
}

public static boolean cherche(int []tab, int j){
    int n=tab.length;
    return DichoRecT(tab, j, 0, n, 0);
}

```

Exercice 3 : (6 points)

Dans cet exercice, on utilise les classes `Liste` et `Cellule` pour les listes chaînées.

```
class Liste {  
    Cellule head;  
}
```

```
class Cellule {  
    int key;  
    Cellule next;  
}
```

1. Quelle liste est stockée dans L après la suite d'instructions suivantes ? (Faire un dessin.)

```
1  a := new Cellule(1, nil)  
2  b := new Cellule(2, a)  
3  c := new Cellule(3, nil);  
4  d := new Cellule(4, c);  
5  
6  a.next := d;  
7  b.key := 5;  
8  L := new List(b);
```

▷ **Réponse :** 5→1→4→3

2. Ecrire une fonction `Liste copie(Liste l)` qui renvoie une copie de la liste l à l'aide d'une fonction auxiliaire récursive.

▷ **Réponse :**

```
Cellule copie_aux(Cellule cell){  
    if cell == nil return nil;  
    return = new Cellule(cell->key, copie_aux(cell->next));  
}
```

```
Liste copie(Liste l)  
    new liste L;  
    L->head=copie_aux(l->head);  
    return L;  
}
```