Complexité : un exemple pour le lycée

G. Aldon - J. Germoni - J.-M. Mény

IREM de Lyon

Mars 2012

Un problème sur les triangles

Écrire un algorithme répondant à la demande suivante :
Entrée : un entier naturel $p > 0$.
Sortie : les triangles à côtés entiers, rectangles, de périmètre p.

Un problème sur les triangles

Écrire un algorithme répondant à la demande suivante :
Entrée : un entier naturel $p > 0$.
Sortie : les triangles à côtés entiers, rectangles, de périmètre p.

Première version :

Pour a de 1 à p :
    pour b de 1 à p :
        pour c de 1 à p :
            tester le triplet (a,b,c), le stocker si satisfaisant.

Programme xcas

Xcas

```
triangle(p) :={
local a,b,c,L ;
L :=NULL ;
pour a de 1 jusque p faire
pour b de 1 jusque p faire
pour c de 1 jusque p faire
si a+b+c==p et (a^2+b^2==c^2 ou a^2+c^2==b^2 ou
b^2+c^2==a^2)
alors L :=L,[a,b,c] ;fsi ;
fpour ; fpour ; fpour ;
retourne L ;} : ;
```

Programme python

Python

```
def triangle (p) :
    L=[ ]
    for a in range(1,p+1) :
        for b in range(1,p+1) :
            for c in range(1,p+1) :
                if (a+b+c==p) and (a**2+b**2+c**2 == 2*max([a**2,b**2,c**2])) :
                    L.append([a,b,c])
    return L
```

Première approche expérimentale

Sur une calculatrice Ti82, plus de 5 secondes pour un périmètre $p = 10$.

Première approche expérimentale

Sur une calculatrice Ti82, plus de 5 secondes pour un périmètre $p = 10$.

Hypothèse de proportionnalité temps – nombre de boucles.
Quel temps pour un périmètre $p = 1000$?

$$\frac{5 \times 100^3}{3600 \times 24} \approx 58 \text{ jours}$$

Amélioration de l'algorithme

On cherche a, b, c tels que $a \leqslant b \leqslant c$:
Pour a de 1 à $p/3$:
pour b de a à $p/2$:
si $p - a - b \geqslant b$ alors tester le triplet (a,b,p-a-b) et le stocker si satisfaisant

ou même :

Pour a de 1 à $p/3$:
pour b de a à $(p - a)/2$:
si $p - a - b \geqslant b$ alors tester le triplet (a,b,p-a-b) et le stocker si satisfaisant

Programme xcas

Xcas

```
triangle(p) :={
local a,b,c,L;
L :=NULL;
pour a de 1 jusque p/3 faire
pour b de a jusque (p-a)/2 faire
c :=p-a-b;
si a^2+b^2==c^2 alors L :=L,[a,b,c];fsi;
fpour; fpour;
retourne L;} :;
```

Programme python

Python

```
def triangle(p) :
    L=[ ]
    for a in range(1,floor(p/3)) :
        for b in range(a,floor((p-a)/2)) :
            c=p-a-b
            if (a**2+b**2==c**2) : L.append([a,b,c])
    return L
```

Complexité : les triangles entiers

Comparer les temps de calcul expérimentalement et expliquer.

FICHIER SAGE

Complexité : les triangles entiers

Évaluation de la complexité

Première version :

Complexité : les triangles entiers

Évaluation de la complexité

Première version :
Nombre de IF : p^3

Complexité : les triangles entiers

Évaluation de la complexité

Première version :
Nombre de IF : p^3

Seconde version :

Complexité : les triangles entiers

Évaluation de la complexité

Première version :
Nombre de IF : p^3

Seconde version :
Nombre de IF :

$$\sum_{a=1}^{\lfloor p/3 \rfloor} \left(\frac{p}{2} - a + 1 \right) \leqslant \frac{1}{9} p(p+3)$$

ou même

$$\sum_{a=1}^{\lfloor p/3 \rfloor} \left(\frac{p-a}{2} - a + 1 \right) \leqslant \frac{1}{12} p(p+1)$$