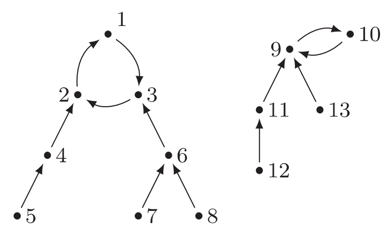
Informatyczny kącik olimpijski (89): Darmowe rozmowy

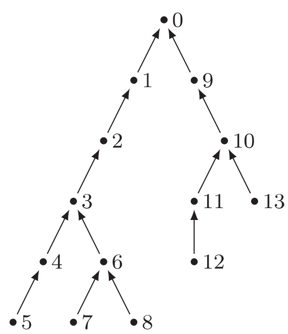
W tym miesiącu rozwiążemy zadanie *Darmowe rozmowy* z Obozu Naukowo-Treningowego im. Antoniego Kreczmara w roku 2011. Firma telekomunikacyjna, chcąc poinformować swoich klientów o nowej promocji, zleciła jednemu ze swoich pracowników, aby osobiście zadzwonił on do niektórych klientów. Klientów jest n, a ponadto każdy z nich może zadzwonić do jednej ustalonej osoby za darmo. Pracownik wychodzi z założenia, że promocja jest tak świetna, że każdy klient, który się o niej dowie, będzie chciał się podzielić tą wiedzą z kimś innym, ale że ludzie są z natury oszczędni, poinformuje on tylko tę osobę, do której może zadzwonić bezpłatnie. Pracownik ma czas wykonać k telefonów do klientów. Należy wyznaczyć, do których powinien zadzwonić, aby zmaksymalizować liczbę osób, które dowiedzą się o promocji.



Rys. 1. Przykładowy graf o $n = 13$ wierzchołkach. Dla $k = 2$ najbardziej opłaca się zadzwonić do klienta 12 (co spowoduje, że o promocji dowiedzą się klienci 12, 11, 9 i 10) oraz jednego z klientów 5, 7 lub 8 (co poinformuje dodatkowych pięciu klientów).

Zbiór klientów możemy przedstawić jako graf skierowany o n wierzchołkach, w którym istnieje krawędź od wierzchołka i do wierzchołka j, jeśli klient i może za darmo zadzwonić do klienta j (rys. 1). Zatem zadzwonienie do klienta i spowoduje, że klienci odpowiadający wszystkim wierzchołkom osiągalnym z wierzchołka i dowiedzą się o promocji.

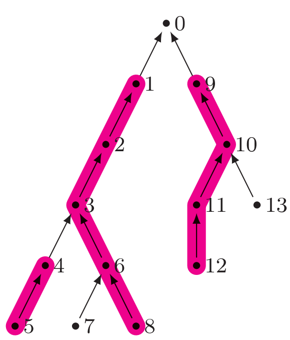
Graf skierowany, w którym z każdego wierzchołka wychodzi dokładnie jedna krawędź, ma dość specyficzną strukturę: każda jego spójna składowa jest cyklem z podoczepianymi drzewami. W przypadku naszego zadania możemy tę strukturę jeszcze uprościć, konstruując skierowane drzewo o tej własności, że z optymalnego rozwiązania dla drzewa łatwo odtworzymy optymalne rozwiązanie dla pierwotnego grafu. Mianowicie każdą składową zastępujemy pojedynczą ścieżką o tej samej długości co cykl w tej składowej, a do końca tej ścieżki podczepiamy wszystkie drzewa ze składowej. Na końcu zaś wszystkie ścieżki podczepiamy do nowego wierzchołka 0 (rys. 2). Pokazanie odpowiedniości między rozwiązaniami dla drzewa i pierwotnego grafu pozostawimy jako nietrudne ćwiczenie dla Czytelników.



Rys. 2. Drzewo skonstruowane na podstawie grafu z rysunku 1.

