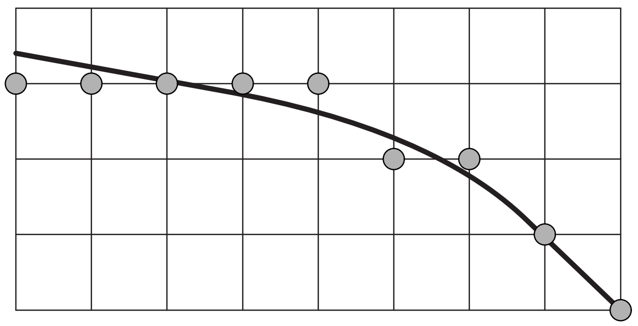
Dessine moi...

http://www.enseignement.polytechnique.fr/profs/informatique/Luc.Maranget/IF/14/

1 Dessine moi un cercle

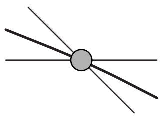
Le but de cet exercice est le dessin d'un cercle à l'écran, en suivant le principe de l'algorithme de Bresenham pour le tracé des droites. On cherche à produire une approximation de Freeman : lorsque la courbe traverse une des lignes de la grille, on « allume » le point de la grille qui est le plus proche de l'intersection.



On note que l'approximation de Freeman est conservée par les transformations qui laissent la grille invariante.

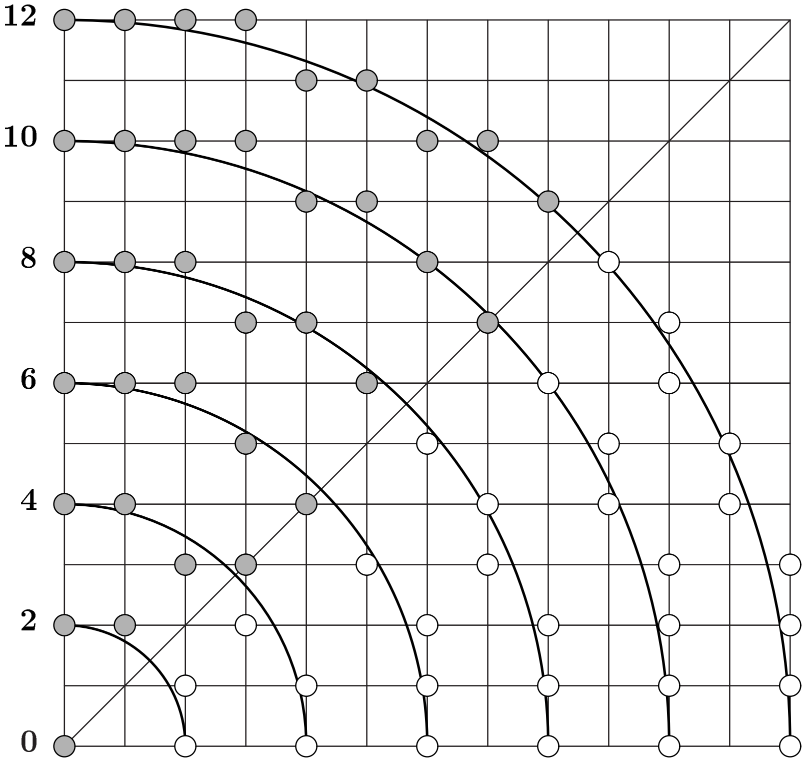
En pratique on construit l'approximation de proche en proche en suivant les mouvements du roi sur un échiquier. On pose que la courbe est définie par une fonction continue et suffisamment lisse $y = f(x)$ sur un certain domaine.

