L10

7 December, 2023 14:31

L10 zad	1	2	3	+4	5	-6	7	8	-9	+10	11	suma
pkt.	1	1	1	?			1	?				4-6/10
maxpkt.	1	1	1	1	1	0,5	1	1	0,5	1	1	10

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

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

3. Czy poniższy algorytm zawsze znajduje MST w grafie spójnym G?

   Załóżmy, że krawędzie grafu są posortowane wg wag: $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_m)$. Dla każdej krawędzi o indeksie i w kolejności od m do 1 wykonaj następujące: jeśli wyrzucenie e_i nie rozspaja G, wyrzuć e_i z G.

4. (+) Udowodnij, że algorytm Prima znajdowania MST działa poprawnie.

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

6. (-) Jak zmodyfikować algorytm Boruvki, by działał również w grafach, w których krawędzie mogą mieć takie same wagi?

7. Krawędzie pewnego grafu G pokolorowano na czerwono i niebiesko. Kiedy graf ten zawiera drzewo rozpinające niemonochromatyczne, tzn. zawierające krawędź każdego koloru?

8. Krawędzie pewnego prostego grafu spójnego G pokolorowano na czerwono, zielono i niebiesko. G ma przynajmniej 4 krawędzie oraz przynajmniej jedną krawędź każdego z trzech kolorów. Czy graf ten w każdym przypadku zawiera drzewo rozpinające trójkolorowe?

   A gdyby kolorów było cztery i G był dodatkowo dwudzielny, czy zawsze zawierałby drzewo rozpinające zawierające przynajmniej jedną krawędź każdego z czterech kolorów? Czy dwudzielność jest potrzebna?

9. (-) Udowodnij lub obal: Jeśli T jest minimalnym drzewem spinającym grafu G, to ścieżka łącząca wierzchołki u i v w drzewie T jest minimalną wagowo ścieżką między u i v w grafie G.

10. (+) Niech $G = (A \cup B, E)$ będzie grafem dwudzielnym, a M i N jego dwoma skojarzeniami. Pokaż, że istnieje skojarzenie M' takie, że każdy wierzchołek $a \in A$ skojarzony w M jest również skojarzony w M' oraz każdy wierzchołek $b \in B$ skojarzony w N jest również skojarzony w M'.

11. Pokaż jak znaleźć największe skojarzenie w drzewie T.

1 (done)

7 December, 2023 14:31

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

MST, minimalne drzewo rozpinające

Drzewo zawierające wszystkie wierzchołki grafu o najmniejszej możliwej wadze (suma krawędzi możliwie jak najniższa)

Dowód będzie nie wprost.

1) Załóżmy, że istnieją dwa różne MST (A, B) spełniające warunek z treści zadania

2) A i B są różne (mimo zawierania tych samych wierzchołków)

⇓

Istnieje niezerowa ilość krawędzi którymi się różnią.

3) Weźmy jedną z tych krawędzi - tę o najmniejszej wadze. Bez straty ogólności można założyć że krawędź ta należy do grafu A. Nazwijmy ją a_1.

4) B jest drzewem $\Rightarrow \{a_1\} \cup B$ jest cyklem (dla uproszczenia nazwijmy ten cykl C)

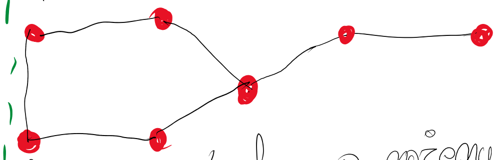
5) A jest drzewem $\Rightarrow$ Cykl C zawiera krawędź a_2 spoza drzewa A. $a_2 \in B$

6) Zauważmy że waga a_2 jest większa od a_1 - a_1 było wybrane jako krawędź o najniższej wadze z krawędzi należących do jednego z tych dwóch grafów.

