

Algorithmique procédurale

Examen - ING1 GI

7 janvier 2019

Modalités

- Durée : 2 heures
- Vous devez rédiger votre copie à l'aide d'un **stylo à encre** exclusivement.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document n'est autorisé.
- Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.
- Aucun déplacement n'est autorisé.
- **Aucune question au professeur n'est autorisé.** Si vous pensez avoir détecté une erreur, continuez en expliquant les hypothèses que vous faites.
- Aucun échange, de quelque nature que ce soit, n'est possible.
- Le barème est donné à titre indicatif.

Rappel :

La **procédure erreur(msg : Chaîne de caractère)** permet de quitter un programme en affichant le message **msg** à l'utilisateur.

Exercice 1 : Zéro d'une fonction (5 pts)

Le principe de la dichotomie pour déterminer un zéro (une racine) d'une fonction est le suivant : on considère deux nombres réels a et b et une fonction réelle f continue sur l'intervalle $[a, b]$ telle que $f(a)$ et $f(b)$ soient de signes opposés. D'après le théorème des valeurs intermédiaires, f a au moins un zéro dans l'intervalle $[a, b]$. La méthode de dichotomie consiste à diviser l'intervalle en deux en calculant $m = (a + b)/2$. Il y a maintenant deux possibilités : soit $f(a)$ et $f(m)$ sont de signes contraires, soit $f(m)$ et $f(b)$ sont de signes contraires. On rappelle le même principe sur l'intervalle restreint $[a, m]$ ou $[m, b]$ jusqu'à trouver un zéro de la fonction. Numériquement, il est fort possible qu'on ne trouve pas exactement 0, donc on acceptera une solution à un ε près.

On suppose que la fonction $f(x : \text{réel}) : \text{réel}$ est définie.

Version 1 : itératif

1. Ecrire la méthode itérative `zeroFonction` qui, pour un intervalle $[a, b]$ donné, renvoie un zéro de la fonction à un ε près.

2. Donner l'invariant de boucle de la fonction `zeroFonction`.
3. Démontrer l'invariant donné dans la question précédente.

Version 2 : récursif

4. Ecrire la méthode récursive `zeroFonctionRec` qui, pour un intervalle $[a, b]$ donné, renvoie un zéro de la fonction à un ε près.
5. Donner la formule de récurrence du calcul de complexité de la fonction `zeroFonctionRec`.
6. Résoudre la formule précédente et donner la complexité de la fonction `zeroFonctionRec`.

Exercice 2 : Les lutins du Père Noël (5 pts)

Le Père Noël a fini de faire sa visite habituelle chez les humains et retourne au Pôle Nord vers ses lutins afin de confectionner les différents cadeaux pour les enfants. Il a enregistré toutes les informations dans le fichier "Noel2018.txt". Pour chaque enfant qu'il a visité, il a noté son numéro de cheminée (unique), son prénom, son âge et le cadeau qu'il a demandé. Voici un extrait de son fichier :

```
E415F62 Thomas 5 Camion de pompier en LEGO
E82G34D Charlotte 7 Barbie fait du ski
K12FG890 Kim 3 Peluche panda
```

Le lutin en chef qui reçoit le fichier veut écrire un programme afin de traiter la production de tous ces jouets. Ne connaissant pas le nombre d'enfants qu'a visité le Père Noël, il veut stocker toutes les données dans une table de hachage. Il a déjà réfléchi à une fonction de hachage qui permet de convertir le numéro de cheminée en entier compris entre 1 et 42314, le nombre de lutin en service. Le prototype de cette fonction est

```
h(numeroCheminee : chaine) : entier.
```

On suppose que la fonction `convertir (nombre : chaine) : entier`, qui permet de convertir en entier une chaîne de caractère contenant un nombre, est déjà définie.

1. Définir l'enregistrement qui permet de stocker les informations du fichier dans une table de hachage.
2. Ecrire la méthode `extraireDonnees`, qui permet de lire l'ensemble des informations du fichier et de les stocker dans la table de hachage qui sera retournée.
3. Afin qu'en usine ils puissent faire plusieurs jouets en même temps, ils veulent connaître le nombre de jouets du même type à faire. Ecrire la méthode `nombreDeJouets` qui, pour un nom de jouet donné, renvoie le nombre de fois que ce nom apparaît.
4. Dès qu'un jouet est terminé, emballé et étiqueté pour un enfant, un lutin le supprime de la table de hachage. Ecrire la méthode `supprimerJouetDe` qui, pour un enfant

identifié par son numéro de cheminée, enlève l'information associée dans la table de hachage.

5. Le lutin en chef veut connaître, à chaque fin de journée, le nombre de cadeaux restant à gérer. Ecrire la méthode `nombreCadeauxRestant`.

Exercice 3 : Pivot dans le tri rapide (1,5 pts)

On exécute l'algorithme du tri rapide sur un tableau. On suppose qu'après l'application de la procédure partition, on obtient le tableau $\{3,1,2,4,5,8,7,6,9\}$. Quels sont tous les éléments qui ont pu servir de pivot à cette étape ? Justifier.

Exercice 4 : Complexité des tris sur certaines instances particulières (2,5 pts)

Donner la liste complète des algorithmes de tri par comparaison de paires d'éléments étudiés dans ce module. Pour chacun d'entre eux, quelle est la complexité de l'algorithme :

- si on prend en entrée un tableau déjà triée ?
- si on prend en entrée un tableau déjà triée dans l'ordre inverse ?
- si on prend en entrée un tableau dont tous les éléments sont identiques ?

Justifier précisément toutes vos réponses.

Exercice 5 : Calcul des inversions d'un tableau (6 pts)

On appelle *inversion* deux indices i et j d'un tableau tab tels que $i < j$ et $tab[i] > tab[j]$. On suppose que tous les éléments du tableau sont distincts.

- 1) Quel est le nombre d'inversions dans le tableau $\{1,3,5,2,4,6\}$?
- 2) Quel est le nombre minimal d'inversions dans un tableau de taille n ? Justifier.
- 3) Quel est le nombre maximal d'inversions dans un tableau de taille n ? Justifier.
- 4) Donner un algorithme naïf pour calculer le nombre d'inversions. Quelle est sa complexité ?
- 5) Comment modifier le tri par insertion pour calculer le nombre d'inversions ? Quelle est la complexité de ce deuxième algorithme ?
- 6) Donner une équation de récurrence pour le nombre d'inversions en utilisant le principe "diviser pour régner". (Indication : cette équation est composée de 3 termes.) Comment modifier le tri fusion pour calculer le nombre d'inversions ? Quelle est la complexité de ce troisième algorithme ?