Nom :
Prénom :
Groupe de TD :

Partiel LU2IN003

Mercredi 29 Mars 2023, 1.5 heures
Aucun document autorisé

Exercice 1 : Etude d'un algorithme itératif (7.5 points)
Soit un entier $n \in \mathbb{N}$. Un diviseur $d \in \mathbb{N}^\star$ de n est dit *propre* si $d \neq n$. Un nombre n est *parfait* si il est égal à la somme de ses diviseurs propres.
Par exemple, les diviseurs propres de $n = 6$ sont les entiers 1, 2 et 3 et on observe que $6 = 1 + 2 + 3$. Le nombre 6 est donc parfait. De même, $28 = 14 + 7 + 4 + 2 + 1$ est parfait. Les nombres parfaits sont assez rares : le suivant est 496, puis 8128. De plus, on ne sait pas aujourd'hui si il existe des nombres parfaits impairs.
L'algorithme 1 retourne True si et seulement si n est parfait. Le code `n//2+1` retourne $\lfloor \frac{n}{2} \rfloor + 1$. Pour la boucle `for`, la variable i prend les valeurs dans $\{1, \dots, \lfloor \frac{n}{2} \rfloor\}$ en ordre croissant. Le code `n%i` retourne « n modulo i », autrement dit le reste de la division euclidienne de n par i.

```
def EstParfait(n):
    sum=0
    for i in range(1,n//2+1):
        if (n%i==0):
            sum=sum+i
    return sum==n
```

Algorithme 1 : Retourne True si et seulement si n est parfait.

1. (1 point) Démontrer que la fonction `EstParfait` se termine.

2. On souhaite maintenant démontrer la validité de cette fonction. Pour cela, soit la suite $sum_{i^\star}$, pour $i^\star \in \{0, \dots, \lfloor \frac{n}{2} \rfloor\}$ telle que $sum_0 = 0$ et pour $i^\star > 0$, $sum_{i^\star}$ a pour valeur celle de la variable sum à la fin du corps de boucle pour $i = i^\star$.
   (a) (½ point) Donner les valeurs successives de la suite $sum_{i^\star}$ pour $n = 12$ et $i^\star \in \{0, \dots, 6\}$;
   (b) (½ point) Pour une valeur n fixée, exprimer un invariant de boucle $\mathcal{P}(i^\star)$ en utilisant $sum_{i^\star}$;

(c) ($2\frac{1}{2}$ points) Démontrer l'invariant de boucle $\mathcal{P}(i^\star)$ par récurrence.

3. (1 point) En déduire la validité de la fonction `EstParfait`.

4. (1 point) Peut-on identifier un pire des cas et/ou un meilleur des cas pour `EstParfait` ?
En déduire la complexité de cette fonction.

On considère maintenant l'Algorithme 2 qui retourne le plus petit entier parfait pair.

```
def PlusPetitParfaitPair():
    p=2
    while True:
        if EstParfait(p):
            return p
        p=p+2
```

Algorithme 2 : Retourne le plus petit parfait pair.

5. (a) (½ point) Démontrer la terminaison de cet algorithme.
   (b) (½ point) On souhaite maintenant modifier cet algorithme pour calculer le plus petit nombre premier impair. Que pensez vous de la terminaison de ce nouvel algorithme ?

Exercice 2 : Arbres binaires (7.5 points)

L'ensemble $ABba$ est défini inductivement de la manière suivante :

— $(b, \emptyset, \emptyset) \in ABba$

— si $G \in ABba$ et $D \in ABba$ alors $(a, G, D) \in ABba$.

1. (1 point) Dessiner un arbre de $ABba$ de taille 11 (c'est-à-dire ayant 11 nœuds au total).

2. (1 point) Donner une définition inductive de la taille $n(A)$ d'un arbre $A \in ABba$.

3. ($1\frac{1}{2}$ points) Démontrer, par induction structurelle, que la taille d'un arbre de $ABba$ est toujours impaire.

4. (3 points) Démontrer, par induction structurelle, que si $A \in ABba$ est de taille $n = 2k + 1$ alors le parcours infixe de A est égal à $[1] + [0, 1] * k$ (c'est-à-dire à la liste `[b,a,b,a,b,...,a,b]`, de longueur $2k + 1$).

5. (1 point) Dessiner un arbre de $ABba$ dont le parcours préfixe est $[a, a, b, a, b, b, a, b, a, b, a, b, b]$. Y-a-t-il plusieurs solutions ?

Exercice 3 : QCM (5 points) Un seul choix est possible. Une bonne réponse = 0.5 points. Une mauvaise réponse = -0.25 points. Si la note globale du QCM est négative, on la considère égale à 0.

