

Examen du 18-19 juin 2020

A disposition le 18 juin à 8h. A rendre pour le 19 juin à 10h

Préliminaires : *Ce sujet est constitué de 6 exercices qui peuvent être traités dans l'ordre de votre choix. Le barème est donné à titre indicatif. Il est conseillé de lire l'intégralité du sujet avant de commencer. Vous devez déposer votre "copie" sur le dépôt de moodle, vous pouvez déposer un texte électronique ou une photographie de votre copie à votre convenance. En cas de problème contacter cd@irif.fr*

Exercice 1 : (4 points)

Exécutez à la main le tri par insertion vu en cours sur le tableau suivant :

45	7	20	1	60
----	---	----	---	----

Exercice 2 : (4 points)

T est un tableau d'entiers de longueur n.

```
1 WhoAmI(T) :  
2   for i from n-1 downto 1  
3     for j from 0 to i-1  
4       if T[j+1] < T[j]  
5         exchange T[j+1] and T[j]
```

1. Exécuter l'algorithme `whoAmI` sur le tableau $T = \{10, 2, 8, 7, 4, 3, 5\}$. Que fait `WhoAmI(T)` ? Combien de comparaisons sont effectuées ?

Exercice 3 : (4 points)

1. En utilisant un algorithme dichotomique, trouver l'indice de la première occurrence d'un nombre j dans le tableau T . Dans le cas où j n'est pas présent dans T l'algorithme doit renvoyer -1 . Quelle est sa complexité ?
2. En vous inspirant de l'algorithme de la question précédente, écrire un algorithme qui renvoie le nombre d'occurrences de j dans T . Quelle est sa complexité ?
3. Comparer la complexité de l'algorithme de la question précédente avec l'algorithme qui consiste à parcourir tout le tableau T en utilisant un accumulateur. Lequel est le plus efficace ?

Exercice 4 : (4 points)

Étant donné un tableau de n éléments, nous voulons écrire deux algorithmes qui permettent de vérifier que les n éléments soient tous distincts.

1. Le premier algorithme doit utiliser un tri : écrivez le pseudo-code et une description courte de cet algorithme.
2. Pour le deuxième algorithme nous supposons que les valeurs de tous les éléments soit comprises entre 0 et $n - 1$ et nous utilisons un tableau auxiliaire de taille n : écrivez le pseudo-code et une description courte de cet algorithme.

Exercice 5 : (4 points)

Écrire un algorithme récursif qui calcule le carré d'un entier n , sans utiliser l'opération de multiplication ou de puissance ni les boucles `for` ou `while`. Écrivez le pseudo-code et une description courte de cet algorithme.

Exercice 6 : (4 points)

Le *coefficient binomial* peut être défini comme une fonction qui associe à deux entiers m et k , tels que $k \leq m$, le nombre de parties de m à k éléments. Elle s'écrit $\binom{m}{k}$, et se lit « k parmi m ».

1. Écrivez un algorithme récursif qui prend en entrée deux entiers m et k , et calcule $\binom{m}{k}$, en vous aidant du fait que $\binom{m}{m} = \binom{m}{0} = 1$, et en utilisant la formule de Pascal :

$$\binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$$

La seule opération arithmétique autorisée est l'addition¹.

2. Votre algorithme est-il récursif terminal ? Justifiez.
3. Montrez que votre algorithme termine toujours.

1. k parmi m est aussi égal à $\frac{m!}{k!(m-k)!}$, mais cela ne vous sera pas utile pour cet exercice.