Matematyka Dyskretna L
Lista 10

Krystian Jasionek

17 grudnia 2020

Zadanie 1.

Przypuśćmy, że w grafie G wszystkie wagi krawędzi są różne. Pokaż, nie używając żadnego algorytmu, że G zawiera tylko jedno minimalne drzewo rozpinające.

Rozwiązanie

Dowód przeprowadzimy nie wprost. Załóżmy, że w grafie G istnieją dwa różne MST, nazwijmy je T_1 oraz T_2. Wtedy istnieje pewne $e = (v, v')$ – bez straty ogólności powiedzmy, że $e \in T_1$ – takie że $c(e) = min\{e$ takich, że $e \notin T_1$ lub $e \notin T_2\}$ i $e \notin T_2$. Wiemy, że T_2 jest MST, zatem musi zawierać ścieżkę p z v do v', gdzie $p \neq e$. W takim razie, po dodaniu e do T_2 otrzymalibyśmy cykl q, bo istniałyby dwie różne ścieżki z v do v'. Zauważmy, że gdyby $q \in T_1$ również otrzymalibyśmy cykl, ale wiemy, że T_1 jest acykliczne. Wynika stąd, że istnieje krawędź $e' \in q$, taka że $e' \notin T_1$. Rozważmy drzewo T' powstałe przez dodanie do T_2 krawędzi e i usunięcie z niego e'. Zauważmy, że skoro wszystkie krawędzi w grafie są różne oraz e ma minimalną wagę spośród krawędzi, które nie należą do któregoś z drzew T_1 i T_2, zatem $c(e) < c(e')$. W takim razie łączna waga drzewa T' jest mniejsza niż łączna waga drzewa T_2. Ale założyliśmy, że T_2 jest MST, zatem ma minimalną wagę. Otrzymaliśmy sprzeczność, więc G musi zawierać tylko jedno minimalne drzewo rozpinające.

Zadanie 2.

Niech T będzie MST grafu G. Pokaż, że dla dowolnego cyklu C grafu G drzewo T nie zawiera jakiejś najcięższej krawędzi z C.

Rozwiązanie

Dowód przeprowadzimy nie wprost. Załóżmy, że T jest MST grafu G oraz T zawiera pewną najcięższą krawędź z cyklu C, zawartego w G. Niech $c = (v, v')$ to najcięższa krawędź z C. Usuńmy z T krawędź c, otrzymamy wtedy dwie spójne składowe drzewa T. Zauważmy, że skoro C jest cyklem, to istnieje jakaś krawędź $c' \in C$, $c' \notin T$, ponieważ cykl zawiera co najmniej trzy krawędzie, które nie są mostami. W takim razie rozważmy T', powstałe przez połączenie tych dwóch składowych krawędzią c'. Zauważmy, że T' jest spójne i zawiera tyle samo krawędzi i wierzchołków co T, zatem jest drzewem. Wiemy, że c było najcięższą krawędzią w C, zatem $c(c) > c(c')$. Możemy powiedzieć, że łączna waga T' jest mniejsza niż łączna waga T, gdyż różnią się one tylko jedną krawędzią – c i c'. Ale założyliśmy, że T jest MST, czyli ma minimalną wagę, otrzymaliśmy sprzeczność. Stąd wiemy, że takie drzewo T nie może zawierać najcięższej krawędzi cyklu.

Zadanie 4.

Udowodnij, że algorytm Prima znajdowania MST działa poprawnie.

Rozwiązanie

Dowód przeprowadzimy indukcyjnie. Weźmy dowolny graf spójny $G = (V, E)$ o nieujemnych wagach na krawędziach. Niech waga krawędzi będzie $w : E \to R \geqslant 0$. Pokażemy, że dla każdego kroku algorytmu (w szczególności ostatniego) istnieje minimalne drzewo rozpinające T_n, zawierające drzewo G_n, powstałe w n-tym kroku algorytmu Prima.

