

# Examen final

## Cours d'analyse, algorithmique

—Master 2 CCI—

---

Les documents manuscrits, sujets de travaux pratiques et dirigés ainsi que les supports de cours sont autorisés. Tous les autres documents tels que livres, calculatrices, téléphones portables et ordinateurs sont interdits.

Les exercices sont indépendants. Si l'on ne sait pas justifier une question, on peut néanmoins utiliser la réponse dans la suite.

**Durée : 3 heures**

---

► **Exercice 1. (Intersection d'ensembles codés par tableaux triés)**

Dans cet exercice, on code les ensembles de nombres à l'aide de tableaux sans répétitions. Dans un premier temps, on ne suppose pas que les tableaux sont triés. Par exemple l'ensemble  $A := \{1, 4, 6, 7\}$  pourra être encodé par le tableau `tabA`  $:= [4, 6, 1, 7]$  de taille `tailleA`  $:= 4$ .

1. Étant donné, un tableau `tabA` de taille `tailleA` encodant l'ensemble  $A$  et un élément  $e$ , écrire un algorithme qui répond si  $e$  appartient à l'ensemble  $A$ .
2. En déduire un algorithme qui prend en paramètre deux tableaux
  - `tabA` de taille `tailleA` encodant un ensemble  $A$
  - `tabB` de taille `tailleB` encodant un ensemble  $B$et qui remplit un tableau `tabC` de taille `tailleC` pour coder l'intersection  $C := A \cap B$  de  $A$  et  $B$ . Note : On pourra supposer que `tabC` est alloué avec une taille suffisante, on ne demande pas d'écrire cette allocation.
3. Montrer que la complexité de l'algorithme obtenu est dans le pire des cas

$$O(\text{tailleA} \times \text{tailleB}).$$

Dans cette deuxième partie, on suppose que les tableaux `tabA` et `tabB` sont triés.

4. En s'inspirant de l'algorithme de fusion de deux tableaux, écrire un algorithme qui remplit le tableau `tabC` de taille `tailleC` pour coder l'intersection  $C := A \cap B$  de  $A$  et  $B$ .
5. Montrer que la complexité de l'algorithme est dans le pire des cas

$$O(\text{tailleA} + \text{tailleB}).$$

► **Exercice 2. (Recherche dans un tableau)**

On dispose d'un tableau **trié**  $T$  de nombres réels (`double` en C) de taille  $n$ . Soit  $x$  un nombre réel. On veut trouver, parmi les nombres du tableau, lequel approche le mieux  $x$ . Par exemple, pour le tableau

0	1	2	3	4	5	6	7	8
0.2	1.5	2.7	4.2	5.8	7.7	7.9	11.0	12.6

Si l'on cherche 1.6, la réponse est 1.5.

1. Dans cette question, on suppose de plus donné un intervalle d'indices du tableaux  $[\text{min}, \text{max}]$  **de taille supérieure à 2** (c'est-à-dire  $\text{max} \geq \text{min} + 1$ ) et tel que

$$T[\text{min}] \leq x \leq T[\text{max}].$$

En utilisant une recherche dichotomique, écrire un algorithme qui retourne le meilleur approximant  $y$  de  $x$  dans l'intervalle  $[\text{min}, \text{max}]$ . Indication : le cas de base est obtenu quand  $\text{max} = \text{min} + 1$ .

2. Répondre au problème posé en traitant à part les cas où  $x \leq T[0]$  et  $x \geq T[n-1]$ .
3. Exécuter l'algorithme sur le tableau en exemple pour  $x = 5.9$ ,  $x = 2.3$  et  $x = 0$ .
4. Quelle est la complexité?

► **Exercice 3. (Programme inconnu)**

On considère le programme suivant :

```
1  int cat(int n, rest)
2  {
3      int res = 0, i;
4      if (rest[n] != 0) return rest[n];
5      for (i=0; i<n; i++)
6          res = res + cat(i, rest)*cat(n-1-i, rest);
7      rest[n] = res;
8      return res;
9  }
```

On initialise un tableau `rest` de 5 entiers de la manière suivante :

$i$	0	1	2	3	4
<code>rest[i]</code>	1	0	0	0	0

On appelle `cat(4, rest)`. Donner la trace d'exécution de l'algorithme ainsi que la valeur retournée et l'état du tableau `rest` à la fin de l'appel? Indication : il y a 21 appels récurifs.

► **Exercice 4. (Dessin de triangle)**

Écrire un programme qui, pour un entier positif  $h$  donné, affiche le triangle équilatéral «pointe en haut» remplis avec des étoiles, la base ayant pour longueur  $2 * h + 1$ , et la hauteur  $h$ . Une case sur deux contient une étoile :

Un exemple : lorsque  $h$  vaut 5, l’affichage est

```
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * *
```

L’affichage doit bien entendu se faire ligne par ligne. On pourra utiliser les commandes `Affiche("*")` et `Affiche("_")` pour afficher une étoile ou un espace et `NouvelleLigne` pour aller à la ligne.