Round Robin: A Fair Tournament System

Keywords: combinatorics, probability and statistics, combinatorics

Imagine you're organizing a school tournament in table tennis, chess, e-sports, or futsal. You want it to be as fair as possible – so that every player has the chance to face all the others. That's exactly what the round robin system is for.

Its main advantage is fairness: the final ranking depends solely on the players' or teams' performance, not on a random draw of opponents. On the other hand, the number of matches grows quickly with the number of participants – planning such a tournament can be quite a challenge. And this is where combinatorics comes into play – the mathematics of counting possibilities.

Futsal Tournament

Exercise 1. Nine teams have registered for a futsal tournament. It will be played in a round robin format, meaning each team plays one match against every other team. For each win, a team earns 2 points; for a draw, 1 point; and for a loss, 0 points. The final ranking is determined by the total number of points earned across all matches.
How many matches need to be played in the tournament? In how many different ways can the tournament schedule be arranged, assuming there is only one field available and the matches are played one after another?

Solution. The total number of matches played corresponds to the number of all unordered pairs that can be formed from nine teams. In other words, it equals the number of two-element combinations without repetition from a set of nine elements. This gives a total of

$$\binom{9}{2} = 36.$$

To determine the number of possible tournament schedules, we are essentially counting all the different orderings of the 36 matches. Therefore, the total number of possible match schedules is

$$36! = 371{,}993{,}326{,}789{,}901{,}217{,}467{,}999{,}448{,}150{,}835{,}200{,}000{,}000 \doteq 3.72 \cdot 10^{41}.$$

Let us note that if we were to gather a comparable number of grains of sand, each with a volume of approximately $10^{-13}\,\mathrm{m}^3$, the entire pile would have a volume on the order of $10^{28}\,\mathrm{m}^3$, which is roughly ten times the volume of the Sun. Rather than a pile, this would be a relatively massive celestial body.

Exercise 2. Show that if any team in the tournament described in the previous problem earned a total of 13 points, then it must be among the top four teams in the tournament.

Solution. We will use proof by contradiction. Suppose that five teams each earned 13 or more points. Since each match distributes 2 points between the two teams, the total number of points distributed in the entire tournament is $2 \cdot 36 = 72$. Among the five teams, at least $5 \cdot 13 = 65$ points must have been distributed. That would leave at most 7 points for the remaining four teams.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.

Funded by the European Union

However, these four teams play $\binom{4}{2} = 6$ matches among themselves, and must therefore divide 12 points between them. So the total number of points would have to be at least 77, which is impossible. We have reached a contradiction.

Therefore, there can be at most four teams with 13 or more points.

A Fairer Tournament

This time, seven teams have signed up for the next edition of the futsal tournament described in the previous problems. When preparing the tournament schedule, the organizer introduced a new condition: no team is allowed to play in two back-to-back matches. This way, players can avoid playing while tired, and the tournament becomes fairer.

Libor came up with an algorithm for generating a sequence of matches that meets this requirement. His idea is based on the following table:

tým 2	D_1					
tým 3	D_2	D_1				
tým 4	D_3	D_2	D_1			
tým 5	D_4	D_3	D_2	D_1		
tým 6	D_5	D_4	D_3	D_2	D_1	
tým 7		D_5	D_4	D_3	D_2	D_1
	tým 1	tým 2	tým 3	tým 4	tým 5	tým 6

Figure 1: Table for generating a fair tournament schedule

Each cell in the i-th row and j-th column corresponds to the match between team $(i + 1)$ and team j. The desired match sequence will follow the order in which Libor selects these cells. For clarity, we will label the cells according to the teams involved in each match, such as $[1; 2]$, $[3; 5]$, and so on. Next, we define the longest diagonal starting at $[1; 2]$ and ending at $[6; 7]$ as D_1, the shorter diagonal starting at $[1; 3]$ and ending at $[5; 7]$ as D_2, and so on.

Libor's algorithm proceeds as follows: - First, select the cell in the first column and last row, i.e. $[1; 7]$; - Next, go through all the cells of diagonal D_1 that lie in even-numbered columns, from left to right; - Then go through the remaining cells of diagonal D_1 that lie in odd-numbered columns, again from left to right; - Then go through all the cells of diagonal D_2 from left to right; - Then do the same for diagonal D_3, then D_4, and so on.

