Posplošeni latinski kvadrati

Matjaž Konvalinka

Math. Subj. Class. (2002): 05B15

Pojem latinskega kvadrata posplošimo: $2n \times n$ matrika je *latinski kvadrat reda* 2 *dimenzije* n, če se vsako število od 1 do n pojavi natanko enkrat v vsaki vrstici in natanko dvakrat v vsakem stolpcu ter če so permutacije v vrsticah matrike med seboj različne. Prešteli bomo latinske kvadrate reda 2 dimenzije n za $n \leq 6$.

GENERALIZED LATIN SQUARES

We generalize the well-known concept of a latin square: we call a $2n \times n$ matrix a *latin square of order* 2 *and dimension* n if each of the numbers 1 to n appears exactly once in each row and exactly twice in each column, and if the permutations appearing in the rows of the matrix are all different. We find the number of all latin squares of order two and dimension n for $n \leq 6$.

Na koliko načinov lahko v vrstice matrike reda $n \times n$ vpišemo permutacije n elementov, tako da dobimo vsa števila od 1 do n tudi v vsakem stolpcu? Matrika s to lastnostjo se imenuje latinski kvadrat. Štetje latinskih kvadratov je eden od klasičnih problemov kombinatorike; do danes so znana števila latinskih kvadratov do $n = 10$, glej tabelo 3 in [2].

Standardni problem posplošimo: iščemo število takih matrik $2n \times n$ z različnimi permutacijami po vrsticah, da vsa števila od 1 do n v vsakem stolpcu nastopajo natanko dvakrat. Matriko z zahtevanimi lastnostmi bomo imenovali *latinski kvadrat reda* 2. Število latinskih kvadratov dimenzije n bomo označevali z $L(n)$ ali z $L_1(n)$, število latinskih kvadratov reda 2 dimenzije n pa z $L_2(n)$.

Preden poskusimo prešteti latinske kvadrate pri nekaterih n, si poglejmo, od kod problem izvira. Človeški možgani prepoznajo objekt, čeprav se slike, ki dejansko padejo na očesno mrežnico, med seboj bistveno razlikujejo v odvisnosti od oddaljenosti, svetlobe, vremenskih razmer (dež, megla), bioloških sprememb (rast las, listja), zornega kota in še česa. Kljub tolikemu številu možnosti pa je naša zanesljivost pri prepoznavanju znanih oseb, živali, predmetov itd. osupljiva.

Nevrofiziološke in psihološke študije nakazujejo več možnih razlag tega fenomena. Ena od njih je, da se transformacijsko invariantne reprezentacije objekta (torej tega, da objekt prepoznamo neodvisno od zgoraj omenjenih okoliščin) naučimo tako, da vidimo zaporedje različnih projekcij objekta v kratkih časovnih intervalih, torej da naši možgani grupirajo slike, ki so jih prejeli približno hkrati; seveda objekt prepoznamo tem lažje, čim večkrat in v čim bolj različnih situacijah ga vidimo.

Teorijo potrjuje naslednji poskus, ki ga je izvedel Guy Wallis (glej [3]). Petnajst ljudi razdelimo na tri skupine po pet in obraz vsakega izmed njih fotografiramo pod petimi različnimi koti: -90° (levi profil), -45°, 0° (*en face*), 45° in 90° (desni profil). Prva faza poskusa je učenje. Računalnik 30-krat izvede naslednje: izbere eno od (treh) skupin in prikaže obraze ljudi iz te skupine pod različnimi koti. Če so na primer izbrane osebe A, B, C, D in E iz prve skupine, lahko najprej pokaže obraz osebe C pod kotom $-90°$, potem obraz osebe E pod kotom $-45°$, potem osebo A pod kotom $0°$, nato osebo D pod kotom $45°$ in na koncu osebo B pod kotom $90°$. Pri tem zahtevamo, da izbere vsako skupino natanko desetkrat, da vsako osebo z vsakega kota prikaže po dvakrat in da nobena dva izbora nista popolnoma enaka. V drugi fazi poskusa (testiranje) poskusni osebi 120-krat pokažemo dva obraza pod različnima kotoma, vsakič mora ugotoviti, ali prikazujeta isto osebo. Fazi učenja in testiranja pri vsaki poskusni osebi ponovimo trikrat.

