Examen

Exercice 1
```
public class Queens {
  final int SIZE=4;
  int [] solution;
  boolean [] ligne, diag1, diag2;
  public Queens(){
    solution = new int[SIZE];
    ligne = new boolean[SIZE];
    diag1 = new boolean[2*SIZE-1];
    diag2 = new boolean[2*SIZE-1];
    for (int j=0; j < SIZE; ++j)
      ligne[j]=diag1[j]=diag2[j]=true;
    for (int j=SIZE; j < 2*SIZE-1; ++j)
      diag1[j]=diag2[j]=true;
  }
  void essayer(int c){
    for (int l=0; l < SIZE; ++l){
      if(ligne[l] && diag1[l+c] && diag2[l-c+SIZE-1]){
        solution[c]=l;
        ligne[l]=diag1[l+c]=diag2[l-c+SIZE-1]=false;
        if(c!=SIZE-1) essayer(c+1);
        ligne[l]=diag1[l+c]=diag2[l-c+SIZE-1]=true;
      }
    }
  }
}
```

1) Quelles modifications faut-il apporter au code ci-dessus, pour que, étant déclaré

```
Queens Q = new Queens();
```

l'appel

```
Q.essayer(0);
```

a) Affiche le premier tableau `solution` trouvé;

b) Affiche tous les tableaux `solution` trouvés.

2) Donnez tous les contenus de tous les tableaux `solution` trouvés.

3) On remplace `final int SIZE=4;` par `final int SIZE=5;` donnez les contenus des 3 premiers tableaux `solution` trouvés.

Exercice 2 On rappelle que l'insertion et la recherche d'un élément dans un arbre rouge et noir sont en $O(\lg n)$.

Soient un entier b et des entiers $a_1, \ldots, a_n$. Décrire un algorithme en $\Theta(n \log n)$ qui dit s'il y a deux entiers a_i,a_j dont la somme vaut b.

Exercice 3 1) Quelle est la complexité de l'algorithme de Strassen pour la multiplication de deux matrices $n \times n$?

2) Soient a,b,c,d des réels. Le produit de deux nombres complexes $x = a+ib$ et $y = c+id$ est $xy = (ac - bd) + i(ad + bc)$. Donnez un algorithme qui calcule xy en effectuant que 3 produits de deux nombres réels au lieu de 4.

Exercice 4
```
public static int algo(int n){
  if(n==0) return 1;
  fonct(n);
  return algo(n/2)+algo(n/2)+algo(n/2)+algo(n/2);
}
```

1) Donnez la valeur asymptotique $\Theta(\cdot)$ du temps d'exécution de `algo` dans les cas suivants:

a)
```
public static void fonct(int n){
 compt=0;
 for(int i=0;i<n;++i) compt++;
 System.out.println(compt + "passages");
}
```

b)
```
public static void fonct(int n){
 compt=0;
 for(int i=0;i<n;++i)
    for(int j=i;j<n;++j) compt++;
 System.out.println(compt + "passages");
}
```

c)
```
public static void fonct(int n){
 compt=0;
 for(int i=0;i<n*n;++i)
    for(int j=i;j<n*n;++j) compt++;
 System.out.println(compt + "passages");
}
```

2) Quelle valeur renvoie `algo(8)` dans les cas a-b-c?