
Défis 1 : Primalité

Tous les programmes sont à rédiger en Python ou Sage, et à écrire sous forme de fonctions, dans un même fichier. Vous êtes encouragés à utiliser un notebook (Jupyter).

Ces problèmes admettent plusieurs solutions, plus ou moins astucieuses. Les applications proposées servent entre autres à vérifier que votre algorithme n'est pas trop grossier : si vous avez implémenté un algorithme raisonnable, les réponses doivent être presque instantanées.

N'hésitez pas à vous renseigner pour résoudre des problèmes !

Les exercices marqués d'une étoile sont pour ceux qui veulent aller plus loin.

Exercice 1. Écrivez une fonction `test_premier` qui prend en argument un entier $n \geq 1$, et ressort `True` si n est premier, et `False` sinon.

Application : L'entier 2 147 483 647 est-il premier ?

Exercice 2. Écrivez une fonction `crible_Eratosthene` qui prend en argument un entier $n \geq 1$, et ressort la liste de tous les nombres premiers inférieurs ou égaux à n .

Application : Combien y a-t-il de nombres premiers inférieurs ou égaux à 100 000 ?

Exercice 3. Un test de primalité naïf d'un entier $n \neq 2$ consiste à vérifier si $2^{n-1} \equiv 1 [n]$. Tous les nombres premiers (sauf 2) vérifient ce critère, tandis que peu de nombres composés le vérifient. En vous aidant de la fonction `pow`, écrivez un programme `test_Fermat_naif` qui prend en argument un entier $n \geq 1$, et ressort `True` si n passe ce test, et `False` sinon.

Application : Comparez la liste des nombres premiers inférieurs ou égaux à 1 000, et la liste des entiers inférieurs ou égaux à 1 000 passant `test_Fermat_naif`. Renseignez-vous sur les quelques différences entre ces deux listes.

Quel est l'avantage de `test_Fermat_naif` par rapport à `test_premier` ?

Exercice 4. * Étant donné un nombre premier p et un entier n , la **valuation** p -adique de n est le plus grand entier $k \geq 0$ tel que p^k divise n . Écrivez une fonction `valuation` qui prend en argument deux entiers $p, n \geq 1$, et ressort la valuation p -adique de n (il est inutile de tester si p est un nombre premier).

Application : Écrivez 89 952 499 728 comme produit d'une puissance de 3 et d'un nombre premier avec 3.

Exercice 5. ** Modifiez le crible d'Eratosthène pour obtenir une fonction `crible_facteurs_premiers` qui prend en argument un entier $n \geq 1$, et ressort la liste de tous les facteurs premiers de tous les entiers inférieurs ou égaux à n .