Ljudi iz posamezne skupine učenec vedno vidi skoraj hkrati, v približno zveznem premikanju glave od leve proti desni; zato se v njegovih možgani ti obrazi »zlijejo« v en sam objekt. Posledica tega je, da bistveno uspešneje razlikuje obraze ljudi iz različnih skupin. Velja celo več: večkrat ko poskusna oseba ponovi fazi učenja in testiranja, bolje razlikuje obraze iz različnih skupin in slabše razlikuje obraze iz iste skupine.

Na koliko načinov lahko računalnik izbere obraze v fazi učenja? Če petim osebam priredimo števila od 1 do 5, je vsak izbor zaporedja slik oseb iz te skupine permutacija števil od 1 do 5. Če iz tridesetih izborov petih oseb poberemo skupaj tistih deset, v kateri so ljudje iz ene od skupin, dobimo latinski kvadrat reda 2 dimenzije 5. Če torej izračunamo $L_2(5)$, bo število možnih izborov enako

$$\binom{30}{10} \cdot \binom{20}{10} \cdot \left(L_2(5)\right)^3;$$

najprej moramo izbrati deset (od tridesetih) izborov, v kateri bodo slike oseb iz prve skupine, potem deset (od preostalih dvajsetih) izborov, kjer bodo slike oseb iz druge skupine, število izborov, ki pripadajo vsaki skupini, je natanko $L_2(5)$. S pomočjo računalnika bomo prišli do rezultata $L_2(5) = 40182486220800$, kar bo pomenilo, da je vseh možnih faz učenja kar

$$360148317021007660774856168311841982823614382080000000 \approx 3,6 \times 10^{54}.$$

Malo verjetno je, da bi obstajal kakšen kombinatorični razmislek, ki bi nam dal $L_2(n)$ za poljuben n – taka formula ni znana niti za običajne latinske kvadrate. Prav tako dvomim, da bi obstajala formula, ki bi povezovala število latinskih kvadratov reda 2 z običajnimi latinskimi kvadrati; prva misel bi morda bila, da dobimo vse latinske kvadrate reda 2 tako, da »zlepimo skupaj« po dva običajna latinska kvadrata. Vendar bi tako dobili tudi matrike, ki bi imele v različnih vrsticah isti permutaciji, kar smo prepovedali; poleg tega pa obstajajo latinski kvadrati reda 2, ki *niso* sestavljeni iz dveh običajnih kvadratov: primer je denimo

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 2 \\ 2 & 3 & 1 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 3 & 4 & 2 & 1 \\ 4 & 1 & 2 & 3 \\ 4 & 2 & 3 & 1 \end{bmatrix}.$$

Za začetek opazimo naslednje: iz latinskega kvadrata (reda 2) dobimo nov latinski kvadrat, če

(1) premešamo stolpce,

(2) premešamo vrstice ali

(3) permutiramo števila od 1 do n

Latinska kvadrata reda 2 sta *izomorfna*, če lahko iz enega dobimo drugega z uporabo (1), (2) in (3).
Prvi dve točki lahko zelo preprosto uporabimo: če v latinskem kvadratu reda 2 primerno premešamo stolpce, se prva vrstica spremeni v $12\ldots n$. Premešamo lahko še vrstice, tako da so permutacije razvrščene leksikografsko. Z drugimi besedami: če poznamo število latinskih kvadratov reda 2 s prvo vrstico $12\ldots n$ in z urejenimi vrsticami (takemu latinskemu kvadratu bomo rekli *reduciran*, število reduciranih latinskih kvadratov reda 2 dimenzije n pa bomo označevali z $RL_2(n)$), ga moramo pomnožiti z $n!$ (mešanje stolpcev) in z $(2n-1)!$ (mešanje vrstic razen prve), da dobimo končen rezultat. Velja torej

$$L_2(n) = n!(2n-1)!RL_2(n).$$

Odslej bomo šteli samo reducirane latinske kvadrate.
Da obstaja kakšen latinski kvadrat reda 2, mora biti permutacij reda n vsaj toliko kot vrstic, torej $2n$. Torej je $RL_2(1) = RL_2(2) = 0$. Pri $n = 3$ imamo šest vrstic in šest permutacij, rešitev je ena sama:

