

Lab Complexité n°1

Un exemple frappant

Soit la fonction suivante :

```
def fonction1(n):  
    t = []  
    for i in range(n+1):  
        for j in range(n+1):  
            if i**2 + j**2 == n:  
                t.append((i,j))  
    return
```

1. Quel est le but de cette fonction ?
2. En fait l'hypothèse qu'une addition consomme 1 unité de temps et qu'une multiplication en consomme 2, calculez le temps d'exécution en fonction de n.
3. Voici deux autres implémentations différentes de la fonction :

```
def fonction2(n):  
    t = []  
    for i in range(n+1):  
        for j in range(0,i+1) :  
            if i**2 + j**2 == n:  
                t.append((i,j))  
    return t
```

```
def fonction3(n):  
    t = []  
    N = math.floor(math.sqrt(n))  
    for i in range(N+1):  
        for j in range(0,i+1) :  
            if i**2 + j**2 == n:  
                t.append((i,j))  
    return t
```

1. **Verifiez bien que si un couple (i,j) est retourné par la fonction 1 alors il est retournée par les deux autres fonctions.**
2. **En conservant l'hypothèse faite à la question 2, calculez les temps d'exécution respectifs de ces deux nouvelles fonctions.**
4. **Faites l'implémentation de ces trois fonctions sur python et observer le temps de calcul pour différentes valeurs de n (5 valeurs différentes). Calculez les rapports de temps d'exécutions entre la fonction 1 , la fonction 2 et entre la fonction 2, la fonction 3 pour chaque n.**
5. **Bonus : Est ce qu'il y a une 4ieme manière d'écrire cette fonction qui pourrait nous faire gagner plus de temps ?**