For a tournament with seven teams, this gives the following match sequence:

$$\begin{array}{ccccccc} [1;7], & [2;3], & [4;5], & [6;7], & [1;2], & [3;4], & [5;6], \\ [1;3], & [2;4], & [3;5], & [4;6], & [5;7], & [1;4], & [2;5], \end{array}$$

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.
Funded by the European Union

$$[3;6],\quad [4;7],\quad [1;5],\quad [2;6],\quad [3;7],\quad [1;6],\quad [2;7].$$

Exercise 3. Using Libor's algorithm, list the sequence of matches for a tournament with 9 teams, and verify that no team appears in two consecutive matches.

Solution. Using the algorithm, we obtain the following sequence of 36 matches. It is clear that no two consecutive matches share a team — all team numbers in neighboring pairs are different.

$$\begin{array}{cccccccccccc}
[1;9], & [2;3], & [4;5], & [6;7], & [8;9], & [1;2], & [3;4], & [5;6], & [7;8], & [1;3], & [2;4], & [3;5], \\
[4;6], & [5;7], & [6;8], & [7;9], & [1;4], & [2;5], & [3;6], & [4;7], & [5;8], & [6;9], & [1;5], & [2;6], \\
[3;7], & [4;8], & [5;9], & [1;6], & [2;7], & [3;8], & [4;9], & [1;7], & [2;8], & [3;9], & [1;8], & [2;9].
\end{array}$$

Exercise 4. Does Libor's algorithm work for any number of participating teams? If not, for which values of n does it work? And can you construct the required sequence for those cases?

Solution. Let n be the number of participating teams (from the context of the problem, it is clear that $n > 1$). From Libor's algorithm, we derive the following sequence of cells, divided into several successive sections based on their positions in the table. For diagonal D_1, we need to distinguish between even and odd n:

$[1;n]$,	(1st cell)
$[2;3],[4;5],\dots,[n-1;n]$,	(1st part of diagonal D_1, odd n)
$[1;2],[3;4],\dots,[n-2;n-1]$,	(2nd part of diagonal D_1, odd n)
$[2;3],[4;5],\dots,[n-2;n-1]$,	(1st part of diagonal D_1, even n)
$[1;2],[3;4],\dots,[n-1;n]$,	(2nd part of diagonal D_1, even n)
$[1;3],[2;4],\dots,[n-2;n]$,	(diagonal D_2)
$[1;4],[2;5],\dots,[n-3;n]$,	(diagonal D_3)
$\vdots$	
$[1;i+1],[2;i+2],\dots,[n-i;n]$,	(diagonal D_i, where $i \le n-2$)
$[1;i+2],[2;i+3],\dots,[n-(i+1);n]$,	(diagonal D_{i+1})
$\vdots$	
$[1;n-1],[2;n]$.	(diagonal D_{n-2})

Two consecutive cells belonging to the same section cannot contain the same number. In both parts of diagonal D_1, any two consecutive cells can be written in the form $[j,j+1]$ and $[j+2,j+3]$, and in any diagonal D_i for $i > 1$, any two consecutive cells have the form $[j,j+i]$ and $[j+1,j+i+1]$. Therefore, it suffices to check under which conditions the last cell of one section can share a number with the first cell of the following section. These special cases must then be considered individually.

1st cell – 1st part of diagonal D_1. The cell $[2;3]$ follows immediately after $[1;n]$ regardless of whether n is even or odd. The required condition that all four numbers (from the two consecutive cells) be distinct is therefore not satisfied for $n = 2$ and $n = 3$.

The portal Math4U is developed within the international projects of the
Department of Applied Mathematics (VSB-Technical University of Ostrava)
and other institutions in Czechia, Poland, Slovakia, and Spain.
Funded by
the European Union

1st part – 2nd part of diagonal D_1. For odd n, the cell $[1;2]$ follows after $[n-1;n]$, which again leads to the already mentioned cases $n=2$ and $n=3$. For even n, the transition is from $[n-2;n-1]$ to $[1;2]$, which also fails the condition when $n=4$.

2nd part of diagonal D_1 – diagonal D_2. For odd n, the transition is from $[n-2;n-1]$ to $[1;3]$. The condition of all components being distinct is violated for $n \in \{2;3;4;5\}$. If n is even, the transition is from $[n-1;n]$ to $[1;3]$. All values of n not excluded so far satisfy the required condition.

