4/2 (2006), 427–435
tmcs@inf.unideb.hu
http://tmcs.math.klte.hu

Teaching
Mathematics and
Computer Science

Combinatorics – Competition – Excel

László Zsakó

Abstract. In 2001 the Informatics Points Competition of the Mathematics Journal for Secondary School Students (KÖMAL) was restarted [1]. The editors set themselves an aim to make the formerly mere programming competition a bit more varied. Therefore, every month there has been published a spreadsheet problem, a part of which was related to combinatorics. This article is intended to discuss the above mentioned problems and the solutions given to them at competitions. We will prove that traditional mathematical and programming tasks can be solved with a system developed for application purposes when applying a different way of thinking.

Key words and phrases: combinatorics, algorithm, Excel, permutations, recursion, competition.

ZDM Subject Classification: C70, K20, R20, R70.

The Mathematics Journal for Secondary School Students (KÖMAL) used to publish programming tasks in the 1980s but then discontinued this type of exercise for long years. It was not until 2001 that KÖMAL's Informatics Points Competition was restarted. The informatics editorial board, under my direction for 3 years, decided to make the formerly mere programming points competition a bit more varied. Therefore, every month there was published a spreadsheet problem, a part of which was related to combinatorics. The basics of how to solve these tasks can be found in an essential mathematical work published in Hungarian [2].

Turul Török devoted a great deal of work to devising a method for the application of spreadsheets in teaching mathematics [3] so several tasks were inspired by his ideas. This article is aimed at discussing the above mentioned problems proving that traditional mathematical and programming tasks can be solved with a system developed for application purposes, as well.

Spreadsheets are characterized by containing in the cells of a 2-dimensional grid numbers or formulae that are to be calculated on the values of other cells. They offer an excellent but simple method to compute tables such as Pascal's triangle. The solution always lies in forming the computation rule for a given task. Then counting on the rule, you can write a function in accordance with the syntax of the spreadsheet and then extend it over the required domain of the table.

Importantly enough, the set of formulae cannot contain a recursive reference i.e. there must be at least one definite computation sequence. It is typical of spreadsheets that any formula can be easily extended over a rectangular domain. Therefore the function should be formed by leaving the values of the cells not defined in the task empty.

As a first task of this type, we chose the simplest and best-known example. Evidently, our aim was just to make secondary school students participating become familiar with non-traditional, table-like solutions of such tasks.

Basically, the students were confused by the fact that in traditional algorithmic programming you should allocate arrays based on the formula in the specification, decide on the computation order then write loops into the program. Whereas here you should not or better to say, must not deal with the computation order.

Pascal's triangle

I.9. *The usual arrangement of binomial coefficients (Pascal's triangle) can be seen in the figure on the right. Apart from the elements along the sides, it holds for each of them that they are the sum of the element above and the one to the left above.*

Make an Excel table which is able to give the first $n+1$ rows of Pascal's triangle in such a way. Write the value of n in Cell H *Row* 1 ($1 \le n \le 20$)*. In each case make the table consist of exactly $n+1$ row.*

In this case the task was quite simple. Just take any of the connections regarding binomial coefficients and rephrase them in a form that corresponds to the syntax of the spreadsheet.

	A	B	C	D	E	F	G	H
1	1						N =	6
2	1	1						
3	1	2	1					
4	1	3	3	1				
5	1	4	6	4	1			
6	1	5	10	10	5	1		
7	1	6	15	20	15	6	1	

First, you have to consider that for each n $B(n,0)=1$, $B(n,n)=1$, i.e. every cell in Column A and cells A1, B2, C3, … must have the value of 1. There is just one extra task: leave cells belonging to values greater than n empty. Since there is not a loop variable, in the formula you need the row index of the current cell, which is given by the `ROW()` function of the spreadsheet. Spreadsheets usually number rows beginning with 1 and in Pascal's triangle they should be numbered from 0 so they must be filled in from Row 1 to Row $n+1$, with everywhere applying the following formula: `=IF(ROW()<=$H$1+1;1;"")`.

You have several possibilities for the other cells:

- $B(n,k)=B(n-1,k)+B(n-1,k-1)$,
- $B(n,k)=\dfrac{n}{k}\,B(n-1,k-1)$,
- $B(n,k)=\dfrac{n}{n-k}\,B(n-1,k)$.