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix}.$$

Že pri $n = 4$ pa bi brez računalnika prišli do konca le z veliko potrpežljivosti. Na kratko opišimo, kako teče algoritem. V tabelo dimenzije $2n \times n$ na začetku vpišemo elemente, ki jih bomo fiksirali (vedno bosta to prva vrstica in prvi stolpec, torej $12\ldots n$ in $1122\ldots nn$, kasneje pa tudi druga vrstica). Glavni so trije števci: i označuje trenutno vrstico (in je na začetku 2, če je fiksirana prva vrstica, oziroma 3, če je fiksirana tudi druga), j trenutni stolpec (in je na začetku 2, če je fiksiran prvi stolpec), m pa šteje latinske kvadrate (in je na začetku 0). Na vsakem koraku pogledamo, ali lahko na mesto (i,j) vpišemo kakšno število: to število še ne sme biti uporabljeno na poljih $(i,1),(i,2),\ldots,(i,j-1)$ in ne več kot enkrat uporabljeno na poljih $(1,j),(2,j),\ldots,(i-1,j)$; poleg tega moramo paziti, da je vsaka soda vrstica (po leksikografski urejenosti) za prejšnjo vrstico, z drugimi besedami, če so števila na poljih $(i,k),(i-1,k)$ za $k = 1,2,\ldots,j-1$ enaka, število na (i,j) ne sme biti manjše od tistega na polju $(i-1,j)$. Če števila, ki ustrezajo vsem tem zahtevam, obstajajo, vpišemo najnižjega, ki ga še nismo uporabili, in povečamo j (oziroma povečamo i in damo j na 2, če smo že na koncu vrstice); če ne, zmanjšamo j (oziroma zmanjšamo i in spremenimo j v n, če smo na začetku vrstice). Če je i enak $2n+1$, smo izpolnili celo tabelo, z drugimi besedami, za 1 povečamo m; ko je i enak številu fiksiranih vrstic, smo našli že vse možnosti in s programom zaključimo.
Program je torej dovolj preprost in, napisan v Mathematici, nam v nekaj stotinkah sekunde da rezultat $RL_2(4) = 85$ – in s tem $L_2(4) = 85 \cdot 4! \cdot 7! = 10281600$. Izomorfnostnih razredov

je 8, njihovi predstavniki pa so:

$$\begin{bmatrix}1&2&3&4\\1&2&4&3\\2&1&3&4\\2&1&4&3\\3&4&1&2\\3&4&2&1\\4&3&1&2\\4&3&2&1\end{bmatrix}\quad\begin{bmatrix}1&2&3&4\\1&2&4&3\\2&1&3&4\\2&3&4&1\\3&4&1&2\\3&4&2&1\\4&1&2&3\\4&3&1&2\end{bmatrix}\quad\begin{bmatrix}1&2&3&4\\1&2&4&3\\2&3&1&4\\2&3&4&1\\3&4&1&2\\3&4&2&1\\4&1&2&3\\4&1&3&2\end{bmatrix}\quad\begin{bmatrix}1&2&3&4\\1&2&4&3\\2&3&1&4\\2&4&1&3\\3&1&4&2\\3&4&2&1\\4&1&3&2\\4&3&2&1\end{bmatrix}$$