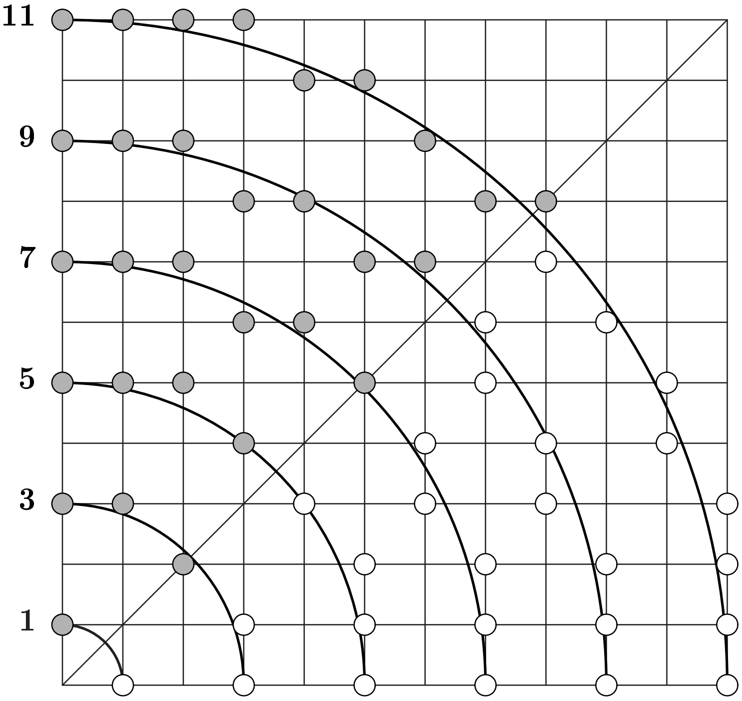
1. On suppose que f est telle que, sur le domaine concerné et pour tout point de la courbe, la courbe est entièrement comprise dans les octants W-NW et E-SE (cela doit revenir à une hypothèse f décroissante, convexe et de dérivée comprise entre 0 et -1)

   Montrer que les mouvements possibles sont limités à deux et concevoir un test simple pour choisir.
2. Considérer le cas d'un cercle centré à l'origine et de rayon R entier, on définira un domaine adéquat, un premier point et la condition de passage au point suivant.
3. Écrire un premier jet du programme qui dessine la fraction de cercle définie ci-dessus. On suppose donnée une méthode `pxl(int x, int y)` pour allumer le point (x, y).
4. Modifier le programme simple afin qu'il n'effectue que des calculs entiers et aucune multiplication entre deux termes non-constants.
5. Examiner sérieusement la délicate question de la limitation du domaine. On pourra s'inspirer des dessins de quarts de cercle donnés par la figure 1. Cette figure illustre l'approximation de Freeman. Les points gris (x_0, y_0) sont construits par une marche du roi sous condition $x_0 \leq y_0$ et les point blancs sont les symétriques (y_0, x_0) des gris pour $x_0 < y_0$.
6. Examiner la question bien plus facile du dessin d'un cercle de rayon entier R centré en un point de coordonnées entières (x_c, y_c). On supposera que l'écran est en mode XOR, c'est à dire qu'allumer un point allumé revient à éteindre.

Fig. 1 – Quelques quarts de cercle pour des rayons entiers

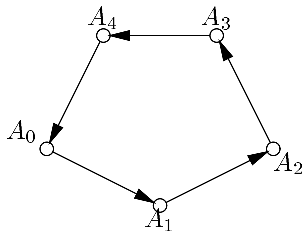




2 Enveloppe convexe incrémentale

Le but de cet exercice est la présentation d'une méthode finalement simple pour construire l'enveloppe convexe d'un ensemble de points donnés au fur et à mesure. Dans tout l'exercice on suppose que les points sont en *position générale* c'est à dire qu'il n'existe pas 3 points alignés.

Les polygones (convexes) seront représentés par la liste de leurs points pris dans le sens trigonométrique. On note C_n un polygone à n côtés. Voici par exemple un polygone $C_5 = A_0, A_1, A_2, A_3, A_4$.



1. Caractériser les points P intérieurs à C_n à l'aide des déterminants $\det(\overrightarrow{A_iP}, \overrightarrow{A_iA_{i+1}})$ (on considère aussi la dernière arête $\overrightarrow{A_{n-1}A_0}$).

   On remarquera que chaque vecteur $\overrightarrow{A_iA_{i+1}}$ divise le plan en deux demi-plans caractérisés par le signe du déterminant $\det(\overrightarrow{A_iP}, \overrightarrow{A_iA_{i+1}})$
2. On se donne un point supplémentaire P, donner une méthode pour construire l'enveloppe convexe des points $A_0, A_1, A_2, A_3, A_4, P$.
3. En déduire une méthode pour construire l'enveloppe convexe de n points donnés au fur et à mesure. Estimer le coût de cette méthode.
4. Montrer que si les points sont présentés triés selon l'ordre lexicographique, alors on peut concevoir un algorithme linéaire en le nombre de points. Bien identifier la structure de données nécessaire.
5. On se donne une classe des points (et listes de points).

