Algorytmy zachłanne

poniedziałek, 27 maja 2024 10:56

$C: \{c_1, c_2, \dots c_n\}$ – nominały monet

$R \in \mathbb{N}$

$C: \{1, 2, 5, 9, 10\}$

$R: 17$

$17 = 10 + 5 + 2$ ← strategia zachłanna

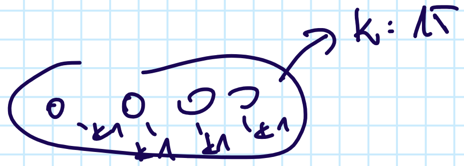
np dla 18, 14 daje rozwiązanie strategia zachłanna, ale nie jest to rozw. optymalne

gdybyśmy wyrzucili 1, to strategia zachłanna nie da 6 np bo da 5, a nie mamy 1, a należy wziąć 3×2.

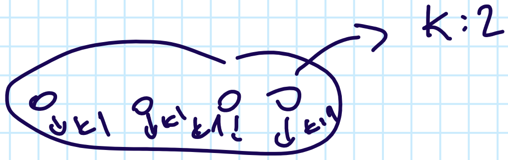
Pokrycie zbioru → chcemy mieć wszystkie elementy z Uniwersum najmniejszym kosztem

czwartek, 30 maja 2024 16:32

1) Najliczniejszy podzbiór $\xrightarrow{\text{koszt}}$



2) Najtańszy podzbiór →



3) Najtaniej pokrywający elementy

cena średniego pokrycia elementów których jeszcze nie pokryliśmy

możemy oszacować $\quad cost(\text{strategia } 3) \le \underbrace{O}_{\text{współczynnik aproksymacji: } \log n}(\text{koszt}(OPT))$

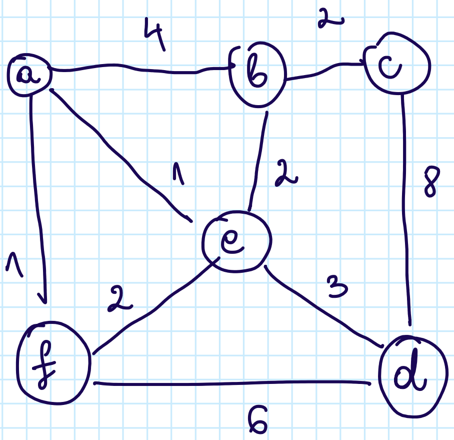
$$cne(S_i) = \frac{c(S_i)}{|S_i / C|}$$

↓

cena na element

Algorytm Prima

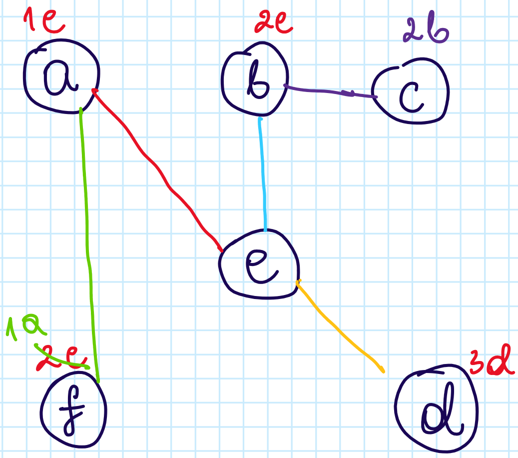
czwartek, 30 maja 2024 18:23



Wybieramy dowolny wierzchołek (e) i liczymy odl. osiągane do innych wierzchołków

wybieramy ten wierzchołek, o najniższej wadze

Czynność powtarzamy dla nowego wierzchołka aktualizując wagi



Algorytm Prima - dowód indukcyjny

W każdym kroku zbiór wybranych krawędzi jest podzbiorem minimalnego drzewa rozpinającego

Podstawa indukcji

- zbiór pusty jest podzbiorem MST.

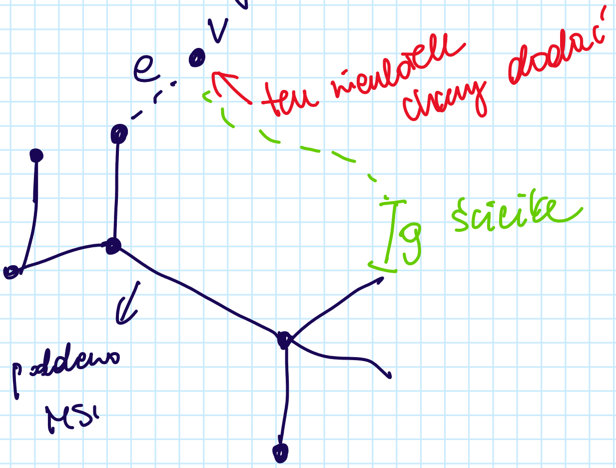
Krok indukcyjny

F - zbiór krawędzi, które zostały już wybrane

T - minimalne drzewo rozpinające

e - krawędź, którą chcemy dodać

e - krawędź, którą chcemy dodać



Chcemy pokazać, że F+e jest podzbiorem krawędzi MST

- jeżeli e należy do MST to git bo należy
- jeżeli e nie należy do MST, to wówczas musi istnieć jakaś ścieżka łącząca wierzchołek v z drzewem T

i na pewno na tej ścieżce istnieje krawędź g która bezpośrednio łączy się z drzewem T. i jej waga jest na pewno większa bądź równa od wagi e (gdyby była mniejsza, to wybralibyśmy g zamiast e)

