Matrices twadolicieuses, solution

Il est possible d'abstraire les twadolicious matrices en ensemble de circuits, de la manière suivante: positionnez-vous sur un "1" quelconque comme point de départ d'un circuit. Ensuite, déplacez-vous à l'autre "1" de la même rangée. Ensuite, déplacez-vous à l'autre "1" de la même colonne. Ensuite, déplacez-vous à l'autre "1" de la même rangée... ainsi de suite jusqu'à revenir au "1" initial.

Certaines twadolicious matrices sont formées d'un seul circuit, d'autres plusieurs. En reprenant l'exemple habituel,

1 0 0 1 0	0 1 1 0 0	0 1 0 0 1
0 0 1 0 1	0 0 1 1 0	1 0 1 0 0
1 1 0 0 0	1 1 0 0 0	0 0 0 1 1
0 0 1 1 0	0 0 0 1 1	0 1 0 1 0
0 1 0 0 1	1 0 0 0 1	1 0 1 0 0

les deux premières matrices sont formées d'un seul circuit tandis que la troisième est formée de deux circuits. On dira alors que les deux premières matrices sont mono-circuits tandis que la troisième est poly-circuit.

La taille d'un circuit sera le nombre de lignes (ou colonnes) couvertes par le circuit. Chaque circuit de taille i contient donc $2i$ "1". La morphologie d'une twadolicious matrix sera le multiensemble des tailles de chaque circuit de la matrice. Si on prend encore l'exemple précédent, les deux premières matrices ont une morphologie $\{5\}$, tandis que la troisième a une morphologie $\{3, 2\}$.

Il est maintenant temps , pour répondre à la première question, de faire plein de combinatoire.

Nommons T_n le nombre de n-twadolicious matriceset I_n le nombre de matrices n-twadolicious matrices mono-circuits .

D'abord, comptons le nombre de manières de former une n-twadolicious matrix mono-circuit. Il y a C_2^n manières de choisir où placer les "1" sur la première rangée. Ensuite, suivons la colonne du plus à gauche de ces deux "1". Il y a $n-1$ manières de choisir sur quelle rangée de cette colonne placer l'autre "1". Suivons cette rangée. Il y a $n-2$ manières de choisir sur quelle colonne de cette rangée placer l'autre "1"...ainsi de suite. On obtient $I_n = C_2^n \times (n-1) \times (n-2) \times (n-2) \cdots = \frac{n!(n-1)!}{2}$.

Pour former une n-twadolicious matrix poly-circuit, on choisit la taille du circuit touchant la première rangée de la matrice (nommons cette taille i). Ensuite, on choisit les autres rangées couvertes par ce circuit ($\binom{n-1}{i-1}$ possibilités), les colonnes couvertes par ce circuit ($\binom{n}{i}$ possibilités), comment organiser ce circuit (I_i possibilités) et comment organiser le reste de la matrice (T_{n-i} possibilités). Il s'en suit l'équation récursive suivante:

$$T_n = I_n + \sum_{i=2}^{n-2} \binom{n}{i} \binom{n-1}{i-1} I_i T_{n-i} \tag{1}$$

TODO

Le fichier TwadoCounts.java calcule cette formule jusqu'à la valeur de n voulue.

Pour la deuxième question, tout repose sur le théorème suivant: deux matrices sont twado-similar si et seulement si elles ont la même morphologie. (TODO explanations).

Preuve:

$\Rightarrow$: permuter deux lignes (ou deux colonnes) n'affecte pas la composition des circuits. En effet les "1", malgré les permutations, ne changent pas de position dans leur circuit. Donc les circuits maintiennent leur taille.

$\Leftarrow$: on ne détaillera pas la preuve, mais mais pour passer d'une twadolicious matrix à l'autre (avec même morpholgie), il suffit de "mapper" chaque circuit de la première sur un circuit de la deuxième de même taille. On vous laisse y penser.

Il suffit alors de compter le nombre de morphologies possibles pour un certain n.