   ```
   class Point {
     int x, y ;
     PointList next ;

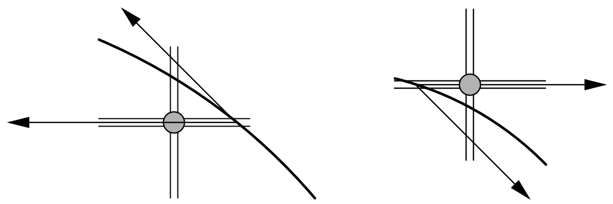
     PointList (int x, int y, PointList next) {
      this.x = x ; this.y = y ; this.next = next ;
     }
   }
   ```

   Programmer la méthode d'extension du polygone convexe `p` par un point `q`.
6. Se passer de l'hypothèse de la position générale.

Solutions des exercices

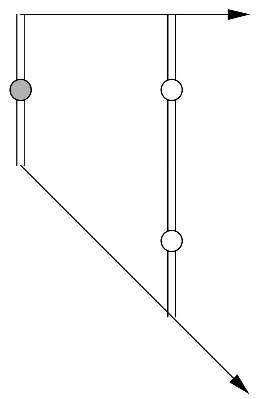
Exercice 1

1. Soit $(x0, y0)$ approximation, la courbe intersecte nécessairement la « croix » de taille 1 centrée en $(x0, y0)$. On utilise facilement l'interprétation géométrique pour montrer que toute intersection de la courbe avec la barre horizontale entraîne l'intersection avec la barre verticale.

   Soit donc le point (x_0, y_0) approximation, par le résultat précédent la courbe intersecte le segment unité centré en (x_0, y_0). On suppose que $x_0 + 1$ appartient au domaine de f. Par hypothèse la courbe intersecte le segment unité centré en (x_0, y_0), et l'axe vertical $x = x_0 + 1$ au point $(x_0 + 1, f(x_0 + 1))$.

   D'après l'interprétation géométrique, on voit que l'intersection avec la verticale $x = x_0 + 1$ se situe nécessairement dans l'un des deux segments unité centrés en (x_0+1, y_0) ou (x_0+1, y_0-1) (soit $y0 - \frac{3}{2} \leq f(x_0 + 1) \leq y_0 + \frac{1}{2}$ en formules).

   On devra donc allumer l'un des point $(x_0 + 1, y_0)$ ou $(x_0 + 1, y_0 - 1)$, pour savoir lequel il suffit de savoir si la courbe passe au dessus ou au dessus de leur milieu, $(x_0 + 1, y_0 - \frac{1}{2})$. En règle générale on comparera $f(x_0 + 1)$ et $y_0 - \frac{1}{2}$.
2. Dans le cas du cercle de rayon R centré à l'origine.
   - Le domaine est limité au huitième de cercle compris entre le nord et le nord-est.
   - Un premier point est connu : le point $(0, R)$.
   - Le test de choix entre $(x_0 + 1, y_0)$ et $(x_0 + 1, y_0 - 1)$ repose sur la position du point $(x_0 + 1, y_0 - \frac{1}{2})$ soit sur le signe de

     $$d = (x_0 + 1)^2 + (y_0 - \frac{1}{2}) - R^2$$

     Si d est positive alors le point suivant est $(x_0 + 1, y_0 - 1)$, sinon c'est $(x_0 + 1, y_0)$.
3. ```
   static void cercle(int r) {
     int x = 0 ;
     int y = r ;
   ```
   ```

```
    while (x <= y) {
      pxl(x,y) ;
      double d = (x+1)*(x+1) + (y-0.5)*(y-0.5) - r*r ;
      x++ ;
      if (d > 0) { y-- ; }
    }
  }
```

On notera que l'on évite la question de la limitation du domaine vers la droite. En effet, les raisonnements du 1. ne sont certainement pas tous valables pour le dernier point affiché.

4. Pour éviter les multiplications il faut calculer incrémentalement la quantité d. Initialement on a $d = 1^2 + (R - \frac{1}{2})^2 - R^2 = 1 + \frac{1}{4} - R$. On remarque ensuite qu'augmenter x de 1 revient à augmenter d de $2x+3$ et que diminuer y de 1 revient à diminuer y de $2y-2$. En effet on a d'une part $((x+1)+1)^2-(x+1)^2 = (2x+3)(2-1)$ et $((y-1)-\frac{1}{2})^2-(y-\frac{1}{2})^2 = (2y-2)(-\frac{3}{2}+\frac{1}{2})$. Par ailleurs pour x et y entiers, $e = d - \frac{1}{4}$ est entier et les signes de d et e sont identiques. La nullité de e entraînant $d = \frac{1}{4} > 0$. (C'est-à-dire que le cas $e = 0$ est en fait $d > 0$). On peut donc recoder la méthode `cercle` ainsi :