Counting on the first one, the formula to be written in cell B3: `=IF(ROW() <=$H$1+1;A2+B2;"")`. Then extend this formula in the interior of the triangle, which is rather clumsy. Since it is also difficult to extend 1s into the diagonal, a formula should be found that holds for both the diagonal and the area over it.

The correct solution has a formula containing a triple alternative (interior, diagonal, over the diagonal) to replace the above A2+B2: `=IF(ROW()>COLUMN(); A2+B2;IF(ROW()=COLUMN();1;""))`.

The application of binomial coefficients

I.49. *N supporters watching a match throw up their caps in the air as their team wins and then every fan catches exactly one cap falling down.*

Write a program that displays the number of possible ways of exactly k ($0 \leq k \leq n$) ones out of the n ($0 \leq n \leq 12$) fans getting their own caps back.

Example for all ns and ks not greater than 4:

	K = 0	K = 1	K = 2	K = 3	K = 4
N = 0	1				
N = 1	0	1			
N = 2	1	0	1		
N = 3	2	3	0	1	
N = 4	9	8	6	0	1

The solution here is far from being well-known. You need various combinatoric descriptions. The solution of the problem is a product of two numbers. One of them is: the number of ways of selecting k elements out of n elements – they are the binomial coefficients denoted by $B(n, k)$ while the other one is the number of ways of permuting $n - k$ elements in a way that none of them remains in their original place – this is the $S(n-k)$ number of the scattering permutation of $n-k$ elements.

Therefore the solution is: $H(n, k) = B(n, k)\, S(n - k)$, the first of which can be computed just like the one in problem I.9. The latter, however, is much more complicated.

Consider that $\sum_{k=0}^{n} H(n, k) = n!$, which leads to $n! = \sum_{k=0}^{n} \binom{n}{k} S(n - k)$. Since $\binom{n}{k} = \binom{n}{n-k}$ therefore $n! = \sum_{k=0}^{n} \binom{n}{n-k} S(n - k)$ that is $n! = \sum_{k=0}^{n} \binom{n}{k} S(k)$.

And now you can get down to the spreadsheet. The problem is computed for $n \leq 12$ though even this way you will receive many large values. On worksheet `Bin` compute the values of $B(n, k)$ based on problem I.9.! Start a worksheet called `Work`, where you compute the factorials (column A), and the powers of -1 (column B).

Let both A1 and B1 be 1 (0 factorial, or -1^0)! Then let A2 equal to `=A1*(ROW()-1)`, and B2 equal to `=-B1`! The formulae can be extended downwards to the required extent.

On worksheet `Disp` compute the values in the sum of the numbers of disperse permutations where the value of A1 is: `=Bin!A1*INDEX(Work!$A$1:Work!$A$13;COLUMN();1)*INDEX(Work!$B$1:Work!$B$13;COLUMN();1)`.

Now this can be spread regularly over the required domain. With the aid of this, you can compute the values of $S(n)$ in column C of worksheet `Work`. Let the value of C1 be: `=SUM(OFFSET(Disp!$A$1:Disp!$M$13;ROW()-1;0;1;ROW()))*B1`. This formula can also be spread in column C.

There is nothing left but have $H(n,k)$ computed by the multiplication of two relevant values on worksheet `Solution`.

Let cell A1 contain `=Bin!A1*OFFSET(Work!$C$1:Work!$C$13;ROW()-COLUMN();0;1;1)`! This formula can be extended to domain A1..M13 of the worksheet. There is one more thing to add i.e. like in problem I.9., make an empty cell appear at places with an index greater than n.

