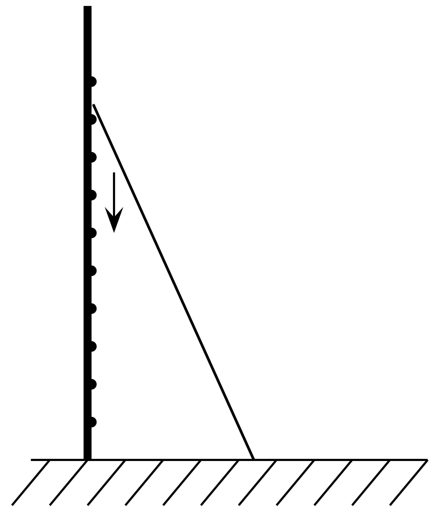
Algorithmique et programmation
Constructions géométriques

Glissement le long d'une échelle Ue poutre de 10m de haut est appuyée contre une échelle verticale. Elle est initialement verticale, puis glisse sur le sol et tombe d'échelon en échelon jusqu'à se retrouver complètement à l'horizontal sur le sol.



On note A et B les points aux extrémités de la poutre.

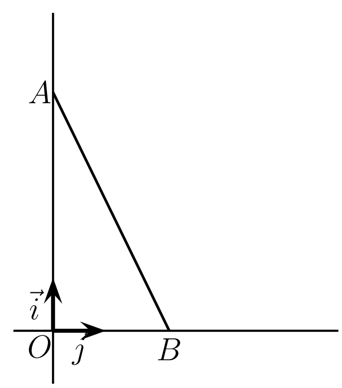
1. Donner les coordonnées de A et de B lorsque la poutre est verticale.
2. Donner les coordonnées de A et de B lorsque la poutre a glissé d'un échelon.
3. Donner les coordonnées de B lorsque l'ordonnée de A est y.
4. Écrire un algorithme et un programme qui trace toutes les positions successives de la poutre, échelon après échelon, depuis sa position verticale jusqu'à sa position finale horizontale.

   Avec les calculatrices TI, dans le menu `dessin`, la commande `Ligne`(x_A, y_A, x_B, y_B) permet de tracer le segment $[AB]$

   En Python, avec `Libxy`, les commandes $A = (x_A, y_A)$, $B = (x_B, y_B)$ puis `Ligne(A,B)` permettent de tracer le segment $[AB]$.
5. Donner, en fonction des coordonnées de B, celles de B' symétrique de B par rapport à l'axe des ordonnées. Tracer alors aussi les segments $[AB']$.
6. Donner, en fonction des coordonnées de A, celles de A' symétrique de A par rapport à l'axe des ascisses. Tracer alors aussi les segments $[A'B]$ et $[A'B']$.

   On obtient une **astroïde**.

   On peut obtenir une figure plus détaillée en utilisant des échelons écartés de 50 cm seulement (au lieu de 1 m), ou encore de 10 cm...



Triangle de Sierpiński - Construction géométrique *a priori* aléatoire.

On considère dans le plan repéré, le triangle ABC dont les sommets sont les points $A(0;0)$, $B(0;1)$ et $C(1;0)$.

Soit de plus le point $M(0,5;0,5)$.

On va construire un ensemble de points dans le triangle ABC de la façon suivante.

1. On désigne au hasard un des trois sommets A, B ou C.
2. On construit et affiche le milieu du point M précédent et de ce sommet.
3. Le point calculé précédemment devient le point M, et on recommence à l'étape 1.

Construire 100 points suivant cette méthode *(puis 200, 500, 1000...)*.
Que peut-on dire de l'effet de *l'aléatoire* dans cette construction.

