

Zaawansowana Matematyka Dyskretna - Projekt 1

Paweł Borowiecki

1 Redukcja

W celu zmniejszenia złożoności obliczeniowej rozważane są jedynie zredukowane kwadraty łacińskie. Jest to uzasadnione tym, że transpozycja kolumn/wierszy nie zmienia liczby transwersal (również parami rozłącznych).

2 Kompilacja

Testowane na kompilatorze g++ 11.2.0:

```
g++ -O3 -fopenmp -march=native --std=c++11 -o orthog orthog.cpp;  
g++ -O3 -fopenmp -march=native --std=c++11 -o trans trans.cpp
```

3 Użycie

- `./orthog n`
Dla każdego zredukowanego kwadratu łacińskiego rzędu n próbuje wyznaczyć kwadrat do niego ortogonalny. W przypadku powodzenia program wyświetla ortogonalną parę i kończy działanie.
- `./trans n`
Dla każdego zredukowanego kwadratu łacińskiego rzędu n wylicza liczbę transwersal (niekoniecznie rozłącznych). Po sprawdzeniu wszystkich kwadratów (lub przerwaniu działania przez użytkownika) program wyświetla najmniejszą i największą wyliczoną liczbę transwersal wraz z przykładami.
- `./orthog 6 && for n in {2..6}; do ./trans $n; done`
Powyższe polecenie wykona wszystkie wymagane obliczenia. Można również rozszerzyć zakres o $n = 7$ - policzenie transwersal dla wszystkich 16, 942, 080 zredukowanych kwadratów łacińskich rzędu 7 zajmuje poniżej dwóch minut (na procesorze Apple M1).

4 Wyniki

Zadanie 1

Dla $n = 6$ nie istnieje para wzajemnie ortogonalnych kwadratów łacińskich.

Zadanie 2

n	$t(n)$	Przykład z $t(n)$ transwersal	$T(n)$	Przykład z $T(n)$ transwersal
2	0	$\begin{array}{cc} 1 & 2 \\ 2 & 1 \end{array}$	0	
3	3	$\begin{array}{ccc} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{array}$	3	
4	0	$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{array}$	8	$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{array}$
5	3	$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 2 & 3 & 1 \\ 5 & 1 & 4 & 2 & 3 \end{array}$	15	$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \\ 3 & 1 & 5 & 2 & 4 \\ 4 & 5 & 2 & 3 & 1 \\ 5 & 3 & 4 & 1 & 2 \end{array}$
6	0	$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 5 & 6 & 4 \\ 3 & 1 & 2 & 6 & 4 & 5 \\ 4 & 5 & 6 & 1 & 2 & 3 \\ 5 & 6 & 4 & 2 & 3 & 1 \\ 6 & 4 & 5 & 3 & 1 & 2 \end{array}$	32	$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 5 & 1 & 3 & 6 & 4 \\ 3 & 1 & 2 & 6 & 4 & 5 \\ 4 & 3 & 6 & 5 & 1 & 2 \\ 5 & 6 & 4 & 1 & 2 & 3 \\ 6 & 4 & 5 & 2 & 3 & 1 \end{array}$
7	3	$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 1 & 5 & 4 & 7 & 6 \\ 3 & 1 & 2 & 6 & 7 & 4 & 5 \\ 4 & 5 & 6 & 7 & 1 & 2 & 3 \\ 5 & 4 & 7 & 1 & 6 & 3 & 2 \\ 6 & 7 & 4 & 2 & 3 & 5 & 1 \\ 7 & 6 & 5 & 3 & 2 & 1 & 4 \end{array}$	133	$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 6 & 1 & 3 & 4 & 7 & 5 \\ 3 & 1 & 4 & 5 & 7 & 2 & 6 \\ 4 & 3 & 5 & 7 & 6 & 1 & 2 \\ 5 & 4 & 7 & 6 & 2 & 3 & 1 \\ 6 & 7 & 2 & 1 & 3 & 5 & 4 \\ 7 & 5 & 6 & 2 & 1 & 4 & 3 \end{array}$

Uwaga. Uruchomienie `./trans n` może generować przykłady inne niż powyższe jeśli skompilowano z opcją `-fopenmp` (z powodu wielowątkowości).