(1) G – graf spójny o różnych wagach

Pokaż, że zawiera tylko jedno minimalne drzewo rozpinające

Oczywiście każdy graf spójny zawiera przynajmniej jedno drzewo rozpinające.

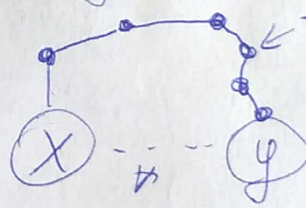
Pokażemy, że jeśli graf G ma różne wagi to zawiera dokładnie jedno MST

D-d (nie wprost): Załóżmy że istnieją dwa MST. Nazwijmy je T i T'. Niech e będzie krawędzią o najmniejszej wadze z T lub T' (nie obu!) (taka istnieje z zał. o różnych wagach).

Bez straty ogólności załóżmy że e to krawędź T.

Wtedy $T' \cup \{e\}$ zawiera cykl (bo istnieje ze spójności T' ścieżka łącząca wierzchołki na końcach e, różna od e)

Jeśli wszystkie inne krawędzie z $T' \cup \{e\}$ były w T to T zawierałby cykl, więc istnieje krawędź f nie należąca do T.



Z założenia o kosztach $c(f) > c(e)$

Możemy zamienić f na e w T' otrzymując drzewo rozpinające o mniejszej wadze, więc T' nie było MST

sprzeczność ↯

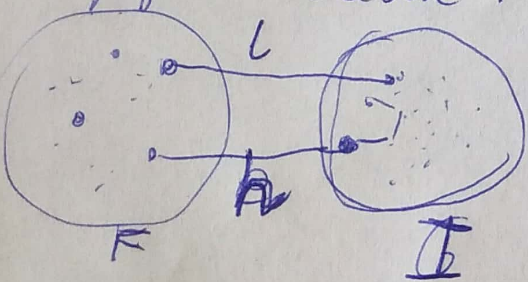
(2) G - graf spójny } założenie
T - MST G

$\forall C$ cyklu G drzewo T nie zawiera najcięższej krawędzi z C

Czyli że MST nie zawiera najcięższej krawędzi dla każdego cyklu

D-d (nie wprost): Weźmy dowolny cykl i jego najcięższą krawędź - h. Załóżmy nie wprost że h należy do drzewa T.

Możemy teraz usunąć krawędź h i otrzymamy podział na dwie spójne składowe F i I



Ale skoro h jest na cyklu w G to istnieje krawędź l która łączy składowe F i I, taka że $c(l) < c(h)$ bo h jest z zał. najcięższe.

Możemy wtedy zbudować drzewo $T' = (T \setminus \{h\}) \cup \{l\}$, które będzie minimalnym rozpinającym o wadze $c(T') < c(T)$ dlatego że $c(l) < c(h)$. Mamy sprzeczność z założeniem że T jest MST
sprzeczności

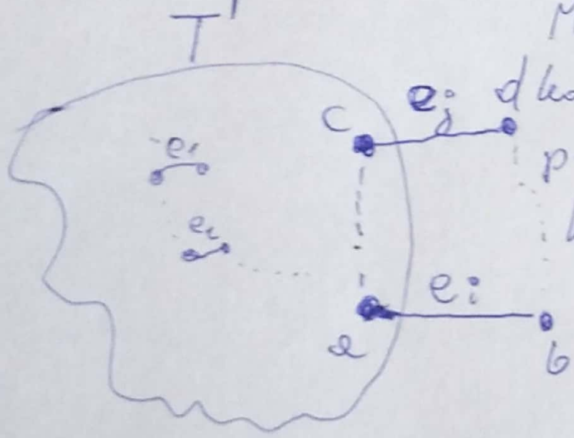
■ E

4) T: Algorytm Prima znajduje MST

D-d (nie wprost): Załóżmy że T nie jest MST

Ponumerujmy wybierane przez algorytm po kolei krawędzie $e_1, e_2, e_3, \dots, e_n$. Niech T' oznacza MST grafu.