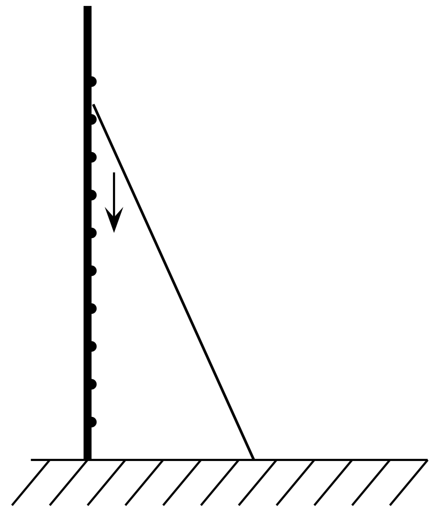
Indications de programmation

TI : Dans le menu `math`, puis probabilités (`PROB`), la fonction `nbrAléatEnt(1,3)` permet de tirer un nombre entier au hasard entre 1 et 3.

Python Il faut charger la fonction `randint` depuis une bibliothèque de fonctions mathématiques : `from random import randint` , en tout début de programme ; ensuite la commande `randint(1,3)` permet de tirer un nombre entier entre 1 et 3 au hasard.

Algorithmique et programmation
Constructions géométriques

Glissement le long d'une échelle Ue poutre de 10m de haut est appuyée contre une échelle verticale. Elle est initialement verticale, puis glisse sur le sol et tombe d'échelon en échelon jusqu'à se retrouver complètement à l'horizontal sur le sol.



On note A et B les points aux extrémités de la poutre.

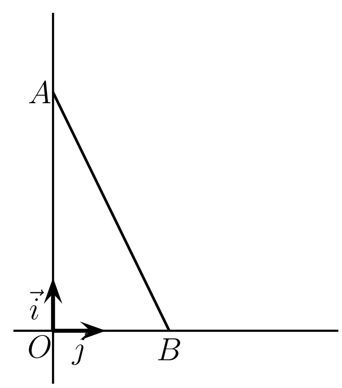
1. Donner les coordonnées de A et de B lorsque la poutre est verticale.
2. Donner les coordonnées de A et de B lorsque la poutre a glissé d'un échelon.
3. Donner les coordonnées de B lorsque l'ordonnée de A est y.
4. Écrire un algorithme et un programme qui trace toutes les positions successives de la poutre, échelon après échelon, depuis sa position verticale jusqu'à sa position finale horizontale.

   Avec les calculatrices TI, dans le menu `dessin`, la commande `Ligne`(x_A, y_A, x_B, y_B) permet de tracer le segment $[AB]$

   En Python, avec `Libxy`, les commandes $A = (x_A, y_A)$, $B = (x_B, y_B)$ puis `Ligne(A,B)` permettent de tracer le segment $[AB]$.
5. Donner, en fonction des coordonnées de B, celles de B' symétrique de B par rapport à l'axe des ordonnées. Tracer alors aussi les segments $[AB']$.
6. Donner, en fonction des coordonnées de A, celles de A' symétrique de A par rapport à l'axe des ascisses. Tracer alors aussi les segments $[A'B]$ et $[A'B']$.

   On obtient une **astroïde**.

   On peut obtenir une figure plus détaillée en utilisant des échelons écartés de 50 cm seulement (au lieu de 1 m), ou encore de 10 cm...



Triangle de Sierpiński - Construction géométrique *a priori* aléatoire.

On considère dans le plan repéré, le triangle ABC dont les sommets sont les points $A(0;0)$, $B(0;1)$ et $C(1;0)$.

Soit de plus le point $M(0,5;0,5)$.

On va construire un ensemble de points dans le triangle ABC de la façon suivante.

1. On désigne au hasard un des trois sommets A, B ou C.
2. On construit et affiche le milieu du point M précédent et de ce sommet.
3. Le point calculé précédemment devient le point M, et on recommence à l'étape 1.

Construire 100 points suivant cette méthode *(puis 200, 500, 1000...)*.
Que peut-on dire de l'effet de *l'aléatoire* dans cette construction.

Indications de programmation

TI : Dans le menu `math`, puis probabilités (`PROB`), la fonction `nbrAléatEnt(1,3)` permet de tirer un nombre entier au hasard entre 1 et 3.

Python Il faut charger la fonction `randint` depuis une bibliothèque de fonctions mathématiques : `from random import randint` , en tout début de programme ; ensuite la commande `randint(1,3)` permet de tirer un nombre entier entre 1 et 3 au hasard.