7) Istnieje więc opcja zastąpienia w drzewie B krawędzi a_2 przez krawędź a_1. B nie jest MST. Powyższy fakt powoduje sprzeczność z założeniem

Przy większej ilości krawędzi działa to tak samo - istnieje przypadek że dostaniemy 2 drzewa z których żadne nie jest MST - zamiast tego z ich krawędzi stworzymy trzecie drzewo które będzie MST

Prawie-drzewo z dodatkową krawędzią



By stworzyć drzewo możemy pozbyć się dowolnej krawędzi należącej do cyklu

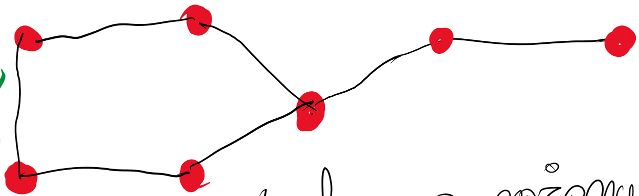
2 (done)

7 December, 2023 14:31

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

Główna myśl zadania bardzo podobna do zadania poprzedniego. Przypomnę najpierw piękną ilustrację z zadania pierwszego by zilustrować intuicyjne rozumienie problemu:

Prawie-drzewo z dodatkową krawędzią

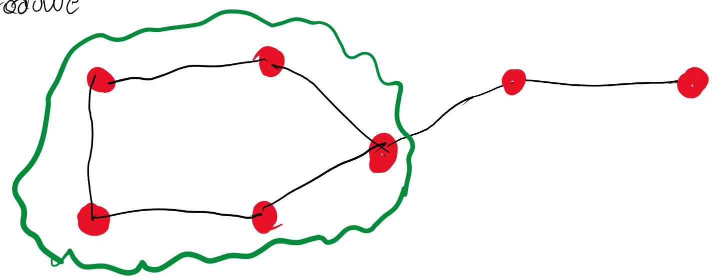


By stworzyć drzewo możemy pozbyć się dowolnej krawędzi należącej do cyklu

A skoro dowolnej - to możemy wybrać najcięższą. Sformalizujmy teraz zapis.

Weźmy T' - jest to T uzupełnione o krawędzie cyklu C które w T się nie znajdują. T' zawiera więc cykl C.

Przykładowe T'



Przykładowe C

Rozważmy teraz wszystkie drzewa które możemy stworzyć z T'. Każde z nich powstanie za pomocą pozbycia się jakiegoś u_i z C.

Po przydługim wstępie właściwy dowód

Załóżmy że T zawiera najcięższą krawędź C i jest MST. Oznaczmy tę najcięższą krawędź jako u_1

Istnieje jednak drzewo T_2 które zawiera lżejszą krawędź u_2 cyklu C i nie zawiera u_1

Załóżmy że T zawiera najcięższą

Istnieje jednak drzewo T_2 które zawiera lżejszą krawędź u_2 cyklu C i nie zawiera u_1

$(T / \{u_1\}) \cup \{u_2\} = T_2$ T_2 ma mniejszą sumę wag i zawiera te same wierzchołki co T.

T nie jest więc wtedy drzewem rozpinającym

3 (done)

7 December, 2023 14:31

3. Czy poniższy algorytm zawsze znajduje MST w grafie spójnym G?

Załóżmy, że krawędzie grafu są posortowane wg wag: $w(e_1) \leq w(e_2) \leq \ldots \leq w(e_m)$. Dla każdej krawędzi o indeksie i w kolejności od m do 1 wykonaj następujące: jeśli wyrzucenie e_i nie rozspaja G, wyrzuć e_i z G.

Tak, algorytm zawsze działa. Intuicja z poprzednich zadań.

1) Graf na początku jest spójny i zawsze sprawdzamy czy przypadkiem go nie rozspoiliśmy, nigdy nie otrzymamy więc grafu niespójnego

2) Zawsze otrzymamy drzewo - usuwamy wszystkie krawędzie które tylko możemy. Jeśli otrzymalibyśmy jakiś cykl w grafie wynikowym, to byłaby jakaś krawędź którą algorytm mógłby usunąć

3) Zaczynamy usuwanie od krawędzi o największej wadze, spełniony będzie więc warunek z zadania drugiego

4) Otrzymamy zawsze n-1 krawędzi, kolejność usuwania nie będzie więc problemem

+4 (done?)