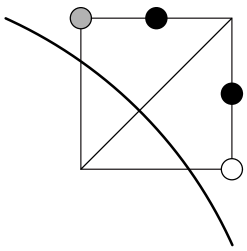
```
static void cercle(int r) {
  int x = 0 ;
  int y = r ;
  int e = 1-r ;

  while (x <= y) {
    pxl(x,y) ;
    if (e >= 0) {
      y-- ; e -= 2*y ;
    }
    x++ ; e += 2*x+1 ;
  }
}
```

5. Le cas délicat concerne bien évidemment le dernier point gris (vers la droite) et en particulier les points allumes avec $x_0 = y_0$. Les rayons 3, 6, 7 et 10 montrent que l'on doit les allumer au moins quand le mouvement qui y aboutit est sud-est. Les rayons 4 et 11 encourageraient plutôt de les oublier.

   Enfin tous ces exemples semblent montrer que procéder à la marche du roi pour $x_0 \leq y_0$ produit bien une approximation de Freeman, mais il faut le montrer !

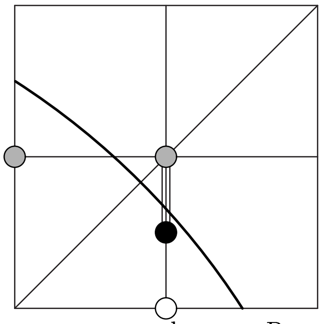
   On considère d'abord le cas où le dernier point allumé est de la forme $(x_0, x_0 + 1)$. Par hypothèse le point $(x_0 + 1, x_0 + 1)$ n'est pas allumé. Et donc le point milieu $(x_0 + 1, x_0 + \frac{1}{2})$ est au dessus du cercle. Par symétrie selon la diagonale, le point $(x_0 + \frac{1}{2}, x_0 + 1)$ est également au dessus du cercle et le point $(x_0 + 1, x_0 + 1)$ ne doit pas être allumé.



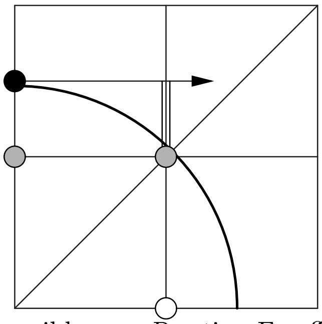
(Les deux points milieu sont en noir, le dernier de la marche est en gris, et le point allumé par symétrie est en blanc.) Le dessin illustre le cas où le dernier point de la marche est au dessus de l'arc, mais le raisonnement ne fait pas intervenir cette hypothèse.

Supposons maintenant que le dernier point allumé est sur la diagonale, c'est-à-dire qu'il est de la forme (x_0, x_0). On considère alors deux sous-cas.

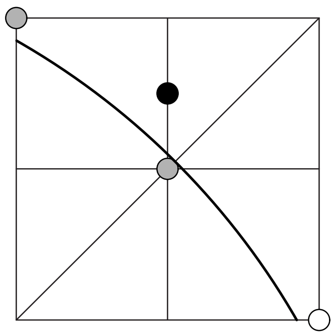
- Si le dernier mouvement est vers l'est, alors le point milieu $(x_0, x_0 - \frac{1}{2})$ est situé à l'intérieur du cercle (ou sous l'arc). On distingue alors encore deux sous-cas.
  - Si (x_0, x_0) est en dehors du cercle alors le cercle intersecte le demi-segment unité vertical inférieur centré en (x_0, x_0) et ce point doit être allumé.

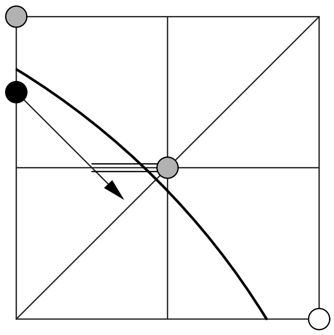
    Il s'agit du cas disgracieux obtenu par exemple pour $R = 4$ et $R = 11$.
  - Sinon, nous savons par ailleurs que l'arc passe à une distance verticale inférieure à $\frac{1}{2}$ de l'avant dernier point $(x_0 - 1, x_0)$. En appliquant l'interprétation géométrique vers la droite au point de passage de l'arc par l'axe vertical $x = x_0 - 1$ on voit que l'arc coupera l'axe $x = x_0$ sous l'ordonnée $x_0 + \frac{1}{2}$.