Eulerian Number Triangle

I.66. *Euler's number triangle is similar to Pascal's triangle, but instead of the binomial coefficients, it contains the Eulerian numbers $E(n,k)$. For $15 \geq n \geq 0$ and $0 \leq k \leq n$, $E(n,k)$ is defined as the number of permutations of $\{1,2,\dots,n\}$ having exactly k permutation ascents (i.e. having k pairs of adjacent positions in the permutation that are out of order).*

*Prepare your sheet (**i66.xls**), that – on entering the value of n into cell* A1 *– displays the Eulerian numbers $E(n,k)$ in the first $n+1$ rows of the sheet. Numbers should only appear in valid cells.*

$n\backslash k$	0	1	2	3	4	5	6	7	8	9	10
0	1										
1	1	0									
2	1	1	0								
3	1	4	1	0							
4	1	11	11	1	0						
5	1	26	66	26	1	0					
6	1	57	302	302	57	1	0				
7	1	120	1191	2416	1191	120	1	0			
8	1	247	4293	15619	15619	4293	247	1	0		
9	1	502	14608	88234	156190	88234	14608	502	1	0	
10	1	1013	47840	455192	1310354	1310354	455192	47840	1013	1	0

These numbers are not taught in secondary school so students cannot be expected to compute the direct formula. After combinatoric consideration, however, they could find a recursive formula. Therefore, the task is to find a connections (one similar to the various connections among the numbers in Pascal's triangle), and then relying on them, writing a formulation similar in its syntax to the one in problem I.9.

The Pascal's triangle had one addition rule so let us try it here. The total number of permutations of n elements having exactly k permutation ascents can be created in two ways from the total number of permutations of $n-1$ elements.

One of them is all the permutations of $n-1$ elements where there is exactly k ascent. If you put the nth number in an ascent, the number of ascents will remain k further on. This number will remain unchanged even if you place the nth number at the beginning of the sequence.

The other one is all the permutations of $n-1$ elements where there are exactly $k-1$ ascents. Here the number of ascents increases by one if the nth element is put at the end of the sequence. The number of ascents also increases if it is put in a place where there was not a growth.

Therefore the solution is: $E(n,k) = (k+1)\,E(n-1,k)+(n-k)\,E(n-1,k-1)$.

Put n in cell A1. From this concludes that the formula `=IF(ROW()<=$A$1+2;ROW()-2;"")` in cell A2 of column A can be spread into maximum 15 rows according to the task. (A similar formula can be entered in the first row.)

Since for $k = 0$ $E(n,k)$ is surely equal to 1 (there is one single monotone decreasing sequence), so cell B2 contains `=IF(A2="";"";1)`, with which column B can be filled in.

For all the other columns leave the area over the triangle empty; surely there must be a 0 in the diagonal ($E(n,n)$) and the other elements can be computed from the previous row with the aid of the above formula. Thus the content of cell C3 to be spread is: `=IF(OR(COLUMN()>ROW();B3="");"";IF(COLUMN()>=ROW();0;(COLUMN()-1)*C2+(ROW()-COLUMN())*B2))`.

Second-Order Eulerian Triangle

I.81. *Prepare a worksheet (i81.xls) that displays second-order Eulerian numbers ($\boldsymbol{E(n,k)}$) in $\boldsymbol{n+1}$ rows if the value of n is written into cell* A1*.*

$\boldsymbol{E(n,k)}$ *can be computed in the following way: First take all permutations of the sequence $\{1,1,2,2,\ldots,n,n\}$ with the property that between the two occurrences* *of any number* $\boldsymbol{m}$ *there are no numbers below* m*. Then* $\boldsymbol{E(n,k)}$ *is the number of the permutations above containing exactly* $\boldsymbol{k}$ *increasing subsequences.*

$n\backslash k$	0	1	2	3	4	5	6	7	8	9	10
0	1										
1	1	0									
2	1	2	0								
3	1	8	6	0							
4	1	22	58	24	0						
5	1	52	328	444	120	0					
6	1	114	1452	4400	3708	720	0				
7	1	240	5610	32120	58140	33984	5040	0			

Please note that when changing from $n-1$ to n, the two nth numbers can be inserted only next to each other. Like first-order Eulerian numbers, here you can also divide the deduction to permutations of $n-1$ elements into two cases.

The first case is absolutely identical with the first case of first-order Eulerian numbers. Whereas in the second case the nth numbers can be put into $2\cdot n-1-k$ place instead of the $n-k$ place.

From the above concludes that $E(n,k)=(k+1)\,E(n-1,k)+(2n-1-k)$ $E(n-1,k-1)$.

Put n in cell A1. Relying on this, the formula `=IF(ROW()<=$A$1+2;ROW()-2;"")` written in cell A2 in column A can be extended into maximum 15 rows according to the task. (A similar formula can be written in the first row.)