Diagonal D_i – diagonal D_{i+1}. The cell $[1;i+2]$ follows after the cell $[n-i;n]$. This gives rise to four possible equalities under which the required condition of all components being distinct is violated:

$$n-i=1, \qquad n-i=i+2, \qquad n=1, \qquad n=i+2.$$

The third equality is clearly not possible. The first equality would imply $i=n-1$, but i can take at most the value $n-2$. If the fourth equality holds, then $i=n-2$; however, diagonal D_{n-2} is the last section, and the sequence ends there — no next diagonal follows. Finally, the second equality can be rewritten as $i=\frac{n-2}{2}$. If n is odd, this cannot hold. But for any even n, there is a unique value of i that satisfies it. Therefore, the algorithm fails for every even value of n; for example, for $n=14$, we have $i=6$, the last element of diagonal D_6 is $[8;14]$ and the first element of diagonal D_7 is $[1;8]$.

In conclusion, Libor's algorithm works without any issues for odd values of n, except for $n=3$ and $n=5$. However, it also trivially works for $n=2$, since two teams play just a single match. For the remaining values — the even n, and the cases $n=3$ and $n=5$ — we now need to try to construct the required sequences in a different way. Let us first note that for $n=3$ and $n=4$, it is impossible:

- For $n=3$, we must order the three matches $[1;2]$, $[1;3]$, $[2;3]$, but every such ordering violates the condition.
- For $n=4$, we may, without loss of generality, choose the first match to be $[1;2]$. The next must then be $[3;4]$, and then again $[1;2]$, which is not allowed.

For all other values of n, we are able to construct sequences with the desired properties. Since there are multiple such sequences (and multiple algorithms to generate them), let us at least give a few examples obtained by modifying the original Libor algorithm. For $n=5$, we modify the algorithm as follows:

- First, select the cell in the first column and last row, i.e. $[1;5]$;
- Next, go through all the cells of diagonal D_1 in even-numbered columns from left to right;
- Then go through the remaining cells of diagonal D_1 in odd-numbered columns from left to right;
- Then go through all the cells of diagonal D_3 from right to left;
- Finally, go through all the cells of diagonal D_2 from right to left.

The resulting sequence is:

$$[1;5], \quad [2;3], \quad [4;5], \quad [1;2], \quad [3;4], \quad [2;5], \quad [1;4], \quad [3;5], \quad [2;4], \quad [1;3].$$

For even n other than 2 and 4, we compute the number $k=\frac{n-2}{2}$. (This number was the source of issues in the general-case discussion above.) We then apply Libor's algorithm with one key modification: we switch the order of diagonals D_{k+1} and D_{k+2} when selecting their cells. Since the rest of the algorithm remains unchanged, it is enough to check only the transitions between these particular diagonals.

Diagonal D_k – diagonal D_{k+2}. The cell $[n-k;n]$ is followed by the cell $[1;k+3]$. Substituting for k, simplifying, and checking for possible repeated numbers yields the following four equalities:

$$\frac{n+2}{2}=1, \qquad \frac{n+2}{2}=\frac{n+4}{2}, \qquad n=1, \qquad n=\frac{n+4}{2}.$$

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.
Funded by the European Union

The second and third equalities cannot hold. The first (and equivalently, the fourth) is satisfied only when $n = 0$ (or $n = 4$), which is invalid in our context.

Diagonal D_{k+2} – diagonal D_{k+1}. The cell $[n-(k+2); n]$ is followed by the cell $[1; k+2]$. Again, we derive four equalities that would violate the condition if true:

$$\frac{n-2}{2} = 1, \qquad \frac{n-2}{2} = \frac{n+2}{2}, \qquad n = 1, \qquad n = \frac{n+2}{2}.$$

None of these equalities can hold, because n cannot be 4, 1, or 2.

Diagonal D_{k+1} – diagonal D_{k+3}. The cell $[n-(k+1); n]$ is followed by the cell $[1; k+4]$. As before, we obtain the following four equalities:

$$\frac{n}{2} = 1, \qquad \frac{n}{2} = \frac{n+6}{2}, \qquad n = 1, \qquad n = \frac{n+6}{2}.$$

The first three equalities are already excluded based on earlier arguments. The fourth equality holds when $n = 6$; however, for this value, diagonal D_{k+3} does not exist, since $k+3 = \frac{6-2}{2} + 3 = 5$. (Recall that for $n = 6$, only diagonals D_1 through D_4 are defined.) Thus, for $n = 6$, the algorithm ends after selecting the members of diagonal $D_{k+1} = D_3$.

The modified algorithm therefore successfully constructs a sequence with the desired properties for all even values of n except for $n = 2$ and $n = 4$. The only natural numbers $n > 1$ for which no such sequence exists are $n = 3$ and $n = 4$.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.

Funded by the European Union