



Listes de nombres premiers Crible d'Ératosthène

Fiche Élève

TS
Spé
Math

Le module *scilab pour les lycées* comporte une commande `liste_premiers(n)` qui retourne la suite des nombres premiers inférieurs à n .

On se propose d'explicitier un programme qui fournit le même résultat.

Il faut s'attendre à ce qu'en terme de temps de calcul, ce programme soit moins performant que la commande directement implémentée dans le logiciel. En revanche, en terme d'algorithmique, on aura la satisfaction de comprendre "comment ça marche".

La méthode que l'on va employer, très efficace, remonte à l'Antiquité. Elle est connue sous le nom de "crible d'Ératosthène". Voici un aperçu de la démonstration : étant donné la liste A des entiers de 2 à n ,

premier passage : on considère le premier élément de A , soit 2, et on supprime de la liste A tous ses multiples sauf lui-même. Ainsi, les éléments 4, 6, 8, 10, 12, \dots , c'est-à-dire tous les nombres pairs de la liste A sauf 2 sont supprimés. La liste A commence donc maintenant par 2, 3, 5, 7, 11, \dots

deuxième passage : on considère alors le deuxième élément de la liste A , soit 3 et on supprime de A tous ses multiples sauf lui-même. On remarque que 6 est déjà parti, que 9 sera supprimé, \dots , etc. La liste A deviendra alors 2, 3, 5, 7, 11, 13, 17, 19, 23, 25, \dots

passages suivants : on itère le procédé avec le troisième puis le quatrième élément de A , \dots , etc.

On aura compris que ce procédé a pour but d'éliminer les nombres composés de la liste A de sorte qu'il ne reste que les nombres premiers.

Quand faut-il s'arrêter? Au $k^{\text{ième}}$ passage, on doit supprimer de la liste A les entiers $2 \cdot A(k)$, $3 \cdot A(k)$, \dots mais en fait $A(k)^2$ est le premier de ces nombres qui n'a pas déjà été supprimé ; de plus, on ne le supprime effectivement que s'il est inférieur ou égal au dernier terme de la liste A . Sinon, cette liste reste inchangée et le crible d'Ératosthène s'arrête.

Il est facile de compléter ces indications en une démonstration.

Commandes scilab utiles pour écrire le programme demandé

• `A=2:n` définit la liste des entiers de 2 à n ($n \geq 2$). A variera ensuite au cours du déroulement de l'algorithme.

Pour toute liste L de nombres entiers supérieurs ou égaux à 0, :

- `L(k)` désigne le $k^{\text{ième}}$ élément de L , si $1 \leq k \leq \text{taille}(L)$ (`taille(L)` est la longueur de la liste L).
- `modulo(L,p)` retourne la liste des restes des divisions des éléments de L par le nombre entier non nul p . Si ce reste vaut 0, l'élément correspondant est un multiple de p .
- `find(modulo(L,p) == 0)` retourne les rangs des éléments de L qui sont des multiples de p .
- Si $1 \leq k \leq \text{taille}(L)$, `L(k) = []` supprime le terme de rang k .
- Si I est une liste d'entiers non nuls, inférieurs ou égaux à `taille(L)`, alors `L(I) = []` supprime les éléments de L dont les rangs appartiennent à I . Par exemple, si $L=[3,7,9,8,5,2]$ et $I=[2,4,5]$, `L(I) = []` retourne `L=[3,9,2]`.
- `L($)` retourne le dernier élément de la liste L .
- `tic()` déclenche le chronomètre. `temps=toc()` ; `afficher(temps)` arrête le chronomètre et affiche le temps écoulé.

Questions :

1. Écrire un programme (une fonction-*scilab*) qui donne la liste des nombres premiers inférieurs ou égaux à n , pour tout entier donné n , supérieur ou égal à 2.
2. Tester ce programme avec $n = 100$.
3. Comparer les temps de calcul du programme trouvé et de la commande *scilab* `liste_preiers(n)` pour $n = 1000\ 000$.
4. Quel est le plus grand nombre premier inférieur à 1000 000 ?
5. Combien y-a-t-il de nombres premiers inférieurs à 1000 000 ?