Podstawa: Dla $n = 1$ otrzymujemy drzewo G_1, zawierajace tylko jeden wierzchołek v_1. Skoro dowolne MST grafu G zawiera wszystkie jego wierzchołki, to w szczególności zawiera także v_1. Zatem weźmy dowolne MST grafu G, nazwijmy je T_1. Mamy wtedy, że $G_1 \subseteq T_1$.

Krok: Załóżmy, że $G_n \subseteq T_n$, gdzie T_n to MST grafu G. Pokażemy, że wtedy istnieje takie MST T_{n+1}, że $G_{n+1} \subseteq T_n + 1$. Zauważmy, że algorytm w $n + 1$-szym kroku dodaje do grafu G_n krawędź $e = (v, v')$, gdzie $v \in G_n$ i $v' \notin G_n$. Rozważmy dwa przypadki:

1° $e \in T_n$. Zauważmy, że wtedy $G_{n+1} \in T_n = T_{n+1}$.

2° $e \notin T_n$. Zauważmy, że skoro T_n to MST grafu G, zatem musi istnieć pewna ścieżka $p \in T_n$ z wierzchołka v do v'. Taka ścieżka musi zawierać także pewną krawędź $e' = (u, u')$, gdzie $u \in G_n$ oraz $u' \notin G_n$, inaczej G_n zawierałoby cykl. Rozważmy graf T', powstały przez usunięcie z T_n krawędzi e' i dodanie e, tzn. $T' = T_n + e - e'$. Zauważmy, że T' ma tyle samo krawędzi co T_n oraz skoro usunęliśmy krawędź e' ze ścieżki p, ale dodaliśmy krawędź e, to T' jest spójne. Zatem T' jest drzewem. Wiemy, że algorytm dobiera w każdym kroku krawędzie o najmniejszej wadze, zatem skoro wybrał krawędź e, a nie e', to $w(e) \leqslant w(e')$. Zatem T' jest MST oraz $G_{n+1} \in T'$, zatem istnieje szukane $T_{n+1} = T'$.

Po każdym kroku algorytmu Prima otrzymujemy drzewo $G_n \subseteq T_n$, gdzie T_n jest MST, zatem po ostatnim kroku dostaniemy $G_m = T_m$, czyli MST, zatem algorytm działa poprawnie.

Zadanie 5.

Załóżmy, że wszystkie krawędzie w grafie mają różne wagi. Udowodnij, że algorytm Borůvki rzeczywiście znajduje drzewo rozpinające, tzn. pokaż, że w żadnej iteracji nie powstaje cykl.

Rozwiązanie

Załóżmy nie wprost, że w którejś iteracji algorytmu Borůvki powstał cykl c. To oznacza, że pojawiła się pewna spójna składowa U, zawierająca ten cykl c. Zauważmy, że w algorytmie dokładamy zawsze najlższejszą krawędź incydentną do danego wierzchołka (składowej) oraz wszystkie krawędzie mają różne wagi. Mamy dwa przypadki:

1° cykl c powstał poprzez połączenie dwóch wierzchołków (składowych) $v1$ i $v2$, tzn. dodano incydentne do nich krawędzie, odpowiednio $e1$ oraz $e2$. Skoro wszystkie krawędzie mają różne wagi i e_1 została dodana jako najlżejsza krawędź incydentna do $v1$, zatem $w(e_1) < w(e_2)$. Ale jednocześnie skoro e_2 została dodana jako najlżejsza krawędź incydentna do $v2$, zatem $w(e_2) < w(e_1)$. Zauważmy, że otrzymaliśmy sprzeczność.