    Ce cas me semble d'ailleurs impossible avec R entier. En effet, x_0 est ici la partie entière de $2^{-\frac{1}{2}}R$, on a donc $2x_0^2 < R^2$. Par ailleurs, on a $(x_0 - 1)^2 + (x_0 + \frac{1}{2}) \geq R^2$ et donc $x_0 \leq 2x_0^2 - R^2 + \frac{5}{4}$, soit encore $x_0 < \frac{5}{4}$ et donc $x_0 = 1$. Soit finalement $R = 1$ ou $R = 2$ qui ne correspondent pas à la configuration (voir la figure 1). De fait cette configuration survient pour R compris entre $2^{\frac{1}{2}}$ et $\frac{3}{2}$.
- Si le dernier mouvement est vers le sud-est, alors le point milieu $(x_0, x_0 + \frac{1}{2})$ est situé à l'extérieur du cercle.
  - Si le point (x_0, x_0) est à l'intérieur, on peut conclure directement.

  - Sinon, l'arc intersecte nécessairement l'axe $x = x_0 - 1$ au dessus du point $(x_0 - 1, x_0 + \frac{1}{2})$ et on peut conclure par l'interprétation géométrique vers la droite à partir de ce point d'intersection et jusqu'à l'axe $y = x_0$.



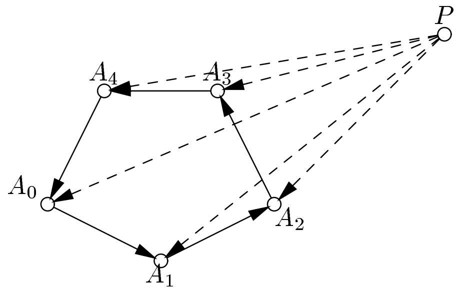
6. De façon générale, si (x, y) est un point de la marche du roi pour le cercle de rayon R centré sur l'origine, il faut allumer les huit points $(x_c + x, y_c + y)$, $(x_c - y, y_c + x)$, $(x_c - x, y_c - y)$, $(x_c + y, y_c - x)$ (correspondant au point de la marche translaté et à trois rotation d'un quart de tour) et $(x_c + y, y_c + x)$, $(x_c - x, y_c + y)$, $(x_c - y.x_c - x)$, $(x_c + x, y_c - y)$ (le symétrique selon la diagonale et trois rotation d'un quart de tour).

   Toutefois, il faut éviter la deuxième série de points dans les cas $x = 0$ et $x = y$.

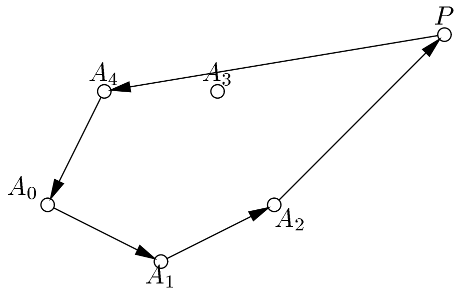
   On notera bien que c'est la conservation de l'approximation de Freeman par toutes ces isométries qui garantit que nous construisons bien une approximation de Freeman du cercle de rayon R centré en (x_c, y_c).

Exercice 2

1. L'intérieur du polygone convexe est défini par des déterminants $\det(\overrightarrow{A_iA_{i+1}}, \overrightarrow{A_iP})$ tous positifs.
2. Pour une arête donnée la convexité de C_n entraîne que l'arête $\overrightarrow{A_iA_{i+1}}$ est visible de P, si et seulement si le déterminant $\det(\overrightarrow{A_iA_{i+1}}, \overrightarrow{A_iP})$ est négatif. Intuitivement, un rayon lumineux partant de P et frappant l'arête pointe alors vers l'intérieur de C_n. Un sommets est éclairé lorsqu'une au moins des arêtes qui le touchent est visible. Il existe alors deux sommets importants, le premier sommet visible et le dernier en suivant le sens positif de rotation (sur la figure A_2 et A_4).



   Or, on peut fabriquer l'enveloppe convexe de C_n union $\{P\}$ en supprimant toutes les arêtes visibles de C_n et en ajoutant une arête du premier sommet visible à P suivie d'une arête de P au dernier sommet visible. On obtient alors le polygone.



