

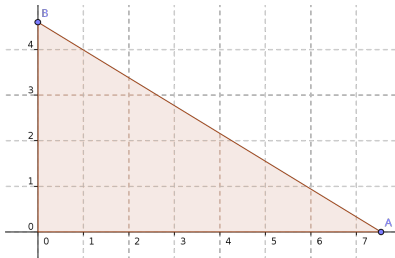
Complexité : première approche.

Académie de Lyon

Dénombrement de couples d'entiers naturels

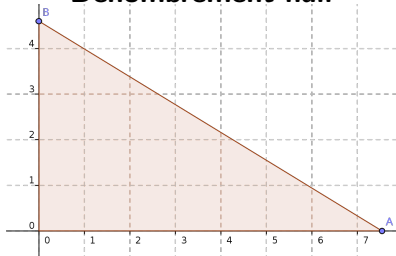
On considère la figure suivante, dans le plan repéré :

- A est un point situé sur $]Ox)$, on note a l'abscisse de A
- B est un point situé sur $]Oy)$, on note b l'abscisse de B



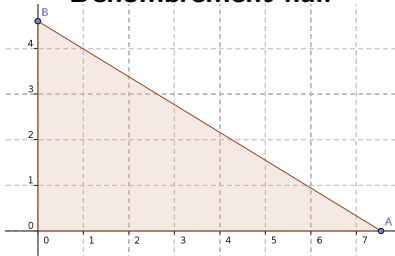
On souhaite dénombrer les points à coordonnées entières qui appartiennent au domaine grisé, frontières comprises.

Dénombrement naïf



- Déterminer une équation réduite de la droite (AB)
- Soit $M(x; y)$ un point du plan. Quelles conditions doivent vérifier x et y pour que M appartienne à la zone grisée ?
- Écrire un algorithme qui teste les points du rectangle de côtés $[OA]$ et $[OB]$ et qui renvoie le nombre de ceux qui appartiennent au domaine grisé.

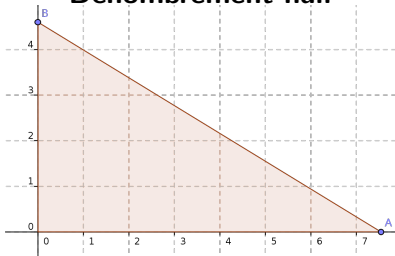
Dénombrement naïf



- $(AB) : y = \frac{-b}{a}x + b$

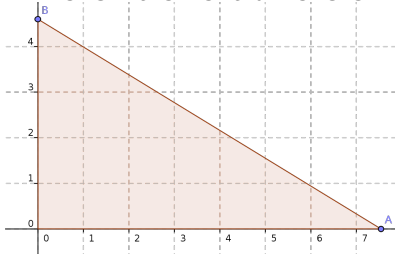
- $M(x; y)$ appartient à la zone grisée lorsque
$$\begin{cases} x \geq 0 \\ y \geq 0 \\ y \leq \frac{-b}{a}x + b \end{cases}$$

Dénombrement naïf



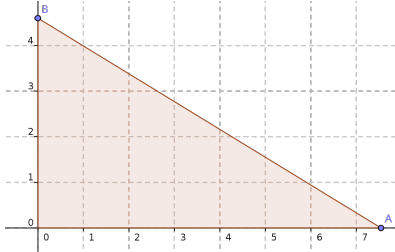
```
def nbpoints(a,b):  
    compteur=0  
    for i in range (0,int(a)+1):  
        for j in range (0,int(b)+1):  
            if j<=-(b*i)/a+b :  
                compteur=compteur+1  
    return compteur
```

Dénombrement amélioré



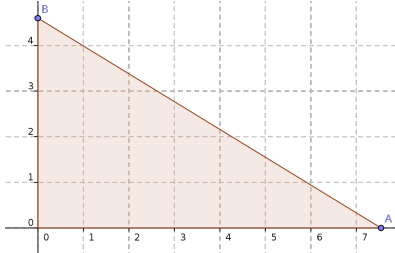
- Combien de points font toujours partie du domaine grisé ?
- Donner les coordonnées d'un point pour lequel la vérification est inutile
- Modifier l'algorithme précédent pour réduire le nombre de boucles.

Dénombrement amélioré



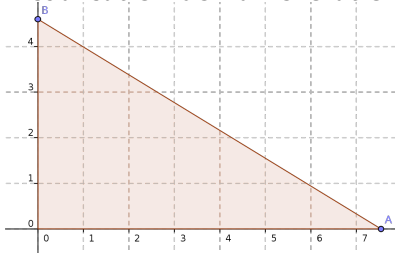
- Combien de points font toujours partie du domaine grisé : $E(a) + E(b) + 1$
- Donner les coordonnées d'un point pour lequel la vérification est inutile : $(a; b)$
- Modifier l'algorithme précédent pour réduire le nombre de boucles.

Dénombrement amélioré



```
def nbpoints_optim(a,b):  
    compteur = 1 + int(a)+int(b)  
    for i in range (1,int(a)+1):  
        for j in range (1,int(-(b*i)/a + b)+1):  
            compteur=compteur+1  
    return compteur
```


Visualisation de l'amélioration



- Utiliser la fonction `time.time` pour évaluer le temps de calcul.
- Écrire un algorithme qui compare et affiche les temps pour les deux méthodes de dénombrement précédentes. avec $b = 1000$ et a variant de 0 à 1000 avec un pas de 25

► lancer