7 December, 2023 14:31

4. (+) Udowodnij, że algorytm Prima znajdowania MST działa poprawnie.

Na papierze

1) Pokażemy że B' jest drzewem

2) rozpinającym

3) minimalnym

ale chyba lepiej nie deklarować

5

7 December, 2023 14:31

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

-6

7 December, 2023 14:31

6. (-) Jak zmodyfikować algorytm Boruvki, by działał również w grafach, w których krawędzie mogą mieć takie same wagi?

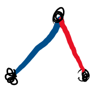
7 (done)

7 December, 2023 14:31

7. Krawędzie pewnego grafu G pokolorowano na czerwono i niebiesko. Kiedy graf ten zawiera drzewo rozpinające niemonochromatyczne, tzn. zawierające krawędź każdego koloru?

MNSP

Indukcja! Dla $n=0,1$ niemożliwe

I)  Dla grafu o dwóch krawędziach

II) Załóżmy że MNSP istnieje dla grafu z n krawędziami z min. 1 czerw i 1 nieb.

III) Rozważmy dla $n+1$ krawędzi: nie musi być ten z poprzedniego, tzn. może być monochromatyczny
$n+1$-szą krawędź. Jest dużo przypadków:
Dołączamy do n-krawędziowego grafu

1) Min. 1 czerw. i 1 niebieska w poprzednim

a) Nie dodajemy wierzchołka $\Rightarrow$ MNSP poprzedniego

b) Dodajemy $\Rightarrow$ MNSP poprzedniego + nowy wierzchołek z krawędzią do niego prowadzącą

2) Poprzednie monochromatyczne MST $\cup$ {new} z aktualnego MST

Musimy dostać krawędź innego koloru

a) Bez nowego wierzchołka $\Rightarrow$ Powstaje cykl, pozbywamy się innej krawędzi i git

b) Z nowym $\Rightarrow$ Bierzemy MST i dołączamy

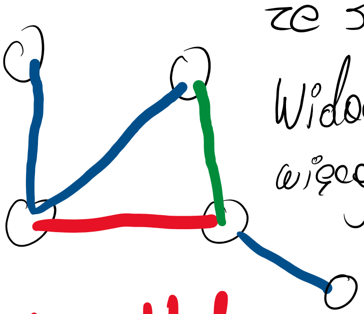
8 (done?)

7 December, 2023 14:31

8. Krawędzie pewnego prostego grafu spójnego G pokolorowano na czerwono, zielono i niebiesko. G ma przynajmniej 4 krawędzie oraz przynajmniej jedną krawędź każdego z trzech kolorów. Czy graf ten w każdym przypadku zawiera drzewo rozpinające trójkolorowe?

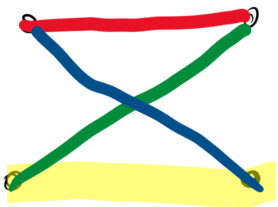
b) A gdyby kolorów było cztery i G był dodatkowo dwudzielny, czy zawsze zawierałby drzewo rozpinające zawierające przynajmniej jedną krawędź każdego z czterech kolorów? Czy dwudzielność jest potrzebna?

a) Min 4 krawędzie to minimum 4 wierzchołki połączone ze sobą. Graf jest spójny z założenia.



Widać wprost że nigdy nie wyrzucimy więcej niż dwóch krawędzi danego koloru

b) **kontrprzykład**



musimy pozbyć się któregoś z tych kolorów

-9

7 December, 2023 14:31

9. (-) Udowodnij lub obal: Jeśli T jest minimalnym drzewem spinającym grafu G, to ścieżka łącząca wierzchołki u i v w drzewie T jest minimalną wagowo ścieżką między u i v w grafie G.

+10

7 December, 2023 14:31

10. (+) Niech $G = (A \cup B, E)$ będzie grafem dwudzielnym, a M i N jego dwoma skojarzeniami. Pokaż, że istnieje skojarzenie M' takie, że każdy wierzchołek $a \in A$ skojarzony w M jest również skojarzony w M' oraz każdy wierzchołek $b \in B$ skojarzony w N jest również skojarzony w M'.

