complexité
Examen

Partie 1

Pour implémenter un compteur binaire sur k bits, nous utilisons un tableau A de k éléments. L'algorithme qui permet d'ajouter 1 au compteur binaire est le suivant :

```
int i ;
for(i =0 ; i<k && A[i]==1 ; ++i)
   A[i]=0 ;
if(i<k) A[i]=1 ;
```

Question 1. (3 points)

Quel est le nombre maximum d'affectations en rouge effectuées lors d'une incrémentation.

Question 2. (7 points)

Nous nous proposons de calculer le nombre moyen d'affectations en rouge effectuées lors d'une incrémentation. Pour cela nous compterons d'abord le nombre total d'affectations en gras effectuées pour $2^k - 1$ incrémentations. Etablir la relation de récurrence qui donne ce nombre, puis résoudre la récurrence.

Partie 2

L'enveloppe convexe d'un ensemble P de Points est le plus petit polygone convexe C tel que tous les points de P sont soit sur C soit à l'intérieur de C.

L'algorithme « naïf » de calcul de l'enveloppe convexe est le suivant :

On calcule tous les points strictement intérieurs à l'ensemble de points : les points de l'ensemble qui ne sont pas strictement à l'intérieur sont des points de l'enveloppe.

Un point p est strictement à l'intérieur de l'ensemble, s'il existe 3 autres points (p_0, p_1, p_2) qui forme un triangle contenant le point p.

Le calcul de la surface d'un triangle donné par 3 points (p_0, p_1, p_2), s'exprime de la façon suivante : $S = \frac{(p_1.x - p_0.x)*(p_2.y - p_0.y) - (p_2.x - p_0.x)*(p_1.y - p_0.y)}{2}$, cette surface est <0 si lors du parcours (p_0, p_1, p_2) on tourne dans le sens inverse des aiguilles d'une montre et >0 dans le cas contraire. Un point p est à l'intérieur du triangle (p_0, p_1, p_2), si et seulement si les surfaces des triangles (p_0, p, p_1), (p_1, p, p_2) et (p_2, p, p_0)sont toutes de même signe.

Le calcul de l'enveloppe convexe s'exprime simplement en énumérant tous les points et tous les triangles :

```
void algoNaïf(Vector lesPoints, Vector enveloppe){
    // un point est sur l'enveloppe s'il n'est à l'intérieur
```

```
        // d'aucun triangle formé par les autres points
        enveloppe.clear();
        // calcul des points strictement intérieurs
        Vector nonEnveloppe = new Vector();
        for( int i = 0; i< lesPoints.size(); ++i)
          for( int j = i+1; j< lesPoints.size(); ++j)
             for( int k = j+1; k< lesPoints.size(); ++k)
               for(int l = 0 ; l< lesPoints.size(); ++l)
                    // le point l est-il à l'intérieur du
                    // triangle (i, j, k)
                   if(l!=i && l!=j && l!=k &&
                       interieur(lesPoints, i, j, k, l))
                    nonEnveloppe.addElement(lp.elementAt(i));
        // calcul de l'enveloppe
        for( int i = 0; i< lesPoints.size(); ++i)
            enveloppe.addElement(lesPoints.elementAt(i));
        enveloppe.removeAll(nonEnveloppe);
    }
```

Question 3. (10 points)

Donner une estimation du temps de calcul en fonction du nombre *n* de points de l'ensemble dont on veut calculer l'enveloppe convexe. On calculera le nombre d'exécutions de l'instruction surlignée.