

**MINI EXAM DU 29 AVRIL 2022 ( 80MN)**

La barème contient 28 points sur 20 donc je ne m'attends pas à ce que vous fassiez tout les exercices.

Lien web :

<http://www-sop.inria.fr/members/Stephane.Perennes/exam.pdf>

**1 Nids de Boucles, Invariants, analyse de programmes****Exercice 1 ( 20mn, 6pts)**

Soit le programme suivant :

```
1 def A(n):
2   for i in range(n+1):
3     print ("attention")
4     for j in range(2*i):
5       print ("nawak")
```

1. Notons  $n_1(n)$  le nombre de fois où “attention” est affiché et  $n_2(n)$  le nombre de “nawak” affichés.
2. Que valent  $n_1(n)$  et  $n_2(n)$
3. Écrire un invariant (pre/post condition portant sur  $i$  et nombre d'affichage de “attention” effectués) permettant de prouver la valeur de  $n_1(n)$ .
4. Décrire (éventuellement approximativement) l'invariant (couple pre/post condition encadrant la ligne print(nawak)) que l'on peut ajouter pour analyser le nombre de nawak affichés.
5. lesquels de ces énoncés sont correct :  
 $n_1(n) = \Omega(\sqrt{n})$ ,  $n_1(n) = \Theta(n)$ ,  $n_1(n) = \Theta(n \ln(n))$ ,  $n_1(n) = O(n^3)$ .  
 $n_2(n) = \Omega(n\sqrt{n})$ ,  $n_2(n) = \Theta(n)$ ,  $n_2(n) = \Theta(n \ln(n))$ ,  $n_2(n) = O(n^3)$ .

**Exercice 2 (10mn, 3pts)**

Soit le programme suivant :

```
1 def TestTriee(L)
2   for i in range(len(L)-1):
3     if L[i]>L[i+1]:
4       return (False)
5   return (True)
```

Montrer que la fonction teste si une liste de nombres est triée dans l'ordre croissant. Comme le programme est très simple je vous demande un invariant précis portant sur les préfixes de la liste.

1. Quel est le temps d'exécution minimum de l'algorithme (meilleure liste) ?
2. Quel est le temps d'exécution maximum de l'algorithme (pire liste) ?

3. Expliquer pourquoi si la liste est dans un ordre aléatoire le programme termine en 1 étape dans la moitié des cas.
4. quelle est la probabilité approximative ( $\Theta()$ ) de terminer après  $k$  tops (quand  $k$  est une constante et  $n$  grand ?

### Exercice 3 10mn (3pts)

```

1  def mystere(n) :
2      inter=1
3      for i in range(n) :
4          inter*=2
5      print(inter)
6      Res=0
7      for i in range(inter) :
8          Res+=1
9      return (Res)

```

Que vaut  $mystere(n)$ ? Motiver succinctement votre réponse.

On modifie le programme :

```

1  def mystere2(n) :
2      inter=1
3      for i in range(n) :
4          inter*=2
5      Res=0
6      for i in range(inter) :
7          Res+=i
8      return (Res)

```

— que vaut  $mystere2(n)$ ?

— Pour quel  $i$  a t'on  $mystere2(2) = \Theta(n^i)$ .

## 2 $\Theta, \Omega, O$

### Exercice 4 ( 5mn,2pts)

Donner la classe d'équivalence  $\Theta()$  de chacune des fonction suivante :

$$\begin{aligned}
 &n^2 \log^2 n + n \log n \\
 &n^3 + n^2 \log n \\
 &2^{n/100} + 100n^{100} \\
 &\log(n!) + 100n
 \end{aligned}$$

**rappel (Stirling) :**

$$n! \sim \frac{1}{\sqrt{2\pi}} \left(\frac{n}{e}\right)^n$$

$A(n) \sim B(n)$  signifie que le rapport  $A/B$  vaut quasiment 1 quand  $n$  est grand.

### 3 Programmation dynamique

#### Exercice 5 : Rendu monnaie ( 20mn, 8pts)

On considère 5 types de pièces : 20, 10, 5, 2, 1 euros et on souhaite réaliser une somme  $S$ . Pour  $i \in \{1, 2, 5, 10, 20\}$  on dispose d'au plus  $q_i$  pièce de valeur  $i$ . Au total il y a  $n$  pièces.

- Écrire un programme idiot qui teste tous les cas possibles (en  $2^n$ ) et retourne une solution si elle existe.
- Écrire un programme dynamique en 5 étapes qui génère successivement les tables  $T_k$  des rendus monnaie possibles avec les  $k$  premiers types de pièces. Par exemple  $T_1 = \{0, 20, 40, \dots, 20q_{20}\}$ . Pour calculer  $T_2$  (pièces 20, 10) on utilisera  $T_1$  et on fera varier le nombre de pièces de 10 euros.
- Utiliser la table  $T_5$  pour résoudre le problème du rendu monnaie.

### 4 Algorithmes Gloutons

#### Exercice 6 : mise en boîte (15mn, 6pts)

On dispose de boîtes identiques de volume  $V$  et d'une liste d'objets  $O_i, i \in I$ , l'objet  $i$  est de volume  $V_i$ . Il y a  $|I| = n$  objets. On doit affecter les objets aux boîtes en essayant d'utiliser un nombre minimum de boîtes.

- Combien y a-t-il de solutions ? Décrire brièvement (en langage naturel) un algorithme exhaustif
- Proposer un algorithme glouton qui va parcourir les objets et choisir ou les placer séquentiellement. Le choix de l'objet et de la boîte s'effectue selon une heuristique de votre choix.
- Proposer une seconde heuristique.