11

7 December, 2023 14:31

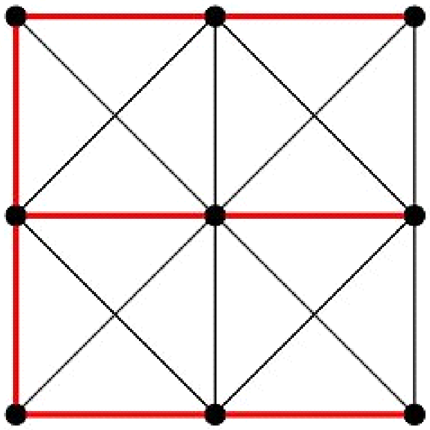
11. Pokaż jak znaleźć największe skojarzenie w drzewie T.

Drzewo rozpinające i MST

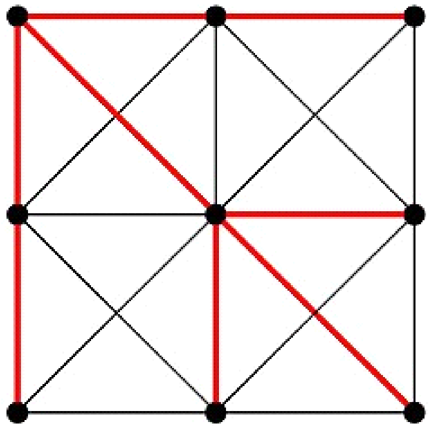
11 December, 2023 22:19

Drzewo rozpinające (ang. *Spanning Tree*) – drzewo, które zawiera wszystkie wierzchołki grafu *G*, zaś zbiór krawędzi drzewa jest podzbiorem zbioru krawędzi grafu.

Konstrukcja **drzewa rozpinającego** polega na usuwaniu z grafu tych krawędzi, które należą do cykli. Najmniejszą liczbę krawędzi jaką trzeba usunąć z grafu, aby graf stał się acykliczny (stał się drzewem) nazywa się **rzędem acykliczności grafu** lub **liczbą cyklometryczną**.



Drzewo rozpinające (czerwone krawędzie) w grafie



Inne drzewo rozpinające w tym samym grafie

Drzewo rozpinające można znaleźć przykładowo wykorzystując algorytm DFS lub Dijkstry.

Minimalne drzewo rozpinające [edytuj]

Artykuł Dyskusja Czytaj Edytuj Edytuj kod źródłowy Wys

Minimalne drzewo rozpinające (ang. *MST*, *minimum spanning tree*) – drzewo rozpinające danego grafu o najmniejszej z możliwych wag, tj. takie, że nie istnieje dla tego grafu inne drzewo rozpinające o mniejszej sumie wag krawędzi.

Definicja formalna [edytuj | edytuj kod]

Dany jest graf ważony *G(V, E, w)*, gdzie **V** – zbiór wierzchołków, *E* – zbiór krawędzi, *w* – funkcja przypisująca krawędzi E_j wagę (liczbę rzeczywistą lub całkowitą).

Minimalnym drzewem rozpinającym nazywamy drzewo rozpinające T, dla którego suma wag

$$\sum_{e \in T} w(e)$$

jest najmniejsza z możliwych. Dla niektórych grafów można wskazać wiele drzew rozpinających spełniających tę własność.

Istnieją trzy deterministyczne algorytmy o złożoności liniowo-logarytmicznej znajdujące dla zadanego grafu minimalne drzewo rozpinające. Są to:

Istnieją trzy deterministyczne algorytmy o złożoności liniowo-logarytmicznej znajdujące dla zadanego grafu minimalne drzewo rozpinające. Są to:

- algorytm Borůvki (błędnie nazywany algorytmem Sollina),
- algorytm Prima (nazywany też algorytmem Dijkstry-Prima),
- algorytm Kruskala.