

TP 9 : Géométrie plane

On va s'intéresser à quelques problèmes géométriques du plan. On représentera un point ou un vecteur par une liste de deux nombres : ses coordonnées dans la base canonique de \mathbb{R}^2 .

Exercice 1. Opérations de base

1. Écrivez une procédure **Scalaire** qui prend en argument deux vecteurs u et v et qui renvoie le produit scalaire de ces vecteurs.
2. Écrivez une procédure **Distance** qui prend en argument deux points p_1 et p_2 et qui renvoie la distance entre ces points.
3. Écrivez une procédure **Mixte** qui prend en argument deux vecteurs u et v et qui renvoie le produit mixte de ces vecteurs.

Exercice 2. Calculs géométriques

1. Écrivez une procédure **Cote** qui prend en argument un point p , un point x_0 et un vecteur u et qui renvoie 'Droite', 'Gauche' ou 'Aligné' suivant le côté où se trouve p par rapport à la droite contenant x_0 et dirigée par u .
2. Écrivez une procédure **PlusProche** qui prend en argument un point p et une liste de points L et qui renvoie le point de L le plus proche de p .
3. Écrivez une procédure **DistanceDroite** qui prend en argument un point p , un point x_0 et un vecteur u et qui renvoie la distance du point p à la droite contenant x_0 et dirigée par u .
4. Écrivez une procédure **AireTriangle** qui prend en argument trois points A, B, C et qui renvoie l'aire du triangle ABC .
5. Écrivez une procédure **DansTriangle** qui prend en argument quatre points A, B, C, p et qui renvoie **true** ou **false** suivant que p se situe à l'intérieur ou à l'extérieur du triangle ABC .
6. Écrivez une procédure **Intersection** qui prend en argument deux points x, y et deux vecteurs u, v et qui renvoie le point d'intersection des droites contenant x (resp. y) et dirigée par u (resp. v) si ce point existe et est unique, et renvoie **false** sinon.

Exercice 3. Algorithmes sur les polygones

1. Par définition, un polygone est *convexe* si tout segment reliant deux sommets est inclus dans le polygone. Écrivez une procédure **EstConvexe** qui prend en argument une liste de points L et qui renvoie **true** ou **false** suivant si le polygone formé par les points dans L est convexe.
2. Écrivez une procédure **AirePolygoneConvexe** qui prend en argument une liste de points L formant un polygone convexe, et qui renvoie la valeur de l'aire du polygone.
Indication : découper le polygone en triangles.

3. Écrivez une procédure **DansPolygoneConvexe** qui prend en argument un point **p** et une liste de points **L** formant un polygone convexe, et qui renvoie **true** ou **false** suivant que **p** se situe à l'intérieur ou à l'extérieur du polygone.
Indication : regarder de quel côté de chaque segment du polygone est le point.
4. Écrivez une procédure **DansPolygone** qui prend en argument un point **p** et une liste de points **L** formant un polygone non croisé, et qui renvoie **true** ou **false** suivant que **p** se situe à l'intérieur ou à l'extérieur du polygone.
Indication : tracer une demi-droite partant du point, et compter le nombre de croisements des côtés du polygone avec la demi-droite.
5. Écrivez une procédure **EnveloppeSuperieure** qui prend en argument une liste de points **L** ordonnés par abscisse croissante et qui renvoie la liste des points de l'enveloppe supérieure, c'est-à-dire le sous-ensemble des points qui forme une ligne brisée concave telle que tous les points de la liste initiale se trouvent en-dessous de cette ligne.
6. Résolvez la question précédente avec au plus $O(n)$ opérations, où n est le nombre de points dans **L**.
7. Écrivez une procédure **EnveloppeConvexe** qui prend en argument une liste de points **L** et qui renvoie la liste des points de l'enveloppe convexe, c'est-à-dire le polygone convexe minimal contenant tous ces points.
Indication : étant donné un polygone convexe, comment peut-on lui ajouter un point et calculer l'enveloppe convexe résultante?
8. Résolvez la question précédente avec au plus $O(n \log n)$ opérations, où n est le nombre de points dans **L**.
Indication : trie les points par abscisse croissante (la fonction de tri Maple **sort** utilise $O(n \log n)$ opérations) puis appliquez une méthode semblable à celle de la question 6.
9. Écrivez une procédure **AirePolygone** qui prend en argument une liste de points **L** formant un polygone non croisé, et qui renvoie la valeur de l'aire du polygone.
Indication : exprimer l'aire du polygone en une somme et différence d'aires de triangles.