Grouper par quatre (Denise Vella-Chemla, 16.5.2019)

On présente ici une caractérisation des nombres premiers particulière, basée sur des regroupements des nombres 4 par 4 et qui permet de distinguer les nombres premiers de la forme $4k+3$ de ceux de la forme $4k+1$ et les distinguer également de leurs puissances, ce qu'on n'était pas parvenue à trouver jusque-là.

On regroupe dans chaque corps premier $\mathbb{Z}/p\mathbb{Z}$ pour p premier ou dans chaque anneau $\mathbb{Z}/n\mathbb{Z}$ pour n impair les nombres 4 par 4 : chaque groupement contient un nombre, son opposé, son inverse (s'il existe, i.e. si p est premier) et l'opposé de son inverse.

Ensuite, on choisit de passer d'un groupement à l'autre en multipliant par deux l'un des éléments d'un groupement : les résultats d'un programme semble indiquer que pour les nombres premiers, un tel procédé permet de "passer par" chaque groupement une fois et une seule (ce qu'on appelle en théorie des graphes parcourir un chemin Hamiltonien), tandis que cela ne semble pas possible dans le cas des anneaux, i.e. quand n n'est pas premier. Voici le programme de calcul des regroupements des nombres 4 par 4.

```
#include <iostream>
#include <stdio.h>

int main (int argc, char* argv[])
{
  int n, numdugroupe, nmin, nmax, k, m ;
  bool marque[200] ;
  int tab[200][4] ;

  nmin = 3 ;
  nmax = 100 ;

  for (n = nmin ; n <= nmax ; n=n+2)
    {
      std::cout << "\n" << n << "␣-->␣\n" ;
      for (k = 1 ; k <= n ; ++k)
        {
          marque[k] = false ;
          tab[k][1] = 0 ;
          tab[k][2] = 0 ;
          tab[k][3] = 0 ;
          tab[k][4] = 0 ;
        }
      tab[1][1] = 2 ; marque[2] = true ;
      tab[1][2] = (n+1)/2 ; marque[(n+1)/2] = true ;
      tab[1][3] = n-2 ; marque[n-1] = true ;
      tab[1][4] = n-(n+1)/2 ; marque[n-(n+1)/2] = true ;
      std::cout << "groupe␣" << "1␣(" ;
      std::cout << tab[1][1] << "," ;
      std::cout << tab[1][2] << "," ;
      std::cout << tab[1][3] << "," ;
      std::cout << tab[1][4] << ").\n" ;
      numdugroupe = 2 ;

      for (k = 3 ; k <= n/2 ; ++k)
        {
          if (marque[k] == false)
            {
              tab[numdugroupe][1] = k ;
              marque[k] = true ;
              tab[numdugroupe][4] = n-k ;
              marque[n-k] = true ;
              for (m = k+1 ; m <= n/2 ; ++m)
                {
                  if (((k*m) % n == 1) || ((k*m) % n == n-1))
                    {
                      tab[numdugroupe][2] = m ;
                      marque[m] = true ;
                      tab[numdugroupe][3] = n-m ;
                      marque[n-m] = true ;
                    }
                }
```

```
            std::cout << "groupe␣" << numdugroupe ;
            std::cout << "␣(" << tab[numdugroupe][1] << "," ;
            std::cout << tab[numdugroupe][2] << "," ;
            std::cout << tab[numdugroupe][3] << "," ;
            std::cout << tab[numdugroupe][4] << ").\n" ;
            numdugroupe = numdugroupe+1 ;
          }
        }
    }
}
```

