

DS03

Algorithmique et programmation

Sources :

Proposition de corrigé

Exercice 1 : Tableaux

- Ce devoir est à faire de façon individuelle.
- Utilisez Python, version 3.
- Créez un répertoire pour faire ce DS.
- Recopiez le fichier `sujet.py` dans ce répertoire, ainsi que le fichier `zeta5.txt`. Vous aurez besoin de ces fichiers pour le DS mais vous ne devrez pas les modifier (sous peine de risquer d'avoir tout faux).
- Avec IDLE créez un nouveau fichier python que vous sauverez dans ce répertoire. C'est ce fichier que vous allez utiliser pour répondre aux questions du DS. Dans la suite, on supposera que vous l'avez appelé `ds.py`. Dans `ds.py`, faites un `import sujet` pour pouvoir utiliser les fonctions fournies dans `sujet.py`.
- On ne vous demande pas de rendre votre programme mais seulement de répondre aux questions posées, dont les réponses sont toutes numériques, **sur le formulaire de réponse joint**.
- **Attention : toutes les questions posées dépendent d'un paramètre α , qui vous est donné sur le formulaire de réponse. La valeur de α est différente pour chacun d'entre vous. Soyez attentif à sa valeur : s'il est faux toutes vos réponses seront fausses.** Dans toute la suite, on considère que la variable Python `alpha` contient la valeur de votre paramètre α . Il vous est donc conseillé, au début de `ds.py`, de faire un `alpha = α` .
- Lorsque la réponse demandée est un réel, on attend que l'écart entre la réponse que vous donnez et la vraie valeur soit strictement inférieur à 10^{-4} . Donnez donc des valeurs avec 5 chiffres après la virgule.
- Vous avez le droit d'utiliser toutes les fonctions prédéfinies de Python que vous voulez, en particulier celles définies dans les modules `scipy.optimize` et `scipy.integrate`.

1 Analyse de tableaux

1.1 Première partie

Exécutez `t = sujet.cree_tableau(alpha)`**Q 1 : Quelle est la longueur du tableau `t` ? Par la suite, on la note N .****Q 2 : Combien d'éléments de `t` sont supérieurs ou égaux à 3000 ?****Q 3 : Combien d'éléments de ce tableau sont divisibles par 3 ?****Q 4 : Quel est le nombre de couples (i, j) tels que $0 \leq i < j < N$ et `t[i] < t[j]` ?**On définit la suite u par

$$\forall n \in \mathbb{N} \quad u_{n+1} = r(15091 u_n, 64007)$$

$$\text{et } u_0 = 10 + \alpha$$

où $r(a, d)$ désigne le reste de la division euclidienne de a par d (en python, on utilise l'opérateur `%` pour calculer ce reste).

Construire un tableau `U` de taille 10^4 tel que pour tout $i \in \llbracket 0, 10^4 - 1 \rrbracket$, `U[i]` contienne u_i .

Q 5 : Que vaut u_{42} ? Q 6 : Que vaut le dernier élément du tableau `U` ? Q 7 : Combien d'éléments du tableau `U` sont-ils divisibles par 17 ? Q 8 : Quel est le nombre d'indices $i \in \llbracket 0, 10^4 - 2 \rrbracket$ tels que $|u_i - u_{i+1}| \leq 1000$? Q 9 : Quelle est la somme des valeurs de ce tableau ?

1.2 Autour de $\zeta(5)$

Dans cette partie, on utilise le fichier `zeta5.txt`. Celui-ci contient un million de décimales (après la virgule) de $\zeta(5) = \sum_{n=1}^{+\infty} \frac{1}{n^5} \approx 1,0369$.

Voici les premières lignes du fichier (à gauche, on a noté en petit les numéros des lignes pour pouvoir en parler ; attention : **ils n'apparaissent pas dans le fichier**) :

```

1 zeta5_huvent with 1000000 digits :
2 zeta5_huvent = 1.
3 0369277551 4336992633 1365486457 0341680570 8091950191 : 50
4 2811974192 6779038035 8978628148 4560043106 5571333363 : 100
5 7962034146 6556609042 8009617791 5597084183 5110721800 : 150
6 8764486628 6337180353 5983639623 6512888898 1335276775 : 200
7 2398275032 0224368457 6644466595 8115993917 9777450392 : 250
8 4464391966 6615966401 6205325205 0215192267 1351256785 : 300
9 9748692860 1974479843 2006726812 9753091990 0774656558 : 350
10 6015265737 3003756153 2683149897 9719350398 3785813199 : 400
11 2288488642 5335104251 6025108499 0434640294 1172432757 : 450
12 6341508162 3322456186 4992714427 2264614113 0075808683 : 500
13
14 1691649791 8137769672 5145590158 0353093836 2260020230 : 550
15 4558560981 5265536062 6530883832 6130378691 7412255256 : 600
16 0507375081 3917876046 9541867836 6657122379 6259477937 : 650
17 8931344280 5560465115 0585291073 6964334642 8934143397 : 700
18 5231743713 3962434331 1485731093 6262213535 7253048207 : 750

```

À partir de la ligne 3, les décimales sont rangées par paquets de 10, eux-mêmes rangés par lignes de 5 paquets, elles-mêmes rangées dans des blocs de 10 lignes (qui contiennent donc 500 décimales ; il y a ainsi 2000 blocs). Attention : il y a des lignes vides (la ligne 13 sur cet extrait puis de manière générale, tous les paquets de 500 décimales). On va faire des statistiques sur le nombre d'apparitions de $2000 + \alpha$ dans l'écriture décimale de $\zeta(5)$. Par exemple, 92 apparaît deux fois dans les 20 premières décimales. Plus subtil : 70 apparaît deux fois dans les 40 premières décimales, dont une fois tronqué par un espace. On peut même trouver des occurrences coincées entre deux lignes (1004, en position 6199), voire entre deux blocs (1039, en position 15498).

Au besoin, vous pouvez ouvrir ce fichier avec n'importe quel éditeur de texte.

Q 10 : Donner le nombre d'occurrences de $2000 + \alpha$ dans les paquets de 10 décimales.

On pourra appliquer un algorithme du type : « Pour chaque ligne l , on casse / selon les espaces. S'il y a le bon nombre de blocs, alors pour chacun des blocs b , et pour chaque i entre 0 et 6 (inclus), on regarde si `b[i : i + 4]` vaut la chaîne représentant $2000 + \alpha$ ».

Q 11 : Donner le nombre d'occurrences de $2000 + \alpha$ dans les paquets de 50 décimales obtenus en concaténant, pour chaque ligne, les 5 paquets de 10 décimales.

Q 12 : Donner le nombre d'occurrences de $2000 + \alpha$ dans les paquets de 500 décimales obtenus en concaténant, pour chaque bloc de 10 lignes, les 50 paquets de 10 décimales.

Q 13 : Donner le nombre d'occurrences de $2000 + \alpha$ dans les 10^6 premières décimales de $\zeta(5)$.