$$\begin{bmatrix}1&2&3&4\\1&2&4&3\\2&3&1&4\\2&4&3&1\\3&1&4&2\\3&4&1&2\\4&1&2&3\\4&3&2&1\end{bmatrix}\quad\begin{bmatrix}1&2&3&4\\1&2&4&3\\2&3&1&4\\2&4&3&1\\3&1&4&2\\3&4&2&1\\4&1&2&3\\4&3&1&2\end{bmatrix}\quad\begin{bmatrix}1&2&3&4\\1&3&4&2\\2&1&3&4\\2&4&1&3\\3&1&4&2\\3&4&2&1\\4&2&1&3\\4&3&2&1\end{bmatrix}\quad\begin{bmatrix}1&2&3&4\\1&3&4&2\\2&1&4&3\\2&4&3&1\\3&1&2&4\\3&4&1&2\\4&2&1&3\\4&3&2&1\end{bmatrix}$$

Opogumljen s hitrostjo in preprostostjo programa sem ga pognal še za $n = 5$ – in čakal na rezultat $RL_2(5) = 922768$ (in s tem $L_2(5) = 40182486220800$) približno 2 uri.
Pri tej hitrosti bi pri $n = 6$ na rezultat čakal okoli 600 let; nujne so bile torej izboljšave postopka. Prva in najbolj očitna je bila, da sem namesto v Mathematici program napisal v hitrejšem Pascalu. Druga je bila uvedba dveh novih tabel, ki sta sproti šteli, koliko posameznih števil je v določeni vrstici oziroma stolpcu (namesto vsakokratnega štetja že vpisanih števil v vrstici in stolpcu).
Naslednje povečanje hitrosti (za približno $(n-2)!$-krat pri nizkih n) izkorišča točko (3) iz definicije izomorfnosti latinskih kvadratov. Denimo, da fiksiramo (poleg prvega stolpca in prve vrstice) tudi drugo vrstico, na primer (pri $n = 5$) 12354. Računalnik hitro prešteje, da je vseh takih latinskih kvadratov 18332. Kaj se pa zgodi, če v teh tabelah zamenjamo 3 in 5? V prvih dveh vrsticah dobimo 12543 in 12534, ker lahko stolpce poljubno mešamo, dobimo v prvi vrstici spet 12345, v drugi pa 12435. Z drugimi besedami: če fiksiramo drugo vrstico kot 12354, je latinskih kvadratov reda 2 toliko, kot če bi bila druga vrstica 12435.
Kratek razmislek pokaže, da lahko vrstico preslikamo v drugo vrstico s permutacijo števil natanko tedaj, ko imata permutaciji v vrsticah pri razbitju na disjunktne cikle isto strukturo. Koliko je različnih struktur? Opazimo, da so različne strukture v bijektivni korespondenci z razdelitvami naravnega števila[1]. Ker je prvi stolpec fiksiran, so pomembne razdelitve števila $n-1$.
Poglejmo si najprej primer $n = 4$. Imamo pet možnih drugih vrstic: 1243, 1324, 1342, 1423, 1432, in dve različni strukturi: transpozicija (1243, 1324, 1432) in tricikel (1342, 1423). Pri fiksirani drugi vrstici 1243 dobimo 13 možnosti, pri 1342 pa 23 možnosti. Skupaj je možnosti $3 \cdot 13 + 2 \cdot 23 = 85$. Pri $n = 5$ so štiri možne strukture: transpozicija (6 različnih vrstic), dve transpoziciji (3), tricikel (8) in štiricikel (6). Pri drugi vrstici 12354, 13254, 12453 in 13452 dobimo zaporedoma 18332, 60548, 33464 in 60570 možnosti, skupaj torej

[1] *Razdelitev naravnega števila* je tako zaporedje $(\lambda_1, \lambda_2, \ldots, \lambda_k)$, da velja $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0$ in $\lambda_1 + \lambda_2 + \ldots + \lambda_k = n$; število različnih razdelitev števila n označimo s p_n. Ker je $5 = 1+1+1+1+1 = 2+1+1+1 = 2+2+1 = 3+1+1 = 3+2 = 4+1 = 5$, je $p_5 = 7$.

