

## Devoir 1 (6%) CSI2510

A remettre sur le Campus Virtuel avant 23h55 le 1<sup>er</sup> octobre 2018

### Question 1. [8 points = 8\*1pt]

Démontrer si les énoncés ci-dessous sont vrais ou faux. Pour ce faire identifier des valeurs pour les constantes  $c$  et  $n_0$  telles que définies dans les définitions de Grand O,  $\Theta$  et  $\Omega$ .

- a)  $\log((n-2)(n+3))$  est  $O(n^2)$
- b)  $\log((n-2)(n+3))$  est  $\Omega(n \log n)$
- c)  $(n + \log(n))(n+8)$  est  $O(n)$
- d)  $(n + \log(n))(n+8)$  est  $\Theta(n \log n)$
- e)  $n^2+2n^2+3n^2+4n^2+\dots+(n)n^2$  est  $O(n^2)$
- f)  $n^2+2n^2+3n^2+4n^2+\dots+(n)n^2$  est  $\Omega(n^2)$
- g)  $50n+ 0.001n^3$  est  $O(n^4)$
- h)  $50n+ 0.001n^3$  est  $\Omega(n^4)$

### Question 2. [4 points = 2+2]

Soit l'algorithme ci-dessous permettant de calculer le produit de deux nombre entiers  $x$  et  $y$ .

```
1.  \ \ x >=y >=0
2.  int Mult (int x, int y){
3.      int R=0;
4.      int a=x, b=y;
5.      while (b>0){
6.          if ( b %2 == 1)
7.              R+= a;
8.          a<<=1;
9.          b>>=1;
10.     }
11.     return R;
12. }
```

- a) Donner le nombre exact de décalage de bits (i.e. l'exécution des lignes 8 et 9) effectués en fonction de la valeur de  $y$ . Justifier votre réponse.
- b) Donner l'ordre de grandeur de la complexité de cet algorithme en termes de Grand O. Spécifier un ordre de grandeur aussi simple que possible.

**Question 3.** [6 points = 2+2+2]

Un algorithme prend 1 seconde sur votre machine pour traiter un ensemble de  $n$  données. Vous remplacez votre machine contre une machine dix fois plus rapide. Quelle est la taille maximale de l'ensemble de données que pouvant maintenant traiter en 1 seconde si vous supposez que le temps d'exécution de l'algorithme est proportionnel à

- a)  $n$   
 b)  $n^2$   
 c)  $\log n$

?

Justifier vos réponses.

**Question 4.** [6 points]

Un tableau  $A$  de grande taille contient  $N$  entiers ayant une valeur entre 0 et  $n-1$  (inclusivement). L'histogramme de  $A$  sera un tableau  $H$  de taille  $n$ . L'entrée  $i$  de ce tableau contient le nombre d'éléments ayant la valeur  $i$  dans le tableau  $A$ .

On vous propose l'algorithme suivant permettant de calculer l'histogramme de  $A$ .

```
for (i=0 ; i<n; i++) {
  H[i]=0;
  for (j=0; j<N; j++) {
    if (A[j] == i)
      H[i]++;
  }
}
```

- a) (2 pts) Donner l'ordre de grandeur de la complexité de cet algorithme en termes de Grand O en fonction de  $N$ , pour les deux cas suivants:
- Le tableau contient des nombres binaires (0 ou 1), i.e.  $n=2$
  - $n$  est à peu près égal à  $N$
- Spécifier un ordre de grandeur aussi simple que possible.
- b) (4 pts) Proposer un algorithme plus efficace permettant de calculer l'histogramme de  $A$ . Quelle est la nouvelle complexité de l'algorithme pour les deux cas de la question précédente.