

Mise en tables et espaces vectoriels sur corps finis

Lionel VIDAL

Un capitaine part en croisière durant tout le mois d'août avec trente-et-une autres personnes. Il souhaite inviter chaque soir six convives à sa table et voudrait que chaque passager rencontre les autres une et une seule fois au cours de ces dîners. Est-ce possible ?

On considère un corps fini K de cardinal q , et un espace vectoriel E de dimension n sur K , donc isomorphe à K^n .

Une droite de E est définie par un vecteur non nul, ce qui donne $q^n - 1$ choix possibles, et les $q - 1$ vecteurs proportionnels au vecteur choisi définissent la même droite. Il y a donc en tout $(q^n - 1)/(q - 1)$ droites distinctes dans E .

Un hyperplan de E est défini par une forme linéaire du dual E^* , et deux formes linéaires définissent le même hyperplan si et seulement si elles sont proportionnelles. Il y a donc autant d'hyperplans dans E que de droites dans E^* , donc que de droites dans E , puisque E^* et E sont isomorphes.

On applique cela à $E = (\mathbb{Z}/5\mathbb{Z})^3$: cet espace vectoriel contient $(5^3 - 1)/(5 - 1) = 31$ droites et donc 31 plans, chaque plan contenant $(5^2 - 1)/(5 - 1) = 6$ droites. Il suffit alors d'associer chaque passager à une droite et chaque jour du mois d'août à un plan pour déterminer chaque jour une tablée de 6 convives, deux passagers ne se rencontrant qu'une fois à table, car deux droites vectorielles distinctes définissent un plan unique !

Addendum. On peut dénombrer les bases de E par une récurrence finie : on choisit un premier vecteur non nul parmi $q^n - 1$; puis un deuxième vecteur non colinéaire au premier parmi $q^n - q$; puis un troisième n'appartenant pas au plan généré par les deux premiers parmi $q^n - q^2$; ... ; et un n -ième vecteurs parmi $q^n - q^{n-1}$. Le nombre de bases de E est donc $(q^n - 1)(q^n - q)(q^n - q^2) \dots (q^n - q^{n-1})$. Par un raisonnement similaire, le nombre de familles de k vecteurs linéairement indépendants est $(q^n - 1)(q^n - q)(q^n - q^2) \dots (q^n - q^{k-1})$. Comme chaque famille engendre un sous-espace vectoriel de dimension k , on peut dénombrer ces sous-espaces, en regroupant les familles qui sont des bases du même sous-espace.

Le nombre de sous-espaces vectoriels de dimension k de E , $0 < k \leq n$, est donc :

$$\frac{(q^n - 1)(q^n - q)(q^n - q^2) \dots (q^n - q^{k-1})}{(q^k - 1)(q^k - q)(q^k - q^2) \dots (q^k - q^{k-1})}, \text{ ce qui généralise les résultats précédents.}$$

Programme. Voici un programme Haskell très simple qui détermine les 31 tables :

```
1 import Data.List (nubBy)
2
3 tables :: [[Int]]
4 tables = map table guests
5   where
6     table (_,v) = map fst $ filter (orthogonal v . snd) guests
7     guests = zip [1..] $ nubBy colinear
8       [(x,y,z) | x<-[0..4], y<-[0..4], z<-[0..4], (x,y,z)/=(0,0,0)]
9     colinear (x,y,z) (x',y',z') =
10       zero (x*y' - x'*y) && zero (x*z' - x'*z) && zero (y*z' - y'*z)
11     orthogonal (x,y,z) (x',y',z') = zero (x*x' + y*y' + z*z')
12     zero x = x `mod` 5 == 0
13
14 main :: IO ()
15 main = print tables
```

En regroupant les vecteurs colinéaires non nuls de $(\mathbb{Z}/5\mathbb{Z})^3$, on associe à chaque invité, numéroté de 1 à 31, une droite vectorielle définie par son vecteur directeur (lignes 7-8). Mais chaque vecteur directeur est normal à un unique un plan vectoriel. Il suffit alors pour chacun de déterminer l'ensemble des vecteurs directeurs qui lui sont orthogonaux, ce qui donne l'ensemble des droites du plan considéré, et donc une table possible (ligne 6).

Ce programme n'est pas optimisé (il ne tient pas compte, par exemple, de la commutativité du produit scalaire), mais donne la solution immédiatement :

```
[[2,7,12,17,22,27],[1,7,8,9,10,11],[6,7,16,20,24,28],[4,7,14,21,23,30],
[5,7,15,18,26,29],[3,7,13,19,25,31],[1,2,3,4,5,6],[2,11,16,21,26,31],
[2,9,14,19,24,29],[2,10,15,20,25,30],[2,8,13,18,23,28],[1,27,28,29,30,31],
[6,11,15,19,23,27],[4,9,16,18,25,27],[5,10,13,21,24,27],[3,8,14,20,26,27],
[1,17,18,19,20,21],[5,11,14,17,25,28],[6,9,13,17,26,30],[3,10,16,17,23,29],
[4,8,15,17,24,31],[1,22,23,24,25,26],[4,11,13,20,22,29],[3,9,15,21,22,28],
[6,10,14,18,22,31],[5,8,16,19,22,30],[1,12,13,14,15,16],[3,11,12,18,24,30],
[5,9,12,20,23,31],[4,10,12,19,26,28],[6,8,12,21,25,29]]
```

Une gageure pour finir : écrire un programme aussi efficace, sinon aussi simple, qui résolve le problème sans utiliser le détour mathématique présenté ici...