Université Abderrahmane Mira de Béjaïa **Département de RO / Master 1**
2015/2016

Examen Algorithmique Avancée

<u>Les calculatrices et autres appareils électroniques sont interdits. Les téléphones mobiles doivent être éteints et rangés dans les sacs.</u>

Exercice 1.

Question 1) Rappeler les définitions des notations O, Ω, Θ pour les fonctions à valeur dans R^{*+}.
Question 2) Montrer que $\sum_{k=0}^{n} 2^k = \Theta(2^{\mathrm{n}}),\ \forall k, n \in N$.
Question 3) On a la récurrence $T(1) = 1\ et\ \forall n > 1, T(n) = n + T\left(\frac{n}{2}\right)$.

1. Montrer que pour tout $k \geq 0, T\left(2^k\right) = 2^{k+1} - 1$.
2. En déduire que $T(n) = \Theta(\mathrm{n})$.

Exercice 2.

Soit le programme suivant :

```
int F(int n){
int i, j = 0; int x = n;
while( x > 0 ){
      for(i = 1; i <= 2x; i++)
            j++;
      x = x/3; }
return j; }
```

a) Dérouler en détail le programme pour calculer F(9) ?
b) Quelle est la complexité de ce programme en nombre d'instructions? justifier votre réponse.

Exercice 3.

Soit $\mathrm{A}[1..n]$ un tableau de n entiers différents triés par ordre croissant : pour tout $1 \leq \mathrm{i} \leq \mathrm{n}$ on a A[i]<A[i+1]. Un point fixe de A est un indice i tel que A[i] = i.
Par exemple, le tableau A = [−12, 1, 3, 4, 8, 9] a deux points fixes : 3 et 4.
1. Donner un algorithme linéaire permettant de connaître le nombre de points fixes d'un tableau A de taille n.
2. Donner un algorithme efficace de type diviser pour régner qui renvoie VRAI si le tableau a au moins un point fixe et FAUX sinon. Quelle est sa complexité en nombre de comparaisons ?
3. On donne un tableau A et un point fixe de ce tableau i. Comment peut-on décider en temps constant s'il existe un autre point fixe ?

Exercice 4.

On suppose un programme mystère qui affiche à l'écran les résultats suivant :

```
mystère(1) = 1
mystère(2) = 1 1 1
mystère(3) = 2 1 1 1 2
mystère(4) = 3 2 1 1 1 2 3
mystère(5) = 4 3 2 1 1 1 2 3 4
```

1. Ecrire un algorithme récursif qui pour un entier n affiche à l'écran mystère(n).
2. Prouver par récurrence que mystère(n) affiche 2n − 1 nombre à l'écran.