Examen

Exercice 1 Soit $T(n)$ le temps d'exécution d'un algorithme en fonction de n satisfaisant $T(n) = 8T(n/2) + f(n)$. Donnez la valeur asymptotique $\Theta(\cdot)$ de $T(n)$ dans les tois cas suivants: (vous appliquerez le théorème du cours en précisant la valeur de ε)

1) $f(n) = 3n^2 + 4 \lg n$;

2) $f(n) = n^3 + 7n^2$;

3) $f(n) = \frac{n^4}{9}$; (vous appliquerez le corollaire en donnant d'abord une constante k et un n_0 tels que $f(2n) \leq kf(n)$ pour tout $n \geq n_0$).

Exercice 2 On considère la complexité de la fonction `main` ci-dessous en fonction du paramètre d'entré n:

```
import java.util.Random;
public class Multi{
  public static int increm(int n){if(n==0) return 0; else return ++n;}
  public static int decrem(int n){if(n==0) return 0; else return --n;}
  public static int MultiDecrem(int n, int k){
    while(n != 0 && k !=0){
      n=decrem(n);
      k--;
    }
    return n;
  }
  public static void main (String...args){
    int n = args[0];
    final int seed = args.length > 1 ?
         Integer.parseInt(args[1]) : (int) System.currentTimeMillis();
    final Random random = new Random(seed);
    int compt=n;
    while(compt--!=0){
      int alea = random.nextInt(3);
      if(alea==0) n=increm(n);
      if(alea==1) n=decrem(n);
      if(alea==2) n=MultiDecrem(n,n/2);
    }
  }
}
```

1) Montrer que la complexité est $\Omega(n)$ et $O(n^2)$.

2) Montrer que la complexité est $\Theta(n)$.

Exercice 3 Soit T un ensemble, $r \in T$, et une fonction p de $T \setminus \{r\}$ dans T. Montrer que T est un arbre si et seulement si $p^k(x) = x$ implique $k = 0$ pour tout $x \in T \setminus \{r\}$.

Exercice 4 Donnez le code Java d'un algorithme `double pow(double x, int n)` qui éleve x à la puissance n et dont la complexité est $O(\lg n)$. Démontrer que la complexité de votre algorithme est bien $O(\lg n)$.

Exercice 5 Donnez le tas-min T obtenu par construction à partir du tableau $A = 2,8,9,7,12,4,7,53,8$. Elaborer un algorithme de tri de tableau de n entiers en $O(n \log n)$.