

Python : notion de complexité (mini projet):

On s'intéresse aux 2 fonctions suivantes:

```
0 def puissance_V1(x,n):
  p = 1
  2   for i in range(1,n+1):
      p = p * x
  4   return p
```

puissance.py

```
0 def puissance_V2(x,n):
  p = 1
  2   while n > 0:
      if n % 2 == 1:
  4         p = p * x
      x = x * x
  6         n = n // 2
  return p
```

puissance.py

1. Expliquer ce que permet de calculer la fonction `puissance_V1(x,n)`. On admettra que que la fonction `puissance_V2(x,n)` permet de faire le même calcul.
2. Calculer 2^5 avec ces 2 fonctions. Calculer ensuite 2^{1000} avec ces 2 fonctions. Que remarque t-on?
3. On souhaite maintenant faire une représentation graphique du temps de calcul de ces 2 fonctions pour différentes valeur de n .

3.1 A quoi sert la fonction suivante?

```
0 def complexite_V1(n):
  t1 = time.perf_counter()
  2   puissance_V1(2,n)
  t2 = time.perf_counter()
  4   duree = t2 - t1
  return duree
```

puissance.py

On n'oubliera pas d'importer la bibliothèque:

```
0 import time
```

puissance.py

3.2 Ecrire une fonction analogue pour `puissance_V2(2,n)`. On l'appellera `complexite_V2(n)`

3.3 Que permet de faire la suite du programme?

```
0 nombre = 100
2 abcisse = []
3 for i in range(1, nombre):
4     abcisse.append(i)
6 temps_V1 = []
7 for i in range(1, nombre):
8     temps_V1.append(complexite_V1(i))
```

puissance.py

3.4 Ecrire une fonction analogue pour puissance_V2(2,n). On l'appellera temps_V2(n)

3.5 Terminer le programme ainsi:

```
0 plt.plot(abcisse, temps_V1, linestyle = 'none', marker = 'o', c = 'black',
1          markersize = 1, label = 'fonction V1')
2 plt.plot(abcisse, temps_V2, linestyle = 'none', marker = 'o', c = 'blue',
3          markersize = 1, label = 'fonction V2')
4 plt.legend()
5 plt.xlabel("puissance")
6 plt.ylabel("temps")
7 plt.title("Complexite experimentale des fonctions puissance")
8 plt.savefig('complexite_puissances.png', dpi = 200)
9 plt.show()
```

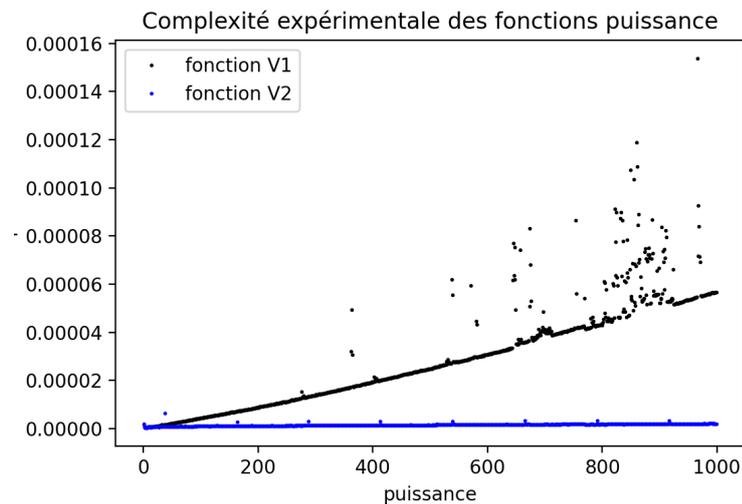
puissance.py

On n'oubliera pas d'importer la bibliothèque:

```
0 import matplotlib.pyplot as plt
```

puissance.py

4. On doit obtenir un graphique du type:



4.1 Analyser ce graphique. Quelle fonction semble la plus performante?

4.2 Que dire des points qui ne sont pas sur la courbe.

4.3 Conclure.