$6 \cdot 18332 + 3 \cdot 60548 + 8 \cdot 33464 + 6 \cdot 60570 = 922768$; računalnik pa jih je moral prešteti samo $18332 + 60548 + 33464 + 60570 = 172914$, porabili smo torej več kot petkrat manj časa kot prej.

V splošnem je možnih drugih vrstic $(n-1)! - 1$ – prvi element namreč mora biti 1, permutacije $12 \ldots n$ pa v drugi vrstici ne more biti. Različnih struktur je za ena manj, kot je razdelitev števila $n-1$, torej $p_{n-1} - 1$, če uporabimo standardno oznako – spet odštejemo 1 zato, ker razdelitev na same cikle dolžine ena v drugi vrstici ni možna. Pri štetju različnih vrstic z določeno strukturo uporabimo nekaj preproste kombinatorike; npr. če štejemo (pri $n = 6$) vse vrstice z dvema transpozicijama, dobimo $5 \cdot 3 = 15$ različnih vrstic (število, ki ne bo v nobeni od transpozicij, izberemo na pet načinov – ker je lahko $2, 3, 4, 5, 6$ – ostala štiri števila pa lahko v dve transpoziciji damo na tri načine).

Kar precej (za okoli 80%) je algoritem pospešila še naslednja ugotovitev. Ko smo v matriki izpolnili že vse vrstice razen zadnjih dveh, sta tudi zadnji dve že skoraj enolično določeni. Oglejmo si primer; denimo, da je prvih 8 vrstic reduciranega latinskega kvadrata reda 2 dimenzije 5 naslednjih:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 5 & 4 \\ 2 & 1 & 5 & 3 & 4 \\ 2 & 5 & 4 & 1 & 3 \\ 3 & 4 & 2 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 3 & 1 & 2 & 5 \\ 4 & 5 & 1 & 3 & 2 \end{bmatrix}$$

Kakšni sta lahko zadnji dve vrstici? Na prvih mestih je seveda 5; ker imamo v drugem stolpcu dve dvojki, štirici in petici ter le eno enico in trojko, bosta na drugem mestu 1 in 3 – očitno bo enica v deveti, trojka pa v deseti vrstici, ker zahtevamo, da so vrstice urejene. Tudi v petem stolpcu manjkata 1 in 3, torej bo 3 v deveti, 1 pa v deseti vrstici. V tretjem in četrtem stolpcu obakrat manjkata 2 in 4. Zato lahko bodisi vpišemo 2 in 4 na tretje in četrto polje v deveti vrstici in 4 in 2 v deseti bodisi obratno – skratka, tabelo lahko dopolnimo na dva načina.

V splošnem je razmislek podoben. Predstavimo stolpce od drugega do n-tega s točkami grafa (ki ima lahko dvojne povezave in zanke), v katerem sta točki povezani, če pripadajočima stolpcema manjka isto število. Vsakemu stolpcu manjkata dve števili, zato bo stopnja vsake točke grafa 2, graf razpade na disjunktne cikle (ki so lahko tudi dolžine 1 ali 2). Zgornjemu primeru bi tako pripadel graf na štirih točkah z dvema cikloma dolžine 2.

Če izberemo eno od dveh možnih števil v nekem stolpcu in v predzadnji vrstici, so s tem enolično določeni vsi stolpci, ki pripadajo istemu ciklu kot izbrani stolpec. Če so vsi cikli dolžine 1, to pomeni, da je v vsakem stolpcu samo ena možna številka in torej da bi morali biti predzadnja in zadnja vrstica enaki, kar pomeni, da matrike ne moremo dopolniti do latinskega kvadrata reda 2. Če pa je število ciklov dolžine več kot 1 pozitivno in ga denimo označimo s k, je možnih dopolnitev do reduciranega latinskega kvadrata 2^{k-1} – v vsakem takem ciklu lahko namreč neodvisno izberemo eno število, ker pa na ta način dobimo tudi latinske kvadrate, ki nimajo urejene predzadnje in zadnje vrstice, moramo rezultat, 2^k, še razpoloviti. Cikle dolžine več kot 1 lahko hitro preštejemo in s tem ustavimo algoritem že, ko je i enak $2n-1$.