1. (½ point) Pour démontrer $a \Rightarrow b$ par l'absurde, on doit :
   - ◯ démontrer que si a et b sont faux, on obtient une contradiction ;
   - ◯ démontrer que si b est faux, alors a est faux ;
   - ◯ démontrer que si a est vrai et b faux, on obtient une contradiction ;
   - ◯ Aucun des choix précédents.
2. (½ point) A quel ensemble apartient $u = \frac{n^2}{\sqrt{n}}$?
   - ◯ $u \in \Omega(n) \cap \mathcal{O}(n^2)$;
   - ◯ $u \in \Theta(n^2)$;
   - ◯ $u \in \Omega(n^2) \cap \mathcal{O}(n^2 \log n)$;
   - ◯ Aucun des choix précédents.
3. (½ point) Soit la suite $u_n = 2u_{\frac{n}{2}} + 1$ définie pour $n = 2^k$, $k \in \mathbb{N}$ et $u_1 = 1$. Quel est le terme général de u_n ?
   - ◯ $u_n = 2^n - 1$;
   - ◯ $u_n = 2n - 1$;
   - ◯ $u_n = \log_2 n - \log_2(1) + 1$;
   - ◯ Aucun des choix précédents.
4. (½ point) Dans les deux questions suivantes, on considère la fonction `mystere` dont le code suit.

```
def mystere(n):
    if (n==1):
        return 5
    return mystere(n-1)+(n+1)
```

   Une seule des affirmation est vérifiée, laquelle ?
   - ◯ `mystere(6)` retourne 30 et `mystere(9)` retourne 47 ;
   - ◯ `mystere(6)` retourne 30 et `mystere(9)` retourne 57 ;
   - ◯ `mystere(6)` retourne 23 et `mystere(9)` retourne 47 ;
   - ◯ Aucun des choix précédents.
5. (½ point) On considère la fonction `mystere` exprimée dans la question précédente. Une seule des affirmation est vérifiée, laquelle ?
   - ◯ Pour $n \geq 1$, `mystere(n)` retourne $2 + \frac{(n+1).(n+2)}{2}$;
   - ◯ Pour $n \geq 1$, `mystere(n)` retourne $5 + \frac{(n+1).(n+2)}{2}$;
   - ◯ Pour $n \geq 1$, `mystere(n)` retourne $5 + \frac{(n+1).(n)}{2}$;
   - ◯ Aucun des choix précédents.

6. (½ point) Dans les deux questions suivantes, on considère la fonction `mystereListe` dont le code suit. $L[1:]$ désigne la liste L sans son premier élement $L[0]$.

```
def mystereListe(L,x):
    res=0
    if (len(L)!=0):
        res=L[0]+x*mystere(L[1:],x)
    return res
```

Une seule des affirmation est vérifiée, laquelle ?

- ◯ La fonction ne se termine pas pour certaines listes ;
- ◯ Pour toute liste de n entiers, l'appel `mystereListe(L,x)` se termine et retourne $\sum_{i=0}^{n-1} x^i.L[i]$;
- ◯ Pour toute liste de n entiers, l'appel `mystereListe(L,x)` se termine et retourne $\sum_{i=1}^{n} x^i.L[i]$;
- ◯ Aucun des choix précédents.

7. (½ point) On suppose dans cette question que la liste L est représentée sous la forme d'une liste simplement chaînée. Quelle est la complexité de la fonction `mystereListe(L,x)` ?

- ◯ $\Theta(n^2)$;
- ◯ $\Theta(n)$;
- ◯ $\Omega(1)$ et $\mathcal{O}(\log n)$;
- ◯ Aucun des choix précédents.

8. (½ point) Soit $\mathcal{A}$ un algorithme de tri par comparaisons. Une seule assertion ne peut pas être vraie, laquelle ?

- ◯ La complexité de $\mathcal{A}$ est en $\Omega(n)$;
- ◯ La complexité de $\mathcal{A}$ est en $\Theta(n \log n)$;
- ◯ La complexité de $\mathcal{A}$ est en $\Theta(n)$.

9. (½ point) La complexité du Quicksort est en :

- ◯ $\Theta(n \log n)$;
- ◯ $\Omega(n \log n)$;
- ◯ $\mathcal{O}(n \log n)$;
- ◯ Aucun des choix précédents.

10. (½ point) La complexité du tri à bulles est en :

- ◯ $\mathcal{O}(n \log n)$ et $\Omega(n)$;
- ◯ $\Omega(n^2)$ et $\mathcal{O}(n \log n)$;
- ◯ $\Theta(n^2)$;
- ◯ Aucun des choix précédents.