Since for $k=0$ $E(n,k)$ is surely equal to 1 (there is one single monotone decreasing sequence), so cell B2 contains `=IF(A2="";"";1)`, with which column B can be filled in.

For all the other columns leave the area over the triangle empty; surely there must be a 0 in the diagonal ($E(n,n)$) and the other elements can be computed from the previous row with the aid of the above formula.

Thus the content of cell C3 to be spread is: `=IF(OR(COLUMN()>ROW();B3="");"";IF(COLUMN()>=ROW();0;(COLUMN()-1)*C2+(2*ROW()-COLUMN()-3)*B2))`.

Conclusion

The solutions of the above tasks and their successful debut at competitions for secondary school students prove that combinatoric problems relying on filling in tables are well suited to this age group, as well. A new opportunity is given for solving tasks of this kind with a tool prepared for basically different purposes i.e. with a spreadsheet.

Since each task is based on sensibly filling in a table after recognizing the suitable connections, this new tool can be used for solving several similar non-combinatoric tasks, as well.

In order to illustrate this, relying on the above principle, try to solve the programming task of the Hungarian National Informatics Competition for Secondary School Students of 2006 [4].

A sequence of characters is a palindrome if it is symmetrical i.e. when read from left to right and from right to left are identical. For example if you insert two characters into the sequence $S =$ "Ab3bd" *, you can turn it into a palindrome (you may have* "dAb3bAd" *or* "Adb3bdA"*). If you insert fewer than* 2 *characters, however, you will not turn this sequence of characters into a palindrome.*

Compute minimally how many characters you need to insert to make word S a palindrome. For each (i, j) $1 \leq i \leq j \leq N$ pairs of index denote with $M(i, j)$ the minimum number of characters you have to insert in order to turn the word $S[i \ldots j] = S[i] \ldots S[j]$ into a palindrome. The solution of the task is $M(1, N)$.

This is a dynamic programming task which can be described with the following formula:

$$M(i,j) = \begin{cases} 0, & \text{if } i \geq j, \\ M(i+1, j-1), & \text{if } i < j \text{ and } S[i] = S[j], \\ 1 + \min(M(i+1, j), M(i, j-1)), & \text{if } i < j \text{ and } S[i] \neq S[j]. \end{cases}$$

Put the letters of the sequence of characters in question into row 1 beginning with the second column. Below this fill in a matrix, in the main diagonal and below the main diagonal of which there is 0 everywhere so the first condition of the formula containing a 3-branch alternative written in cell B2 is:

```
=IF(ROW()>=COLUMN();0;
```

The second branch describes the case when the two suitable letters are identical. It is a bit more complicated to formulate with a spreadsheet: you must use the matrix-function called `INDEX`. If they are identical take the value of the cell to the left down, so the formula continues this way:

```
IF(INDEX($A$1:$Z$1;1;ROW())=INDEX($A$1:$Z$1;1;COLUMN());A4;
```

After all this, the third branch is fairly simple: just increase the minimum of the values of the cells to the left and down with one thus the end of the formula `1+MIN(B3;A2)))`. The formula given this way can be spread from cell B2 downwards to the required distance and then the values in column B to the right to the required distance.

$$\begin{array}{r|ccccccc|}
S = & \mathbf{s} & \mathbf{a} & \mathbf{l} & \mathbf{a} & \mathbf{m} & \mathbf{o} & \mathbf{n} \\
 & 0 & 1 & 2 & 1 & 2 & 3 & \boxed{4} \\
 & 0 & 0 & 1 & 0 & 1 & 2 & 3 \\
 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\
M = & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}$$

References

[1] `http://www.komal.hu/`

[2] Roald R. Graham, Donald E. Knuth, Oren Patashnik, *Concrete Mathematics*, Addison-Wesley Publishing Company Inc., 1994, Hungarian translation: Konkrét matematika, Műszaki Könyvkiadó, 1998..

[3] Török Turul, *Problem solving*, `http://sztech.inf.elte.hu/turul/`, 2005.

[4] `http://nemes.inf.elte.hu/`

EÖTVÖS LORÁND UNIVERSITY
FACULTY OF INFORMATICS
BUDAPEST
HUNGARY

E-mail: `zsako@ludens.elte.hu`

(Received August, 2006)