Czyli $w(g) \geq w(e)$

Z drugiej strony $w(g) \leq w(e)$, bo zgodnie z założeniem T jest MST, czyli wtedy moglibyśmy wymienić krawędź g i dodać zamiast niej e

Czyli $w(g) = w(e)$

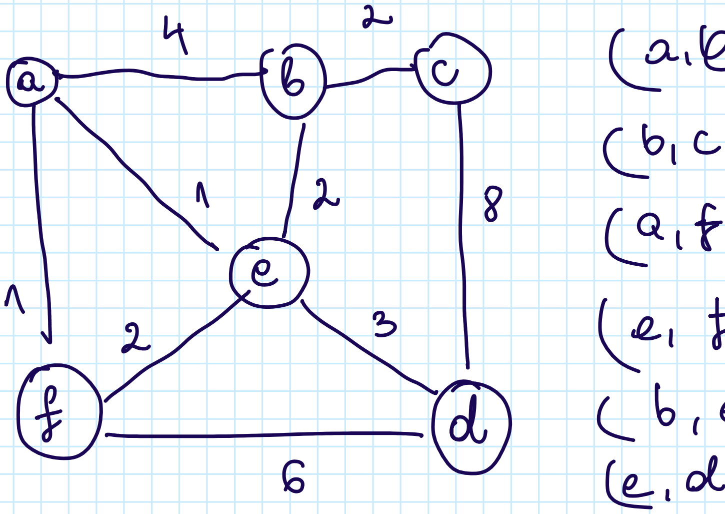
Czyli możemy wymienić krawędź g i dodać e i nadal otrzymamy MST otrzymaliśmy MST

$$O(|E| \log |E|)$$

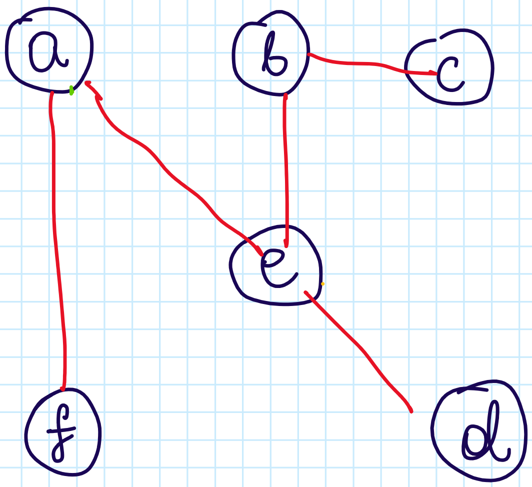
Algorytm Kruskala

czwartek, 30 maja 2024 18:41

Sortuje po wagach wszystkie krawędzie, następnie korzysta z union-find by utworzyć MST



(a, e) 1
(b, c) 2
(a, f) 2
(e, f) 2
(b, e) 2
(e, d) 3
(a, b) 4
(d, f) 6
(c, d) 8



Dowód rozbijmy na trzy części: Graf wygenerowany alg. Kruskala jest MST

1) Na pewno jest drzewem - nigdy nie dodajemy krawędzi, która może nam utworzyć cykl

2) spójne - bo algorytm nie zostawi dwóch

podgrafów, tylko same wybiera krawędź między nimi, a dokładniej tę o najmniejszej wadze

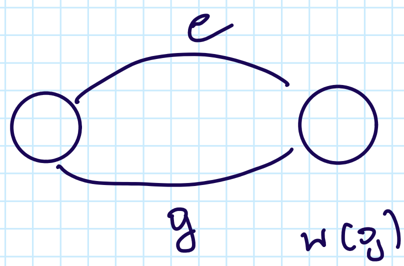
3) suma wag krawędzi wygenerowanego grafu jest minimalna

