

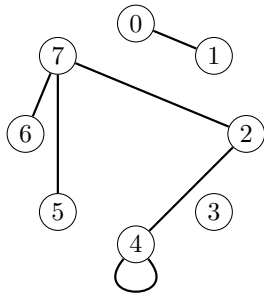
Informatyczny kącik olimpijski (127): Domino

W tym odcinku omówimy rozwiązanie zadania „Domino”, które pojawiło się w 2018 roku w ramach X International Autumn Tournament in Informatics w Szumen, w Bułgarii.

Domino: Zestaw do gry w domino składa się z M kostek ponumerowanych kolejnymi liczbami naturalnymi od 1 do M . Kostka o numerze i jest opisana za pomocą nieuporządkowanej pary liczb (a_i, b_i) , gdzie a_i oznacza liczbę kropek narysowanych na jednej połowie, zaś b_i oznacza liczbę kropek narysowanych na drugiej połowie kostki. Wartości a_i i b_i są liczbami całkowitymi z przedziału $[0; N]$. W zestawie nie ma dwóch takich samych kostek.

Trzeba poukładać kostki domina w łańcuchy. Dwie kostki mogą zostać ze sobą zestawione krótszym bokiem, jeśli odpowiednie połowy tych kostek mają tę samą liczbę narysowanych kropek. Naszym zadaniem jest obliczyć, ile minimalnie łańcuchów należy ułożyć, aby każda kostka domina należała do dokładnie jednego łańcucha, oraz podać te łańcuchy.

Rozwiązanie rozpoczniemy od zbudowania grafu G , złożonego z $N + 1$ wierzchołków oraz M nieskierowanych krawędzi. Wierzchołki ponumerujemy kolejnymi liczbami naturalnymi od 0 do N , każdej liczbie kropek odpowiada jeden wierzchołek. Ponadto, niech istnieje w G krawędź pomiędzy wierzchołkami a_i i b_i dla każdej kostki domina. Przykładowo, jeśli $N = 7$, $M = 6$ oraz dane są następujące kostki domina: $(0, 1)$, $(2, 4)$, $(2, 7)$, $(4, 4)$, $(5, 7)$ i $(6, 7)$, wtedy graf G wygląda jak poniżej:



Możemy teraz wyrazić problem w języku teorii grafów. Dany jest nieskierowany graf. Jaka jest najmniejsza liczba rozłącznych krawędziowo ścieżek, które pokrywają cały graf?

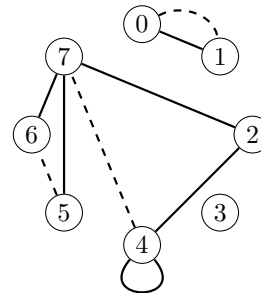
Oczywiste jest, że żadna ścieżka nie będzie przechodziła pomiędzy dwoma różnymi spójnymi składowymi, dlatego każdą spójną składową możemy rozpatrywać osobno. Załóżmy zatem, że rozpatrujemy pewną spójną składową. Zastanówmy się, ile może być w niej wierzchołków o stopniu nieparzystym. Po krótkiej chwili namysłu okazuje się, że wierzchołków o stopniu nieparzystym jest parzyście wiele. Jest tak, ponieważ suma stopni wszystkich wierzchołków jest podwojoną liczbą krawędzi (każda krawędź ma dwa końce). Rozważmy zatem dwa przypadki ze względu na liczbę wierzchołków o stopniu nieparzystym.

0 wierzchołków o stopniu nieparzystym

W tym przypadku każdy wierzchołek ma stopień parzysty. W związku z tym zachodzi warunek konieczny i wystarczający na istnienie w grafie cyklu Eulera (cyklu, który przechodzi przez każdą krawędź dokładnie raz). Popularnym algorytmem do znajdowania cyklu Eulera jest algorytm Fleury’ego. Skoro istnieje cykl Eulera, to wszystkie kostki można ułożyć w jeden łańcuch (ścieżka Eulera).

Co najmniej 2 wierzchołki o stopniu nieparzystym

W tym przypadku jest ich parzysta liczba. Niech k oznacza liczbę wierzchołków o stopniu nieparzystym. Dodajmy do naszego grafu $\frac{k}{2}$ krawędzi łączących wierzchołki o stopniu nieparzystym (każdy wierzchołek o stopniu nieparzystym ma być końcem dokładnie jednej dodanej krawędzi). Dla przykładu opisanego na początku artykułu graf będzie wyglądał następująco (dodane krawędzie zostały oznaczone przerywaną linią):



Otrzymaliśmy multigraf (w multigrafie dwa wierzchołki może łączyć kilka krawędzi), w którym każdy wierzchołek ma parzysty stopień, zatem jest spełniony warunek konieczny i wystarczający na istnienie cyklu Eulera. Znajdźmy więc ten cykl, a następnie usuńmy z niego dodane wcześniej krawędzie. Otrzymane ścieżki opisują kolejne łańcuchy utworzone z kostek domina. W tym przypadku odpowiedzią jest $\frac{k}{2}$.

Optymalność rozwiązania

Pozostało nam jeszcze zastanowić się, czy opisane wyżej rozwiązanie jest optymalne. Zauważmy, że każdy wierzchołek o stopniu nieparzystym musi być początkiem lub końcem jakiegoś łańcucha. Takich wierzchołków jest k , zatem potrzebnych jest przynajmniej $\frac{k}{2}$ łańcuchów. Opisany wyżej algorytm znajduje dokładnie $\frac{k}{2}$ łańcuchów, dlatego jest optymalny.

Złożoność czasowa

Zbudowany przez nas graf ma rozmiar $O(N + M)$. Znajdowanie cyklu Eulera można więc zrealizować w czasie zależnym liniowo od rozmiaru grafu. Całe rozwiązanie działa w czasie $O(N + M)$.

Bartosz ŁUKASIEWICZ