2° cykl c powstał przez połączenie co najmniej trzech wierzchołków (składowych). Niech $v_1, v_2, v_3, ..., v_n$ to kolejne wierzchołki należące do cyklu $c = (e_1, e_2, e_3, ..., e_n)$, gdzie e_i to najlższejsza krawędź dla wierzchołka v_i. Cykl powstał przez dodawanie przez algorytm kolejnych krawędzi e_i, zatem c musi posiadać następującą własność: $\mathbf{w(e_1)} < w(e_2) < w(e_3) < ... < w(e_n) < \mathbf{w(e_1)}$, ale wtedy $w(e_1) < w(e_1)$, czyli otrzymaliśmy sprzeczność. To oznacza, że algorytm Borůvki w żadnej iteracji nie wygeneruje cyklu.

Zadanie 6.

Jak zmodyfikować algorytm Borůvki, by działał również w grafach, w których jakieś krawędzie mają takie same wagi?

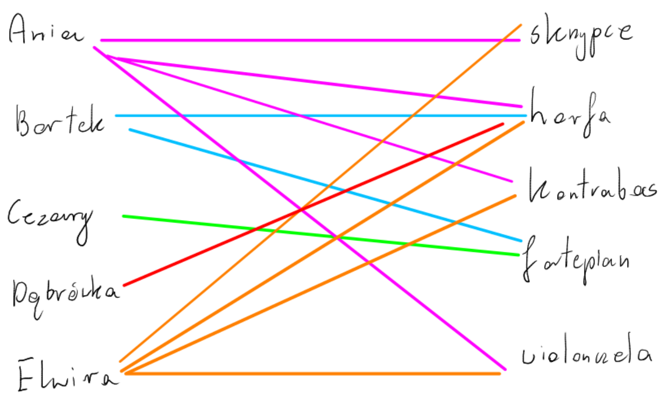
Rozwiązanie

Możemy dodać do algorytmu dodatkową logikę. W przypadku grafów zawierających krawędzie o tych samych wagach możemy poindeksować je kolejnymi liczbami naturalnymi. Gdy w czasie działania algorytmu natrafimy na dwie krawędzie o równej wadze, będziemy wybierali tę o mniejszym indeksie. Wybór dowolnej z nich jest równoważny – mają taką samą wagę, zatem taka modyfikacja nie wpłynie na poprawność algorytmu i dalej będzie on zwracał MST.

Zadanie 10.

W pewniej grupie muzykujących osób Ania gra na skrzypcach, harfie, kontrabasie i wiolonczeli, Bartek gra na harfie i fortepianie, Cezary gra na fortepianie, Dąbrówka gra na harfie, Elwira gra na kontrabasie, skrzypcach, wiolonczeli i harfie. Chcieliby zagrać utwór na fortepian, skrzypce, wiolonczelę, kontrabas i harfę. Czy uda im się dobrać skład?

Rozwiązanie



Każda osoba może grać tylko na jednym instrumencie. Zauważmy, że tylko dwie osoby (Ania oraz Elwira) grają na trzech instrumentach (skrzypcach, kontrabasie i wiolonczeli), zatem nie uda się im złożyć zespołu. Pokażmy to formalnie.

Rozważmy graf $G = (V, E)$, gdzie $V = O \cup I$, gdzie O to zbiór wierzchołków reprezentujących osoby, a I instrumenty. Z twierdzenia Halla wiemy, że warunkiem koniecznym do istnienia skojarzenia doskonałego jest, aby dla każdego

$O' \subseteq O$ oraz dla każdego $I' \subseteq I$ zachodziło $|N(O')| \geqslant |O'|$ i $|N(I')| \geqslant |I'|$, gdzie dla $W \subseteq V$ $N(W) = \{v \in V : \exists_{w \in W} v, w \in E\}$.

Niech $I' = \{skrzypce, kontrabas, wiolonczela\}$. Wtedy $N(I') = \{(Ania, Elwira)\}$, zatem $|I'| = 3$ oraz $|N(I')| = 2$, czyli $|I'| \neq |N(I')|$, więc nie istnieje skojarzenie doskonałe i nie dane będzie im zagrać w zespole.