Examen

Exercice 1
```
public class Queens {
  final int SIZE=5;  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) Après la déclaration `Queens Q = new Queens()` et lors de l'appel `Q.essayer(0)`, donnez tous les contenus du tableau `solution` juste après, et à chaque fois, que le test `if(c!=SIZE-1)` soit faux.

2) Quelle était le contenu du tableau `diag2` juste avant que `Q.essayer(3)` ne soit appelée pour la première fois ?

Exercice 2
```
public static double algo(double n){
  if(n<1.0) return 1.0;
  int x = 2*algo(n/2) + 2*algo(n/2);
  for(int i=0; i<n; i++) x+=2*i;
  return x;
}
```

1) Que renvoient `algo(0)`, `algo(1)`, `algo(2)`, `algo(4)`, `algo(8)` ?

2) Quelle est la complexité de `algo(n)` ?

3) Modifier `algo(n)` pour que la complexité soit $\Theta(n^2)$.

4) Modifier `algo(n)` pour que la complexité soit $\Theta(\log n)$.

5) Modifier `algo(n)` pour que la complexité soit $\Theta(n^{\log_2 3})$.

Exercice 3 Soit (T,r,p) avec T un ensemble fini non vide, $r \in T$, et p une fonction de $T \setminus \{r\}$ dans T. Pour toute fonction p et tout entier k, on définit la fonction:

$$p^k(x) = \begin{cases} x & \text{si } k = 0 \\ p(p^{k-1}(x)) & \text{si } k > 0 \end{cases}$$

On dit que (T,r,p) est un arbre si pour tout $x \in T \setminus \{r\}$, il existe un entier $k \geq 1$ tel que $p^k(x) = r$. Pour tout arbre (T,r,p) et tout $x \in T \setminus \{r\}$, on note $d(T,r,p,x)$ le plus petit k tel que $p^k(x) = r$. On note $h(T,r,p)$ la valeur maximum de $d(T,r,p,x)$ sur tous les $x \in T \setminus \{r\}$.

1) Montrer que $(\{1,\ldots,n\},1,\lfloor x/2 \rfloor)$ est un arbre.

2) Montrer que

$$h\big((\{1,\ldots,n\},1,\lfloor x/2 \rfloor)\big) = O(\log n).$$

3) Donner une fonction p telle que $(\{1,\ldots,n\},1,p)$ soit un arbre tel que:

$$h(\{1,\ldots,n\},1,p) = \Omega(n).$$

Exercice 4 1) Donnez la composition du tableau $A = (51,7,8,16,45,2,71,8,55,13,46)$ après l'appel à la procédure de construction de tas-min.

2) Élaborer un algorithme de tri en $O(n \lg n)$ basé sur l'utilisation des tas-min.

Exercice 5

```
final int n = 10; final int x = 9; final int seed = 134582544;
boolean[][] mat = new boolean [n][n];
final Random generator = new Random(seed);
for(int i = 0 ; i < x ; i++)
        mat[generator.nextInt(n)][generator.nextInt(n)]=true;
boolean test=true;
while(test){
    test=false;
    for(int i = 0 ; i < n ; i++){
      for(int j = 0 ; j < n ; j++){
        if(!mat[i][j]){
          int vois=0;
          if (i>0 && mat[i-1][j]) vois++;
          if (i+1<n && mat[i+1][j]) vois++;
          if (j>0 && mat[i][j-1]) vois++;
          if (j+1<n && mat[i][j+1]) vois++;
          if (vois >= 2){mat[i][j]=true; test=true;}
        }
      }
    }
 }
```

Montrez qu'à la sortie du `while`, on a `mat[i][j]=false` pour au-moins un couple `i,j`.