

## Informatyczny kącik olimpijski (55): Bitoniczny komiwojażer



Tym razem w kąciku lekko zmodyfikowana wersja zadania *Listonosz* z konkursu Wielka Przesmycka 2004. W klasycznym problemie komiwojażera mamy dany graf nieskierowany  $G = (V, E)$ , w którym każda krawędź  $(u, v)$  ma przypisaną pewną nieujemną, całkowitą wagę  $w(u, v)$ , i poszukujemy najlżejszego cyklu przechodzącego przez każdy wierzchołek grafu dokładnie raz. Ten problem jest NP-trudny, co znaczy, że nie należy spodziewać się, że uda się go rozwiązać w czasie wielomianowym (przynajmniej nie w czasie zawodów). W bitonicznej wersji problemu komiwojażera wprowadzamy jedno dodatkowe wymaganie dotyczące cyklu. Otóż przyjmujemy, że wierzchołki grafu są ponumerowane od 1 do  $n$  ( $|V| = n$ ), i poszukujemy najlżejszego cyklu przechodzącego raz przez każdy wierzchołek grafu, w którym ciąg numerów wierzchołków jest bitoniczny, czyli najpierw rosnący, a potem malejący (przy pewnym wyborze wierzchołka startowego cyklu). Jak nietrudno się domyślić, wymaganie, by cykl był bitoniczny, istotnie upraszcza podany problem.

Nazwijmy interesujące nas w tym zadaniu cykle cyklami bitonicznymi. Zaczniemy od spostrzeżenia, że wierzchołek numer 1 na pewno znajduje się na początku bądź na końcu każdego cyklu bitonicznego. W drugim z tych przypadków możemy przenieść ten wierzchołek na początek cyklu. To oznacza, że dowolny cykl bitoniczny zaczyna się w wierzchołku 1, prowadzi pewną ścieżką o rosnących numerach wierzchołków aż do wierzchołka  $n$ , po czym wraca do wierzchołka 1, idąc po malejących numerach wierzchołków. Na cykl takiej postaci możemy też spojrzeć nieco inaczej: z wierzchołka 1 prowadzimy dwie rozłączne ścieżki o rosnących numerach wierzchołków, które na końcu spotykają się w wierzchołku  $n$ . Ścieżki o rosnących numerach wierzchołków będziemy dalej nazywali po prostu ścieżkami rosnącymi.

Takie spojrzenie na nasz problem pozwala uzyskać pierwsze rozwiązanie wielomianowe. Zastosujemy metodę programowania dynamicznego. Dla każdej pary wierzchołków  $i, j$ , takich że  $1 \leq i \leq j \leq n$ , przez  $t[i, j]$  oznaczmy sumę wag w najlżejszej parze ścieżek rosnących zaczynających się w wierzchołku 1, kończących się odpowiednio w wierzchołkach  $i$  oraz  $j$ , oraz przechodzących przez każdy wierzchołek – poza wierzchołkiem 1 i, ewentualnie, wierzchołkiem  $n$  – co najwyżej raz. Po chwili namysłu zauważamy, że aby dało się te wartości obliczać, na  $t[i, j]$  musimy nałożyć jeszcze jeden dodatkowy warunek: nasze dwie ścieżki rosnące muszą przechodzić przez wszystkie wierzchołki ze zbioru  $1, 2, \dots, j$ . Teraz jest już całkiem prosto opisać przejścia ze stanu  $t[i, j]$ . Naszą parę ścieżek rosnących możemy rozszerzyć za pomocą krawędzi wychodzącej z wierzchołka  $i$  albo z wierzchołka  $j$ . Aby był spełniony nasz dodatkowy warunek, dodana krawędź musi prowadzić do wierzchołka  $j + 1$ . W pierwszym przypadku aktualizujemy wartość  $t[j, j + 1]$  (za pomocą  $t[i, j] + w(i, j + 1)$ ), a w drugim – wartość  $t[i, j + 1]$  (tym razem za pomocą  $t[i, j] + w(j, j + 1)$ ). Jeśli jednak  $j = n$ , to mamy tylko jedno przejście, do stanu  $t[n, n]$  (za pomocą  $t[i, n] + w(i, n)$ ). Zaczynamy, oczywiście, od stanu  $t[1, 1] = 0$ , a kończymy w stanie  $t[n, n]$ . Jeśli graf będziemy reprezentować za pomocą tablicy sąsiedztwa (czyli funkcję wagi  $w$  zapiszemy w tablicy dwuwymiarowej), otrzymamy algorytm o złożoności czasowej  $O(|V|^2)$ .