F - zbiór krawędzi w aktualnym kroku

T - MST

e - krawędź którą chcemy dodać

dowód jak w Primie identyczny



$w(g) \geq w(e)$

$w(g) \leq w(e)$

cyli $w(g) = w(e)$

$O(|E| \log |E|)$

Algorytm Boruvki

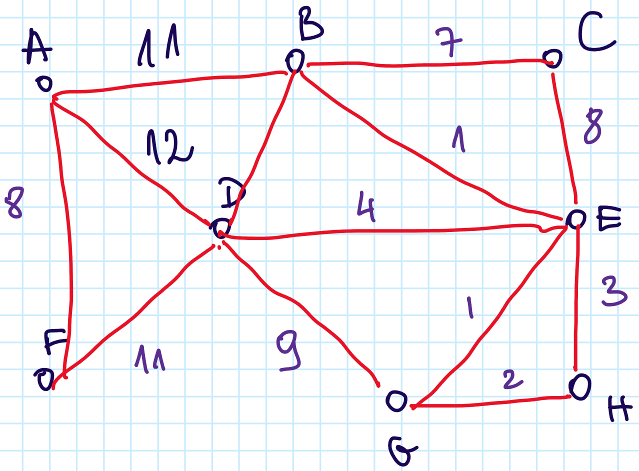
czwartek, 30 maja 2024 19:00

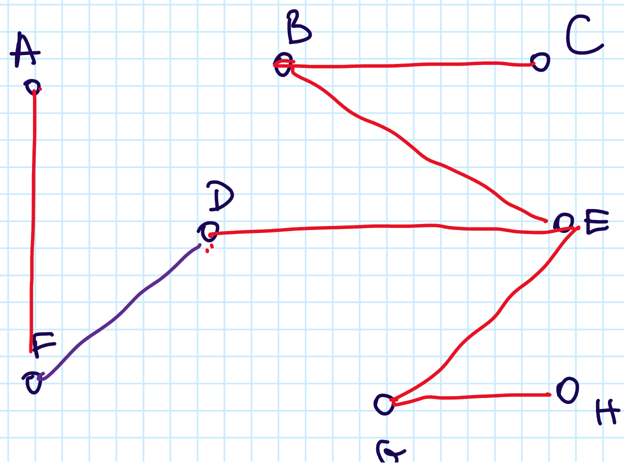
BORUVKA

- wyznacza MST dla grafu nieskierowanego ważonego, o ile jest on spójny

IDEA:

1) Dla każdego wierzchołka v w grafie G przejrzyj zbiór incydentnych z nim krawędzi. Dołóż najlżejszą z nich do rozwiązania.
2) Po tym etapie graf tymczasowego rozwiązania powinien zawierać nie więcej niż $\frac{|V|}{2}$ spójnych składowych. Utwórz graf G', w którym wierzchołki stanowiące spójne składowe zostaną ze sobą „sklejone"
3) Dopóki nie otrzymamy jednej spójnej składowej, wykonujemy kroki 1-2, za graf G podstawiając graf G'.



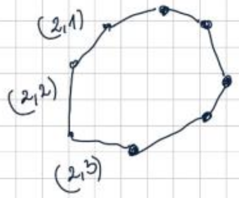


Krok I

Krok II

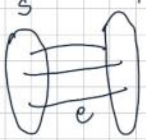
G

Złożoność $O(E \log|V|)$



Tw: *cut property* 0) krawędzie mają różne wagi $\Rightarrow$ istnieje 1 MST

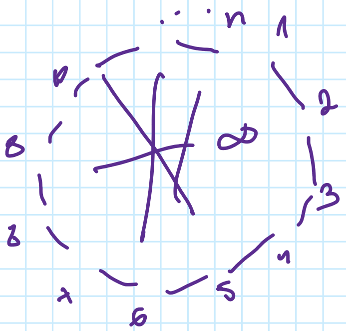
1) $e \in MST$ wtw, gdy $\exists_{S,T,\ S \cup T = V}$ e jest najtańszą krawędzią łączącą S,T

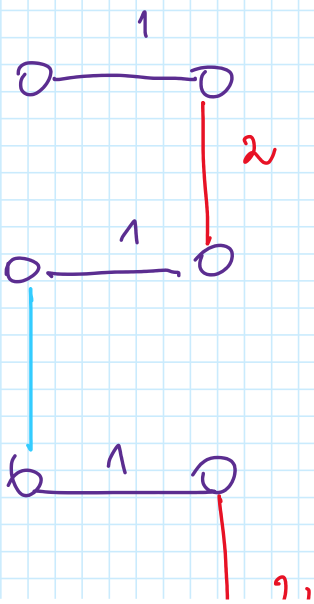


cycle property

2) $e \notin MST \iff \exists$ cykl e jest najdroższe na cyklu

3) z 1) i 2) wynika, że algorytm Boruvki jest poprawny



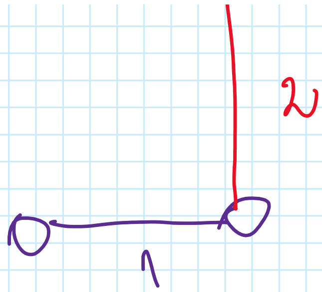


Faza I

Faza II

Faza III

1
vn



2
1