Z opisanim algoritmom se je čas za izračun $RL_2(5)$ zmanjšal na slabe pol sekunde. Za $n = 6$

je običajen osebni računalnik[2] računal približno 67 ur (za milijon možnosti je potreboval okoli 2˙1 sekunde); rezultati so povzeti v tabeli 1.

druga vrstica	vrstic s to strukturo	rezultat
123465	10	5243581176
124365	15	17656056976
123564	20	9597985368
132564	20	32347321656
124563	30	17603986392
134562	24	32300900470

Tabela 1: Število reduciranih latinskih kvadratov dimenzije 6 s fiksno drugo vrstico

Skupni rezultat je tako

$$RL_2(6) = 10 \cdot 5243581176 + 15 \cdot 17656056976 + 20 \cdot 9597985368 +$$

$$+ 20 \cdot 32347321656 + 30 \cdot 17603986392 + 24 \cdot 32300900470 = 2459524009920,$$

$$L_2(6) = 6! \cdot 11! \cdot RL_2(6) = 70686956159405752320000.$$

Oglejmo si nekoliko podrobneje zadnjo tabelo. Kot lahko vidimo, so rezultati pri različnih drugih vrsticah kar precej različni (razmerje med največjim in najmanjšim je približno 6 : 1) – vendar pa sta rezultata pri 124365 in 124563 ter pri 132564 in 134562 zelo skupaj. Isto lahko opazimo pri $n = 5$: pri različnih drugih vrsticah dobimo rezultate, ki so kar daleč narazen, le pri 13254 in 13452 sta rezultata skoraj enaka.

Poglejmo, kako bi ta opažanja razložili. Če je druga vrstica 123465 (prva pa kot vedno 123456), imamo v drugem, tretjem in četrtem stolpcu že po dve dvojki, trojki in štirici. To pomeni, da v preostalih desetih vrsticah odpadejo vse permutacije, ki bi imele na drugem mestu dvojko, na tretjem trojko ali na četrtem štirico. Če je druga vrstica na primer 134562, so v preostalih desetih vrsticah možne vse permutacije. Skratka: več ko imata prvi dve vrstici istih števil na istih mestih, torej več ko ima permutacija v drugi vrstici fiksnih točk, manj možnosti dobimo.

Razlike torej niso presenetljive; presenetljiva pa je regularnost, ki jo opazimo pri razmerjih med rezultati. Oglejmo si tabelo 2, ki nam (za $n = 4, 5, 6$) kaže rezultate pri različnih drugih vrsticah, razvrščene po velikosti, in približek razmerja proti prejšnjemu rezultatu (število $1,7692$ pri $n = 4$ in drugi vrstici 1342 je torej približek za 23/13).

Če imata permutaciji v drugi vrstici isto število fiksnih točk, sta torej rezultata praktično enaka, če pa ima ena permutacija eno fiksno točko manj kot druga, je rezultatov približno 1,8-krat več – empirično ugotovljena ocena, za katero pa je povsem možno, da velja tudi za večje n.

Ko smo posplošili latinske kvadrate v latinske kvadrata reda 2, smo zahtevali, da sta v vsakem stolpcu po dve enaki števili namesto enega. Seveda bi jih lahko zahtevali poljubno mnogo, ne nujno dveh. Rečemo, da je $kn \times n$ matrika *latinski kvadrat reda k dimenzije n*, če ima v vrsticah različne permutacije n elementov in če se v vsakem stolpcu vsako od števil od 1 do n pojavi natanko k-krat. Običajen latinski kvadrat je v tem jeziku latinski kvadrat reda 1.