Może wydawać się nieco zaskakujące, że istnieje bardziej efektywne rozwiązanie naszego problemu. Aby je dostrzec, należy przyjrzeć się dokładniej strukturze cyklu bitonicznego.

Założmy, że gdzieś na tym cyklu, na jednej ze ścieżek rosnących, leży jakaś „długa” krawędź  $(i, j)$ , czyli taka, że  $i + 1 < j$ . Założmy na razie, że  $i \neq 1$  i  $j \neq n$ . Wówczas możemy od razu powiedzieć coś o drugiej ścieżce rosnącej: mamy na niej jakąś krawędź  $(l, i + 1)$  dla  $l < i$ , dalej znajduje się sekwencja krawędzi „jednostkowych”  $(i + 1, i + 2), (i + 2, i + 3), \dots, (j - 2, j - 1)$ , wreszcie z wierzchołka  $j - 1$  wychodzi krawędź prowadząca do jakiegoś wierzchołka  $k > j$ , patrz rysunek.

Powyższe spostrzeżenie pozwala uprościć postać stanu w naszym programowaniu dynamicznym. Otóż wystarczy nam tylko jeden parametr: stan  $t[j]$  będzie reprezentować najlżejszą parę ścieżek rosnących, z których jedna kończy się w wierzchołku  $j$  i wchodzi tam długą krawędzią, natomiast druga kończy się w wierzchołku  $j - 1$ . Oczywiście, ścieżki muszą być rozłączne, pomijając wierzchołek 1, oraz muszą razem zawierać wszystkie wierzchołki z zakresu  $1, \dots, j$ . Przejścia pomiędzy stanami są wyznaczone przez długie krawędzie. Dokładniej, długa krawędź z wierzchołka  $j - 1$  do wierzchołka  $k > j$  generuje przejście ze stanu  $t[j]$  do stanu  $t[k]$ , jeśli tylko istnieje sekwencja krawędzi jednostkowych prowadzących z wierzchołka  $j$  do wierzchołka  $k - 1$ .

W tym algorytmie będziemy mieli co najwyżej tyle przejść między stanami, ile jest długich krawędzi w grafie, a jedyne wyzwanie polega na tym, żeby efektywnie symulować te przejścia. To już nie będzie trudne, będziemy to wykonywać w czasie stałym. Musimy jedynie umieć sprawdzać, czy dwa zadane wierzchołki są połączone sekwencją krawędzi jednostkowych, a jeśli tak, to jaka jest suma wag tych krawędzi. Do tego celu wystarczą nam ciągi sum częściowych dwóch ciągów: ciągu  $a_i$  oznaczającego istnienie krawędzi  $(i - 1, i)$  (jedynek, jeśli krawędź istnieje, zero, jeśli nie istnieje) oraz ciągu  $b_i = w(i - 1, i)$ . Oznaczmy przez  $A$  i  $B$  tablice reprezentujące odpowiednie ciągi sum częściowych (czyli  $A[i] = a_1 + \dots + a_i$  i analogicznie dla  $B$ ). Tablice te łatwo obliczamy w czasie  $O(n)$ . Wówczas przejście między stanami  $t[j]$  i  $t[k]$  za pomocą długiej krawędzi  $(j - 1, k)$  jest możliwe, gdy  $A[k - 1] - A[j] = k - 1 - j$ , a waga tego przejścia to  $w(j - 1, k) + B[k - 1] - B[j]$ .

Ustalenie stanów początkowych i końcowych w usprawnionym programowaniu dynamicznym pozostawiamy Czytelnikowi. Całe rozwiązanie działa w czasie  $O(|V| + |E|)$ , a zatem liniowo od rozmiaru wejścia.

Jakub RADOSZEWSKI

