

Round Robin: sprawiedliwy system turniejowy

Keywords: kombinatoryka, prawdopodobieństwo, statystyka, kombinatoryka

Wyobraź sobie, że organizujesz szkolny turniej tenisa stołowego, szachów, e-sportu lub futsalu. Chcesz, aby był on jak najbardziej sprawiedliwy – tak, aby każdy gracz miał szansę zmierzyć się ze wszystkimi innymi. Do tego właśnie służy system round robin.

Jego główną zaletą jest uczciwość: końcowy ranking zależy wyłącznie od wyników graczy lub drużyn, a nie od losowania przeciwników. Z drugiej strony, liczba meczów szybko rośnie wraz z liczbą uczestników – zaplanowanie takiego turnieju może być sporym wyzwaniem. I tu do gry wkracza kombinatoryka – matematyka liczenia możliwości.

Turniej Futsalu

Zadanie 1. Dziewięć drużyn zarejestrowało się do turnieju futsalowego. Będzie on rozgrywany w formacie round robin, co oznacza, że każda drużyna rozegra jeden mecz z każdą inną. Za każde zwycięstwo drużyna otrzymuje 2 punkty, za remis 1 punkt, a za porażkę 0 punktów. Ostateczny ranking jest ustalany na podstawie łącznej liczby punktów zdobytych we wszystkich meczach. Ile meczów należy rozegrać w turnieju? Na ile różnych sposobów można ułożyć harmonogram turnieju, zakładając, że dostępne jest tylko jedno boisko, a mecze są rozgrywane jeden po drugim?

Zadanie 2. Wykaż, że jeśli jakakolwiek drużyna w turnieju opisanym w poprzednim problemie zdobyła łącznie 13 punktów, to musi być wśród czterech najlepszych drużyn w turnieju.

Bardziej sprawiedliwy turniej

Tym razem do kolejnej edycji futsalowego turnieju opisywanego w poprzednich problemach zgłosiło się siedem drużyn. Przygotowując harmonogram turnieju, organizator wprowadził nowy warunek: żadna drużyna nie może zagrać w dwóch meczach back-to-back. W ten sposób zawodnicy unikają gry na zmęczeniu, a turniej staje się bardziej sprawiedliwy.

Libor wymyślił algorytm generowania sekwencji dopasowań, które spełniają ten wymóg. Jego pomysł opiera się na poniższej tabeli:

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

Rysunek 1: Tabela do generowania sprawiedliwego harmonogramu turniejów

Każda komórka w i -tym wierszu i j -tej kolumnie odpowiada meczowi pomiędzy drużyną $(i+1)$ i drużyną j . Pożądana sekwencja dopasowania będzie zgodna z kolejnością, w jakiej Libor wybiera te komórki. Dla przejrzystości będziemy oznaczać komórki zgodnie z drużynami zaangażowanymi w każdy mecz, takimi jak $[1; 2]$, $[3; 5]$ i tak dalej. Następnie zdefiniujemy najdłuższą przekątną zaczynającą się w $[1; 2]$ i kończącą się w $[6; 7]$ jako D_1 , krótszą przekątną zaczynającą się w $[1; 3]$ i kończącą się w $[5; 7]$ jako D_2 i tak dalej.

Algorytm Libor działa w następujący sposób: - Najpierw wybieramy komórkę w pierwszej kolumnie i ostatnim wierszu, czyli $[1; 7]$; - Następnie przechodzimy przez wszystkie komórki przekątnej D_1 , które leżą w parzystych kolumnach, od lewej do prawej; - Następnie przejdź przez pozostałe komórki przekątnej D_1 , które leżą w kolumnach nieparzystych, ponownie od lewej do prawej; - Następnie przejść przez wszystkie komórki przekątnej D_2 od lewej do prawej; - Następnie zrób to samo dla przekątnej D_3 , potem D_4 i tak dalej.

W przypadku turnieju z siedmioma drużynami daje to następującą sekwencję meczów:

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

Zadanie 3. Korzystając z algorytmu Libora, wypisz sekwencję meczów dla turnieju z 9 drużynami i sprawdź, czy żadna drużyna nie występuje w dwóch kolejnych meczach.

Zadanie 4. Czy algorytm Libora działa dla dowolnej liczby uczestniczących zespołów? Jeśli nie, to dla jakich wartości n działa? I czy możesz skonstruować wymaganą sekwencję dla tych przypadków?