

TP1 2e année IPT : Révisions

Exercice 1 : Test de primalité

1. Écrire une fonction `is_prime(n)` qui prend en argument un entier n et teste sa primalité.
2. Écrire une fonction `next_prime(n)` qui renvoie le plus petit nombre premier strictement supérieur à n .
3. Écrire un script construisant une liste `prime` contenant les 1000 plus petits nombres premiers. Peut-on l'améliorer ?

Exercice 2 : Factorisation

1. Écrire une fonction `factorisation(n)` qui décompose un entier n en produit de facteurs premiers avec multiplicités. Par exemple, `facto(12)` renvoie la liste $[(2,2), (3,1)]$.
2. En déduire une fonction `nb_diviseurs(n)` qui renvoie le nombre de diviseurs d'un entier n , et une fonction `sum_diviseurs(n)` qui renvoie la somme des diviseurs de n .
(On pourra d'abord établir sur papier les formules donnant ces résultats si l'on connaît la décomposition de n en facteurs premiers.)
3. Écrire une fonction `phi(n)` qui calcule l'indicatrice d'Euler $\varphi(n)$.

Rappel : La fonction indicatrice d'Euler $\varphi(n)$ représente le nombre d'entiers entre 1 et n qui sont premiers avec n . Par exemple $\phi(12) = 4$ car parmi les entiers entre 1 et 12, seuls 1, 5, 7 et 11 sont premiers avec 12. La fonction $\varphi(n)$ est multiplicative: si m et n sont premiers entre eux, $\varphi(mn) = \varphi(n)\varphi(m)$.

Questions préliminaires : si p est premier, que vaut $\varphi(p)$? Et si $r > 0$, que vaut $\varphi(p^r)$? En déduire une formule donnant $\varphi(n)$ si l'on connaît la décomposition en facteurs premiers de n .

Exercice 3 : Recherche dichotomique

On suppose que `t` est une liste triée de réels distincts et que x est un réel vérifiant $t[0] \leq x < t[n-1]$ où n est la taille de la liste.

1. Écrire une fonction `dicho(x,t)` retournant l'unique entier i vérifiant $t[i] \leq x < t[i+1]$ en faisant une recherche dichotomique.
2. Donner la complexité de cet algorithme.