Nommons $twx(n, m)$ le nombre de morphologies de taille totale n et dont le plus gros circuit est de taille m. $twx(n, m)$ est donc aussi le nombre de multiensembles qui respecte les critères suivants:

1. Ses éléments sont des entiers supérieurs ou égaux à 2.
2. Son plus grand éléments est m.
3. La somme de ses éléments vaut n.

Pour former un tel multiensemble, on peut prendre un multiensemble qui respecte les critères suivants:

1. Ses éléments sont des entiers supérieurs ou égaux à 2.
2. Son plus grand élément n'est pas plus grand que m.
3. La somme de ses éléments vaut $n - m$.

et on lui ajoute m.

Ceci amène donc à la formule récursive suivante:

$$twx(n, m) = \sum_{i=2}^{m} twx(n - m, i) \tag{2}$$

.

Ensuite, le nombre de morphologies de taille totale n est obtenu en sommant sur toutes les valeurs possibles de taille du plus gros circuit:

$$ceq(n) = \sum_{i=2}^{n} twx(n, i) \tag{3}$$

.

ce qui répond à deuxième question.

La troisième question est la plus difficile. Une 2020-twadolicious matrix est reçue en argument et il faut trouver combien il y a de matrices dans sa classe d'équivalence. La première chose à faire est de parser le fichier TODO.txt pour prendre connaissance de la morphologie de la matrice. Une fois que c'est fait, supposons qu'elle a la morphologie $(t_1, t_2, t_3 \dots t_k)$. k est donc le nombre de circuits et $\sum_{i=1}^{k} t_i = n$.

On doit maintenant former une matrice de cette morphologie. Il y a $\frac{n!}{\prod_{i=1}^{k} t_i!}$ manières de décider quelles colonnes sont réservées à quels circuits. Il y a aussi $\frac{n!}{\prod_{i=1}^{k} t_i!}$ manières de décider quelles rangées sont réservées à quels circuits. Ensuite, chacun de ces circuits peut être individuellement organisé de I_i manières. Ce qui ferait donc le nombre suivant de possibilités?:

$$Answer \stackrel{?}{=} \frac{(n!)^2 \prod_{i=1}^{k} I_i}{\prod_{i=1}^{k} (t_i!)^2} \tag{4}$$

En fait, cette formule est inexacte: elle surestime la réponse. Car si on place le circuit C_r à un ensemble de coordonnées précises A dans la matrice et qu'on place ensuite un circuit C_s de la même taille à un ensemble de coordonnées B, c'est équivalent à placer C_r aux coordonnées B et C_s aux coordonnées A. Plus généralement, si on a j circuits d'une même taille à placer, la formule précédente compte $j!$ fois la même configuration puisque qu'il y a $j!$ manières de permuter ces circuits entre eux. On doit donc diviser la réponse par $\prod_{i=1}^{\infty} \rho_i!$, où $\rho_i!$ est le nombre de circuits de taille i dans la morphologie.

$$Answer = \frac{(n!)^2 \prod_{i=1}^{k} I_i}{\prod_{i=1}^{k} (t_i!)^2 \times \prod_{i=1}^{\infty} \rho_i!} \tag{5}$$

Étant donné que $I_n = \frac{n!(n-1)!}{2}$, la formule ci-dessus se simplifie à

$$Answer = \frac{(n!)^2}{2^k \prod_{i=1}^{k} t_i \times \prod_{i=1}^{\infty} (\rho_i!)} \tag{6}$$

Il aurait été impossible de vous demander de retourner ce nombre comme réponse puisqu'il est trop gros (des milliers de chiffres, en base 10), alors le fichier TODO.java calcule et imprime la valeur de ce nombre en mod 999999937. Puisque ce nombre est essentiellement un énorme produit, la méthode utilisée consiste à sauvegarder tous les facteurs de ce nombre (et leurs exposants) dans un tableau et ensuite multiplier tous ces nombres en faisant régulièrement modulo 999999937 pour garder les nombres d'une taille raisonnable pendant le calcul.

En conclusion, voici les réponses obtenues par le code:

Answer to the first question: 16219731108853857300315707916224741121084288000000 Answer to the second question: 562955806547445223176546 Answer to the third question: 301167224