W tym momencie nasze zadanie jest następujące: chcemy wybrać k wierzchołków w skierowanym drzewie tak, aby liczba wierzchołków z nich osiągalnych była jak największa. Na początek poczyńmy dwie oczywiste obserwacje: wybierając wierzchołki, wystarczy ograniczyć się do liści w drzewie (w szczególności bez straty ogólności możemy założyć, że k jest nie większe niż liczba liści). Ponadto w przypadku $k = 1$ należy wybrać ten z liści, który leży w najdalszej odległości od korzenia drzewa. Okazuje się, że również dalej działa podejście zachłanne: kolejne liście opłaca się wybierać tak, aby leżały jak najdalej od zbioru wierzchołków już zaznaczonych (rys. 3).

Pomysł ten możemy zaimplementować następująco: wykonujemy k faz algorytmu. Zakładamy, że na początku i-tej fazy mamy zaznaczony w drzewie zbiór S_i wierzchołków osiągalnych z liści wybranych w poprzednich fazach. W fazie obliczamy odległości pozostałych wierzchołków do zbioru S_i, wybieramy liść o największej odległości i zaznaczamy wszystkie wierzchołki z niego osiągalne. Pojedynczą fazę wykonujemy w czasie $O(n)$, zatem cały algorytm działa w czasie $O(kn)$.



Rys. 3. Kolejne liście znajdowane przez algorytm zachłanny to, na przykład, 8, 12, 5, 7 i 13. Kolorem zaznaczono wierzchołki osiągalne dodawane w kolejnych fazach algorytmu.

Rozwiązanie to można przyspieszyć. Wierzchołki dodawane w kolejnych fazach tworzą ścieżki. Każda krawędź takiej ścieżki wchodząca do danego wierzchołka wychodzi z jednego z tych jego synów, których poddrzewa mają największą głębokość. Możemy zatem obliczyć wszystkie te ścieżki w czasie $O(n)$, przeglądając drzewo od liści w górę, dla każdego wierzchołka pamiętając głębokość jego poddrzewa. Następnie wybieramy k najdłuższych ścieżek, sortując ich długości przez zliczanie.

A jak udowodnić poprawność rozwiązania zachłannego? Niech R_j będzie zbiorem wierzchołków, które leżą w odległości j od najbliższego liścia. W szczególności R_0 jest zbiorem liści i z tego zbioru chcemy zaznaczyć pewne k wierzchołków. Poruszając się w górę drzewa z tych wierzchołków, zaznaczymy ich rodziców, którzy znajdują się w zbiorze R_1, przy czym opłaca nam się tak wybrać liście, aby zbiór tych rodziców był jak największy. Oczywiście, będzie on miał rozmiar co najwyżej $\min(|R_1|, k)$. W ogólności, ze zbioru R_j zaznaczymy co najwyżej $\min(|R_j|, k)$ wierzchołków. Załóżmy, że zbiór S_i jest optymalnym rozwiązaniem dla $i = k$, to znaczy, że dla każdego j w zbiorze R_j zaznaczamy dokładnie $\min(|R_j|, i)$ wierzchołków. Jeśli liść wybrany w i-tej fazie algorytmu leży w odległości ℓ, to znaczy, że z każdego ze zbiorów $R_0, \ldots, R_{\ell-1}$ zaznaczymy po jednym wierzchołku, natomiast wszystkie wierzchołki w zbiorach $R_\ell, R_{\ell+1}, \ldots$ są już zaznaczone (więc ich rozmiary są nie większe niż i). Wynika z tego, że zbiór S_{i+1} jest też optymalny, bo dla każdego j w zbiorze R_j zaznaczymy $\min(|R_j|, i+1)$ wierzchołków.

Co ciekawe, powyższy dowód daje nam prostsze rozwiązanie w przypadku, gdy interesuje nas tylko maksymalna *liczba* klientów, którzy dowiedzą się o promocji. Wystarczy bowiem w czasie $O(n)$ wyznaczyć rozmiary zbiorów R_j.

Tomasz IDZIASZEK