

Informatyczny kącik olimpijski (129): Telekomunikacja

W tym odcinku omówimy rozwiązanie zadania „Telekomunikacja”, które pojawiło się na drugim etapie Zawodów Drużynowych XIII Młodzieżowej Olimpiady Informatycznej.

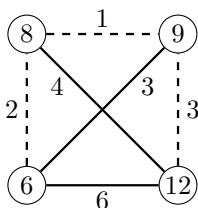
Telekomunikacja: W pewnym państwie znajduje się n miast ponumerowanych kolejnymi liczbami naturalnymi od 1 do n . W każdym mieście znajduje się jedna wieża telekomunikacyjna. Wieża w i -tym mieście jest przystosowana do nadawania i odbierania fal na częstotliwości f_i herców. Dotychczas wszystkie miasta komunikowały się bardzo chaotycznie, dlatego postanowiono to uporządkować. Bezpośrednia komunikacja ma być ograniczona do wybranej sieci połączeń. Sieć ma składać się z $n - 1$ połączeń pomiędzy miastami oraz ma być spójna (innymi słowy ma tworzyć drzewo). Wartością połączenia pomiędzy miastami u i v nazywamy największy wspólny dzielnik liczb f_u i f_v . Król chce wybrać taką sieć, aby suma wartości połączeń była maksymalna. Pomóż mu i podaj tę wartość.

Wstęp

Zacznijmy od opisanego zadania jako problemu grafowego. Dany jest graf pełny $G = (V, E)$, gdzie:

- $V = \{1, 2, \dots, n\}$,
- $E = \{(u, v) : u, v \in V \wedge u \neq v\}$.

Wagą krawędzi $(u, v) \in E$ jest $\text{NWD}(f_u, f_v)$. Chcemy znaleźć sumę wag krawędzi maksymalnego drzewa rozpinającego. Przykładowo, jeśli mamy cztery miasta, w których wieże nadają z częstotliwościami: $f_1 = 8$, $f_2 = 9$, $f_3 = 12$, $f_4 = 6$, to graf wygląda następująco (linie ciągle oznaczają krawędzie maksymalnego drzewa rozpinającego, a suma wag tych krawędzi wynosi $4 + 3 + 6 = 13$):



Niech $M = \max_{u \in V} f_u$.

Rozwiązanie $O(n^2 \cdot \log(M))$

W pierwszym kroku tego rozwiązania zbudujemy wyżej opisany graf, który ma n wierzchołków i $\frac{n(n-1)}{2}$ krawędzi. Wagę dowolnej krawędzi $(u, v) \in E$ możemy obliczyć za pomocą algorytmu Euklidesa, który działa w czasie $O(\log(f_u + f_v))$. Zatem cały graf zbudujemy w czasie $O(n^2 \cdot \log(M))$.

Znajdowanie minimalnego drzewa rozpinającego to znany problem, który może zostać rozwiązany za pomocą jednego z algorytmów: Prima, Kruskala lub Borůvki. Do znalezienia maksymalnego drzewa rozpinającego wykorzystamy zmodyfikowany algorytm Kruskala. Na początku mamy las drzew, gdzie każdy wierzchołek stanowi osobne drzewo. Następnie przeglądamy wszystkie krawędzie w kolejności nierosnących wag. Jeśli rozpatrywana krawędź łączy dwa różne drzewa, to dodajemy ją do drzewa rozpinającego. Sprawdzanie, czy dwa wierzchołki należą do różnych drzew, możemy zrealizować za pomocą struktury zbiorów rozłącznych. Czas wykonania $\frac{n(n-1)}{2}$ operacji (dla każdej krawędzi) na n -elementowym zbiorze wierzchołków wynosi $O(n^2 \cdot \log^*(n))$, gdzie \log^* oznacza logarytm iterowany, czyli liczbę operacji logarytmowania potrzebną do uzyskania

wyniku nie większego od 1. Algorytm wybierze $n - 1$ krawędzi, które tworzą maksymalne drzewo rozpinające. Całkowity czas działania algorytmu wynosi $O(n^2 \cdot \log(n))$ (dominujące jest sortowanie krawędzi po wadze). Jeśli wykorzystamy sortowanie przez zliczanie, to otrzymamy czas $O(M + n^2 \cdot \log^*(n))$. Dowód poprawności algorytmu pozostawiamy Czytelnikowi jako ćwiczenie (dowód jest analogiczny do dowodu algorytmu Kruskala).

Rozwiązanie $O(n + M \cdot \log(M) \cdot \log^*(M))$

W tym rozwiązaniu nie będziemy jawnie konstruowali grafu, natomiast postaramy się szybko znajdować krawędzie maksymalnego drzewa rozpinającego. Zauważmy, że $\text{NWD}(f_u, f_v) \leq \min(f_u, f_v)$ dla dowolnych $u, v \in V$. Zatem krawędzie incydentne z wierzchołkiem $u \in V$ mogą mieć co najwyżej wagę f_u . Na mocy obserwacji połączmy w drzewa wierzchołki o tej samej częstotliwości. Niech $S_x = \{u \in V : f_u = x\}$. Teraz, dla każdego naturalnego $x \in [1; M]$ wierzchołki S_x łączymy w drzewo. Wystarczy np. do wybranego wierzchołka podłączyć pozostałe, tworząc $|S_x| - 1$ nowych krawędzi o wadze x każda. W dalszej części rozwiązania możemy uwzględnić tylko reprezentantów otrzymanych drzew, ponieważ każdy wierzchołek w drzewie ma taki sam zestaw wag krawędzi. Wówczas nie ma dwóch wierzchołków o tym współczynniku.

W zmodyfikowanym algorytmie Kruskala przeglądamy krawędzie w kolejności nierosnących wag. Zatem dla każdego d od M do 1 znajdziemy wierzchołki, które mogą być połączone krawędzią o wadze d . Są to wierzchołki z etykietami: $d, 2d, 3d, \dots, \lfloor \frac{M}{d} \rfloor d$ (wielokrotności d nie większe niż M). Chcemy wszystkie te wierzchołki uspołnić (być może pomiędzy niektórymi wierzchołkami są już krawędzie wygenerowane wcześniej). Wystarczy, że wybierzemy jeden wierzchołek i podłączymy do niego te wierzchołki, które należą do innego drzewa.

Dla każdego d rozpatrzmy $\lfloor \frac{M}{d} \rfloor$ wierzchołków. Zatem w sumie rozpatrzmy $O(M \cdot \log(M))$ wierzchołków. Do sprawdzania spójności wykorzystujemy strukturę zbiorów rozłącznych. W celu szybkiego sprawdzania, czy istnieje wierzchołek z daną etykietą, wystarczy na początku zliczyć wystąpienia w czasie $O(n + M)$. Stąd całkowita złożoność wynosi $O(n + M \cdot \log(M) \cdot \log^*(M))$.

Bartosz ŁUKASIEWICZ