http://irem.univ-reunion.fr/spip.php?article742

Petits exerciciels d'arithmétique

- Algorithmique et programmation
- CoffeeScript
- Exercices en CoffeeScript
- Exercices de classement
-

Date de mise en ligne : lundi 26 mai 2014

On peut classer des nombres entiers selon des catégories de nature arithmétique :

- **nombres pairs et impairs**
- **nombres premiers et composés**
- **résidus quadratiques ou non modulo un nombre donné**
- **nombres parfaits, abondants ou déficients**
- **nombres de Fibonacci ou non...**

Quelques exemples en sont donnés ci-dessous.

Pour commencer, un exercice de dénombrement : Il s'agit de mettre un certain nombre d'objets dans un sac opaque :
<a href="http://irem.univ-reunion.fr/IMG/html/compter.html" title='HTML - 20.9 ko' type="text/html">[HTML - 20.9 ko] **exercice de comptage** pour enfants de CP

Cet exercice vise le niveau CP.

Puis un exercice de classement selon la parité :
<a href="http://irem.univ-reunion.fr/IMG/html/pairs_impairs1.html" title='HTML - 2.4 ko' type="text/html">[HTML - 2.4 ko] **classement par parité** les nombres sont donnés sous forme décimale, et calculés aléatoirement

Un autre, où les nombres doivent être calculés avant de les classer :
<a href="http://irem.univ-reunion.fr/IMG/html/pairs_impairs2.html" title='HTML - 2.4 ko' type="text/html">[HTML - 2.4 ko] **classement par parité** cette fois-ci ce sont des expressions qu'il faut classer

Théorie de ces exerciciels

Le fait qu'il est toujours possible (par un algorithme comme regarder si son dernier chiffre est 0, 2, 4, 6 ou 8) de savoir si un nombre est pair, s'exprime en disant que l'ensemble des nombres pairs est récursivement énumérable. Si, de plus, son complémentaire est aussi récursivement énumérable, on dit que l'ensemble des nombres pairs est un ensemble récursif. C'est le cas, parce qu'une caractérisation des ensembles récursivement énumérables est que leur fonction caractéristique est calculable par algorithme. Or

- La fonction caractéristique des nombres pairs est

  1-x%2

- et celle des nombres impairs est

  x%2

Elles sont donc toutes deux calculables puisque chacune d'entre elles correspond au dernier chiffre binaire de x.

Ainsi, le fait que l'ensemble des nombres pairs est récursivement énumérable, signifie qu'il est toujours possible en un temps fini, de remplir correctement le sac de gauche. Le fait qu'il est également possible en un temps fini de remplir correctement le sac de droite, revient à dire que l'ensemble des nombres pairs est récursif. En fait, les deux notions ne sont pas équivalentes, quoique tous les ensembles finis soient récursifs. Moins évident, il en est de même pour les nombres premiers et les nombres de Fibonacci.

Le théorème de Matiiassevitch dit qu'un ensemble d'entiers est récursivement énumérable si et seulement s'il est l'ensemble des solutions d'une équation diophantienne. Trouver une telle équation pour les nombres de Fibonacci ou pour les nombres premiers, est loin d'être facile. Mais pour les nombres pairs ?

Exercice : Trouver un polynôme P(x) ne prenant que des valeurs paires pour tout x entier relatif.

Enfin un exercice où le classement se fait entre nombres premiers et nombres composés :
<a href="http://irem.univ-reunion.fr/IMG/html/premiers_composes.html" title='HTML - 2.7 ko' type="text/html">[HTML - 2.7 ko] **classement par primalité** l'usage d'une calculatrice est conseillé, pour tester certaines divisibilités

Une remarque probabiliste à propos de celui-ci : Il arrive assez fréquemment que l'ensemble des nombres premiers soit plus gros que celui des nombres composés, alors qu'il n'y a que 46 nombres premiers sur les 200 choisis au hasard. Mais en fait, les nombres ne sont choisis au hasard que parmi les nombres impairs de 3 à 201, et du coup la probabilité de tomber sur un nombre premier en choisissant un nombre au hasard est 0,45. Le nombre de nombres premiers dans l'exerciciel suit donc une loi binomiale de paramètres 10 et 0,45. La probabilité que plus de la moitié des nombres de l'exerciciel soit premiers est donc environ 0,2616 qui n'est pas si négligeable. Voici la loi suivie par le nombre de nombres premiers de l'exercice :

<svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg" xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd" xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape" version="1.2" x="0mm" y="0mm" width="601" height="115.84665" id="svg2" inkscape:version="0.47 r22583" sodipodi:docname="binomPrem.svg"> <rdf:RDF> <cc:Work rdf:about=""> <dc:format>image/svg+xml </dc:format> <dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage" /> </cc:Work> </rdf:RDF> <inkscape:perspective sodipodi:type="inkscape:persp3d" inkscape:vp_x="0 : 0.5 : 1" inkscape:vp_y="0 : 1000 : 0" inkscape:vp_z="1 : 0.5 : 1" inkscape:persp3d-origin="0.5 : 0.33333333 : 1" id="perspective56" /> <sodipodi:namedview pagecolor="#ffffff" bordercolor="#666666" borderopacity="1" objecttolerance="10" gridtolerance="10" guidetolerance="10" inkscape:pageopacity="0" inkscape:pageshadow="2" inkscape:window-width="640" inkscape:window-height="483" id="namedview50" showgrid="false" inkscape:zoom="0.22425739" inkscape:cx="352.54724" inkscape:cy="-66.181091" inkscape:window-x="0" inkscape:window-y="25" inkscape:window-maximized="0" inkscape:current-layer="svg2" /> <line y2="95.846649" x2="600.5" y1="95.846649" x1="0.5" id="line4" style="stroke:#000000;stroke-width:1" /> <rect height="1.0131806" width="5" y="94.833466" x="20.5" id="rect6" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="20.5" id="text8" style="fill:#000000">0 <rect height="8.2896595" width="5" y="87.556976" x="75.045456" id="rect10" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="75.045456" id="text12" style="fill:#000000">1 <rect height="30.521021" width="5" y="65.325623" x="129.59091" id="rect14" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="129.59091" id="text16" style="fill:#000000">2 <rect height="66.591316" width="5" y="29.255341" x="184.13637" id="rect18" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="184.13637" id="text20" style="fill:#000000">3 <rect height="95.346657" width="5" y="0.5" x="238.68182" id="rect22" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="238.68182" id="text24" style="fill:#000000">4 <rect height="93.613083" width="5" y="2.233551"

x="293.22726" id="rect26" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="293.22726" id="text28" style="fill:#000000">5 <rect height="63.827103" width="5" y="32.019562" x="347.77274" id="rect30" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="347.77274" id="text32" style="fill:#000000">6 <rect height="29.841242" width="5" y="66.005402" x="402.31818" id="rect34" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="402.31818" id="text36" style="fill:#000000">7 <rect height="9.1558361" width="5" y="86.690826" x="456.86365" id="rect38" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="456.86365" id="text40" style="fill:#000000">8 <rect height="1.6646974" width="5" y="94.181946" x="511.40912" id="rect42" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="511.40912" id="text44" style="fill:#000000">9 <rect height="0.13620251" width="5" y="95.710449" x="565.95453" id="rect46" style="fill:#0000ff;fill-opacity:0.4;stroke:#0000ff;stroke-width:1" /> <text y="115.84665" x="565.95453" id="text48" style="fill:#000000">10

Pour savoir si un nombre est premier, un moyen rapide est de regarder s'il est dans le tableau des nombres premiers :

```
$("#premiers div").each (x) ->
 if eval($(this).text()) in crible then n++
$("#composés div").each (x) ->
 if eval($(this).text()) not in crible then n++<div class='code_download'  style='text-align: right;'> <a href='http://irem.univ-reunion.fr/local/cache-code/8ae010c92a6aed45b6b2d337700e56ee.txt'  style='font-family: verdana, arial, sans; font-weight: bold; font-style: normal;'>Télécharger
```

Cet algorithme est très rapide mais évidemment à condition d'avoir le tableau des nombres premiers sous la main. Il s'appelle *crible* parce qu'il est calculé au démarrage par la méthode d'Eratosthène :

```
crible = [2..200]
for diviseur in [2..20]
 crible = (x for x in crible when x<=diviseur or x%diviseur isnt 0)<div class='code_download'  style='text-align: right;'> <a href='http://irem.univ-reunion.fr/local/cache-code/fd29ca154f9c61c59777c25ed321aca4.txt'  style='font-family: verdana, arial, sans; font-weight: bold; font-style: normal;'>Télécharger
```

Enfin, pour finir, un exercice de type Rallye mathématique basé sur le théorème des restes chinois :

<a href="http://irem.univ-reunion.fr/IMG/html/resteschinois.html" title='HTML - 4 ko' type="text/html">[HTML - 4 ko]

système conguentiel