Weźmy początkowe wierzchołki $e_1, e_2, \dots, e_{i-1}$ które należą do T' ~~i oznaczmy je jako M~~. Niech e_i - pierwsza krawędź dodana do T, a nie należąca do T'



Musi istnieć ścieżka w T' łącząca końce wierzchołka $e_i \to \{a, b\}$, powiedzmy że $e_j = \{c, d\}$ będzie krawędzią która jako pierwsza wychodzi z T' na tej ścieżce.

Teraz $T'' = T' + \{e_i\} - \{e_j\}$ jest drzewem rozpinającym

1) $c(e_i) = c(e_j)$ tzn. $c(T'') = c(T')$ czyli T'' jest MST, ale wtedy T' musiałby zawierać więcej krawędzi (bo e_j nie był wybrany) sprzeczność

2) $c(e_i) < c(e_j)$, Wtedy $c(T'') < c(T')$, ale T' miał być MST grafu, sprzeczność

3) $c(e_i) > c(e_j)$, Czyli algorytm wybierze krawędź o mniejszej wadze e_j w i-tym kroku, sprzeczność

Czyli T' nie jest minimalnym drzewem rozpinającym grafu ∎

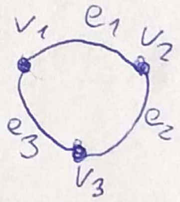
różne wagi → alg. Borůvki znajdzie MST
→ nie powstaje cykl

5 D-d (nie wprost):

Załóżmy że w którejś z iteracji powstaje cykl C.
Powiedzmy, że mamy n spójnych składowych i obecna iteracja generuje cykl. Z każdej ze spójnych składowych wybieramy krawędź o minimalnej wadze.

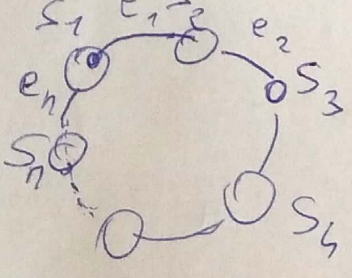
---

np (dla 3 ~~krawędzi~~ wierzchołków)



Bez straty ogólności $c(e_1) < c(e_2) < c(e_3)$
Wtedy algorytm nie wybierze e_3 do połączenia v_3 i v_1 bo
$c(e_3) > c(e_2)$ i $c(e_3) > c(e_1)$

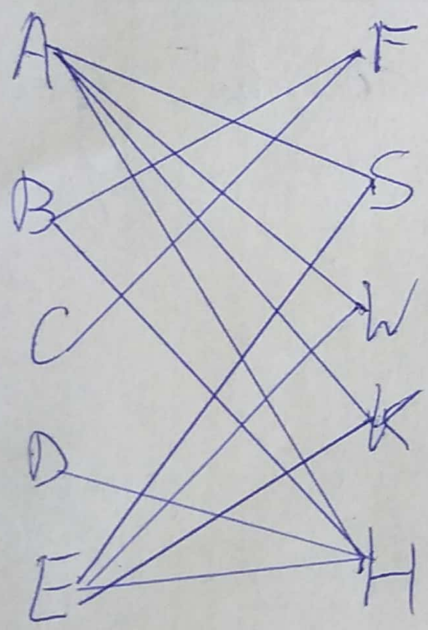
Podobnie dla n spójnych składowych. Niech e_i $i=1,2,\dots,n$ oznacza krawędzie ~~pomiędzy i-tą a i+1~~ na tym cyklu C.



Dobierzmy je tak, że
$c(e_1) < c(e_2) < c(e_3) < \dots < c(e_n)$
S_1 i S_n łączy e_n z działania algorytmu
więc $c(e_1) < c(e_2) < \dots < c(e_n) < c(e_1)$

Sprzeczność ↯

(10) Ludzie 'grają' instrumenty



Widzimy, że D musi grać na H (bo tylko na tym potrafi)

Podobnie C na F

Ale wtedy B który gra na H i F nie ma instrumentu.

Czyli nie da się dobrać składu

Inaczej ~~C, D, E~~ $\underbrace{B, D, C}_{X}$ mają do dyspozycji wyłącznie $\underbrace{F \text{ i } H}_{Y}$

$|N(X)| < |Y|$

Czyli nie zachodzi warunek Halla