3. La méthode consiste à attendre d'avoir trois points (un triangle) et à les ordonner dans le sens positif ($\det(\overrightarrow{A_0A_1}, \overrightarrow{A_1A_2}) > 0$). On peut ensuite construire l'enveloppe convexe incrémentalement, selon la question précédente. Une extension peut demander le parcours complet des points de l'enveloppe (si par exemple A_0 est le dernier point visible). Le coût est donc au pire quadratique. Il est atteint lorsque tous les points sont dans l'enveloppe convexe. et que les parcours commencent au dernier sommet visible.
4. Avec des points triés, chaque nouveau point appartient nécessairement à l'enveloppe convexe construite et au tour suivant ce point sera nécessairement éclairé. On peut alors trouver le dernier point éclairé en tournant dans le sens trigonométrique et le premier point éclairé en tournant dans l'autre sens. Une liste doublement chaînée permet cette opération. Si on trouve p côtés éclairés, on aura calculé $p+2$ déterminants (coût en p), et on enlèvera $p-1$ points (opération en coût constant). Un point donné est ajouté une fois et enlevé au plus une fois. L'algorithme est linéaire, car on peut associer un coût constant à chaque point (un déterminant si le point est enlevé, deux déterminants et une modification de la liste des sommets au moment de l'ajout).
¡¡¡¡¡¡ index.tex
5. Voici le code d'une classe `Convexe` qui réalise le calcul incrémental de l'enveloppe convexe. Les sommets de l'enveloppes sont dans le champ `cvx` sous forme d'une liste bouclée de points. Cette structure de donnée délicate à manipulér est ici indiquée. En effet, une telle liste ressemble fortement aux dessins des enveloppes : des points reliés par des flèches.

```
class Convexe {
  PointList cvx ; // Liste des sommets. Attention, c'est une liste bouclée

  // Travail sur l'enveloppe convexe.

  // Renvoie un entier r
  //     - r < 0  -> q est à droite du vecteur p1 p2
  //     - r > 0  -> q est à gauche du vecteur p1 p2
  //     - r == 0 -> q p1 p2 alignés

  static private int det(PointList q, PointList p1, PointList p2) {
    int r =
      (q.y - p1.y) * (p2.x - p1.x) - (q.x - p1.x) * (p2.y - p1.y) ;
    return r ;
  }


  // Fabrication d'un triangle comme liste bouclée de trois points
  static private PointList triangle (PointList p1, PointList p2, PointList p3)
    PointList r3 = new PointList(p3.x, p3.y, null) ;
    PointList r2 = new PointList(p2.x, p2.y, r3) ;
    PointList r1 = new PointList(p1.x, p1.y, r2) ;
    r3.next = r1 ;
```

```
    return r1 ;
  }

  // Étendre l'enveloppe convexe (liste bouclée cvx)
  // Le point ajouté est le premier de l'argument p
  // etendre renvoie un booléen, qui indique un changement de l'enveloppe.

 boolean etendre(PointList p) {
    if (cvx == null) {
      PointList p1 = p ;
      if (p1.next == null)  return false ;
      PointList p2 = p1.next ;
      if (p2.next == null) return false ;
      PointList p3 = p2.next ;
      if (det(p1, p2, p3) < 0) {
        cvx = triangle(p1,p2,p3) ;
      } else {
        cvx = triangle(p1,p3,p2) ;
      }
      return true ;
    } else {
      PointList first = null , last = null ;
      PointList q ;
      int prev = NONE ;
      int edge ;

      // Recherche des transitions entre ombre et lumière
      q = cvx ;
      do {
        if (det(p, q, q.next) >= 0) {
          edge = LIGHT ;
          if (prev == SHADOW) first = q ;
        } else {
          edge = SHADOW ;
          if (prev == LIGHT) last = q ;
        }
        prev = edge ;
        q = q.next ;
      } while (q != cvx) ;
      // Examen de l'arc issu du premier point
      if (det(p, q, q.next) >= 0) {
        if (prev == SHADOW) first = q ;
      } else {
        if (prev == LIGHT) last = q ;
      }

      // Le point ajouté est à l'intérieur
      if (last == null) return false ;

      // Suppression des sommets éclairés, et ajout du nouveau point
      cvx = new PointList(p.x, p.y, last) ;
      first.next = cvx ;
      return true ;
```

```
    }
  }
}
```

6. L'extension de l'enveloppe convexe est en fait correctement traitée. Un côté est considéré éclairé par un point aligné. Si le point P est situé sur le côte strico-sensu, l'effet sera d'ajouter un point inutile à l'enveloppe convexe. Sinon un point devenu inutile sera enlevé.

   En revanche la collecte des trois premiers points doit être changée pour devenir celle des trois premiers points non-alignés.