Voici son résultat pour les nombres impairs de 7 à 25.

```
7 -->
groupe 1 (2,4,5,3).

9 -->
groupe 1 (2,5,7,4).
groupe 2 (3,0,0,6).

11 -->
groupe 1 (2,6,9,5).
groupe 2 (3,4,7,8).

13 -->
groupe 1 (2,7,11,6).
groupe 2 (3,4,9,10).
groupe 3 (5,0,0,8).

15 -->
groupe 1 (2,8,13,7).
groupe 2 (3,0,0,12).
groupe 3 (4,0,0,11).
groupe 4 (5,0,0,10).
groupe 5 (6,0,0,9).

17 -->
groupe 1 (2,9,15,8).
groupe 2 (3,6,11,14).
groupe 3 (4,0,0,13).
groupe 4 (5,7,10,12).

19 -->
groupe 1 (2,10,17,9).
groupe 2 (3,6,13,16).
groupe 3 (4,5,14,15).
groupe 4 (7,8,11,12).

21 -->
groupe 1 (2,11,19,10).
groupe 2 (3,0,0,18).
groupe 3 (4,5,16,17).
groupe 4 (6,0,0,15).
groupe 5 (7,0,0,14).
groupe 6 (8,0,0,13).
groupe 7 (9,0,0,12).

23 -->
groupe 1 (2,12,21,11).
groupe 2 (3,8,15,20).
groupe 3 (4,6,17,19).
groupe 4 (5,9,14,18).
groupe 5 (7,10,13,16).

25 -->
groupe 1 (2,13,23,12).
groupe 2 (3,8,17,22).
groupe 3 (4,6,19,21).
groupe 4 (5,0,0,20).
groupe 5 (7,0,0,18).
groupe 6 (9,11,14,16).
groupe 7 (10,0,0,15).
```

On retrouve le fait connu que les nombres non premiers à n (i.e. tels que $(n, x) \neq 1$) ne sont pas inversibles (les quadruplets des nombres non inversibles contiennent des zéros parce que le programme

n'a pu leur trouver d'inverse).

Les nombres premiers de la forme $4k+1$ semblent distinguables des nombres premiers de la forme $4k+3$ car un seul de leur quadruplets contient 2 zéros. Aux nombres composés impairs sont associés plusieurs groupements de nombres contenant 2 zéros (pour tous les nombres non-inversibles).
Mais le fait qui est peut-être plus intéressant est qu'il semblerait que l'on puisse parcourir tous les groupes de 4 nombres, une fois et une seule chacun, dans les corps premiers $\mathbb{Z}/p\mathbb{Z}$ (quelle que soit leur forme $4k+1$ ou $4k+3$) en passant simplement d'un élément d'un groupe à un élément d'un autre groupe par une multiplication modulaire par 2 par exemple.

Voici alors les chemins Hamiltoniens pour les nombres premiers 23 et 29 et les chemins qui n'en sont pas pour les nombres composés 25 et 27.

Pour 23, en multipliant par 2, on parcourt les groupements ainsi :

$$G_1 \rightarrow G_3 \rightarrow G_2 \rightarrow G_5 \rightarrow G_4$$

Pour 29, en multipliant par 2, on parcourt les groupements ainsi :

$$G_1 \rightarrow G_3 \rightarrow G_5 \rightarrow G_6 \rightarrow G_2 \rightarrow G_4 \rightarrow G_7$$

Pour 25, en multipliant par 2, on parcourt les groupements ainsi :

$$G_1 \rightarrow G_3 \rightarrow G_2 \rightarrow G_6 \rightarrow G_5 \rightarrow G_6 \ (cycle\ sur\ G_6)$$

Pour 27, en multipliant par 2, on parcourt les groupements ainsi :

$$G_1 \rightarrow G_3 \rightarrow G_6 \rightarrow G_4 \rightarrow G_4 (cycle\ sur\ G_4)$$

Ces propriétés juste découvertes nous font penser aux nombres premiers comme à des sortes d'empilement de carrés (un peu comme les étages d'un immeuble) entre lesquels on se déplace. Au sommet d'un carré donné se trouvent un nombre, son opposé, son inverse et l'opposé de son inverse ; ainsi à chaque carré correspond un petit diagramme qui commute dans la mesure où l'inverse de l'opposé d'un nombre est l'opposé de son inverse. Il faudrait être capable de démontrer que les nombres premiers ont pour propriété que l'application réitérée d'une même opération permet de parcourir tous leurs groupes de nombres associés une fois et une seule (selon un chemin Hamiltonien).

Tout ça a déjà été démontré par Gauss : on vient seulement de comprendre un peu mieux les puissances, et le fait qu'une *racine primitive* de Gauss permet de parcourir toutes les classes modulaires dans $\mathbb{Z}/p\mathbb{Z}$.

Ci-dessous, un des petits carrés de l'immeuble dans $\mathbb{Z}/11\mathbb{Z}$ entre lesquels on monte ou descend par l'élévation à la puissance, en ayant démarré sur une racine primitive pour passer une fois et une seule par chaque coin de chaque étage.