[2] Pentium 4, hitrost 2.4GHz

n	druga vrstica	rezultat	kvocient
4	1243	13	
	1342	23	$1,7692$
5	12345	18332	
	12453	33464	$1,8254$
	13254	60548	$1,8093$
	13453	60570	$1,0003$
6	123465	5243581176	
	123564	9597985368	$1,8304$
	124563	17603986392	$1,8341$
	124365	17656056976	$1,0030$
	134562	32300900470	$1,8295$
	132564	32347321656	$1,0014$

Tabela 2: Razmerja rezultatov pri fiksirani drugi vrstici za $n = 4, 5, 6$

Latinski kvadrat je reduciran, če je permutacija v prvi vrstici identiteta in če so vrstice urejene. Število (reduciranih) latinskih kvadratov reda k dimenzije n označimo z $L_k(n)$ ($RL_k(n)$), kot pri $k = 2$ premislimo, da velja $L_k(n) = n! \cdot (kn - 1)! \cdot RL_k(n)$. Algoritem, opisan zgoraj, brez težav prilagodimo za iskanje latinskih kvadratov reda k. Tabela 3 nam daje $RL_k(n)$ za nekatere k (stolpci) in n (vrstice); do rezultatov za $k = 1$, $n = 8, 9, 10$, nisem prišel sam, glej [4], [1] in [2].

	1	2	3	4	5	6
3	1	1	0	0	0	0
4	4	85	320	170	20	1
5	56	922768	4206026288			
6	9408	2459524009920				
7	16942080					
8	535281401856					
9	377597570964258816					
10	7580701483160132811489280					

Tabela 3: $RL_k(n)$ za nekatere $3 \leq n \leq 10$ in nekatere k

Kaj lahko povemo v splošnem o $RL_k(n)$? Ker mora biti vrstic vsaj toliko, kot je različnih permutacij n elementov, če želimo, da obstaja kak latinski kvadrat reda k dimenzije n, za $k > (n-1)!$ velja $RL_k(n) = 0$. Dokažemo lahko še naslednjo formulo.

Trditev 0.1: *Za* $k \leq (n-1)!$ *velja* $((n-1)! - k) \cdot RL_k(n) = k \cdot RL_{(n-1)!-k}(n)$.

Dokaz: Namesto reduciranih latinskih kvadratov vzemimo latinske kvadrate, kjer so vrstice urejene, stolpci pa ne (torej je lahko permutacija v prvi vrstici poljubna, ne nujno $12 \ldots n$) – imenujmo jih semireducirani in njihovo število (če so reda k in dimenzije n) označimo s

$SRL_k(n)$. Ker iz takih latinskih kvadratov dobimo vse, če premešamo vrstice, je

$$SRL_k(n) = \frac{L_k(n)}{(kn)!} = \frac{(n-1)! \cdot RL_k(n)}{k}.$$

Denimo, da imamo semireduciran latinski kvadrat reda k dimenzije n, z drugimi besedami, izbor kn permutacij (od $n!$ možnih). Če namesto teh permutacij izberemo vse ostale, dobimo $((n-1)! - k)n \times n$ matriko z različnimi permutacijami po vrsticah in z natanko $(n-1)! - k$ enicami, dvojkami itd. v vsakem stolpcu – z drugimi besedami, dobimo semireduciran latinski kvadrat reda $(n-1)! - k$. Tako prirejanje je involucija in zato bijekcija, s čimer smo dokazali

$$SRL_k(n) = SRL_{(n-1)!-k}(n)$$

oziroma

$$\frac{(n-1)! \cdot RL_k(n)}{k} = \frac{(n-1)! \cdot RL_{(n-1)!-k}(n)}{(n-1)! - k}. \qquad \square$$

Literatura

[1] S.E. Bammel, J. Rothstein: The number of 9×9 Latin squares, Discrete Math. 11 (1975), 93-95

[2] B. D. McKay, E. Rogoyski: Latin squares of order 10, Electronic Journal of Combinatorics, Vol. 2 (1995), No. 3 1-4 http://www.combinatorics.org/Volume_2/PDFFiles/v2i1n3.pdf

[3] G. Wallis: The role of object motion in forging long-term representations of objects, Visual Cognit., 2002, 9 (1/2), 233-247

[4] M. B. Wells: The number of Latin squares of order eight, J. Combin. Theory 3 (1967), 98-99