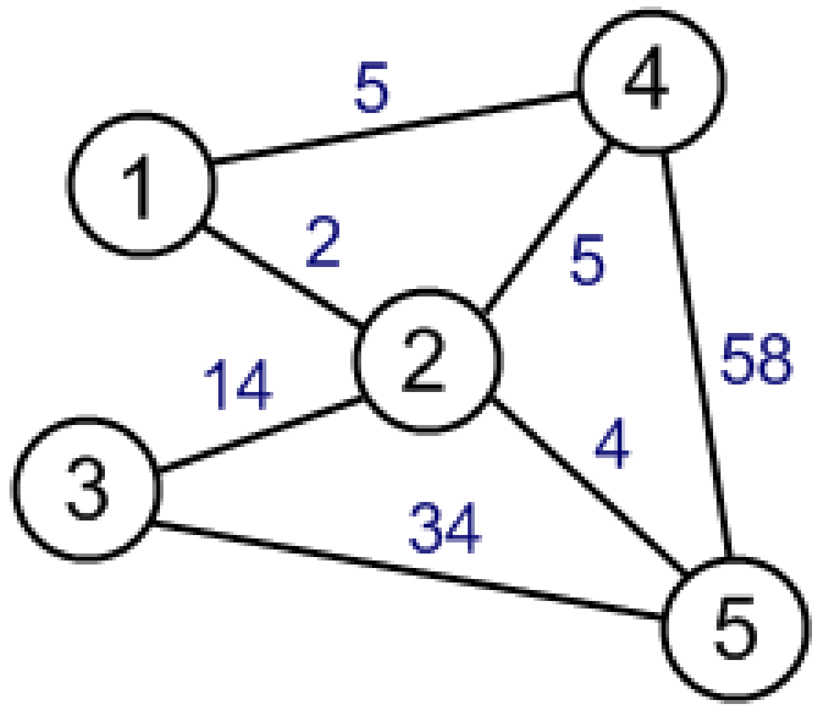
Dengiantis medis I

Vieslav Lapin

Svorinis medis

- Jei -grafo kiekvienai briaunai yra priskirtas svoris – realusis skaičius, tai grafas *G* vadinamas svoriniu grafu.



Uždavinys I

- Duotas svorinis jungusis grafas G=(V,U). Rasti trumpiausią dengiantį medį, t.y. dengiantį medį, kurio briaunų svorių suma būtų mažiausia tarp visų galimų dengiančių medžių.

Uždavinys II

- Optimizavimo uždavinys
- Uždavinio tikslūs sprendimo algoritmai yra efektyvūs
- Du šio uždavinio sprendimo metodai:
  - Kraskalo ($O(m \log m)$ operacijų)
  - Primo ($O(n^2)$ operacijų)

Kraskalo metodas

- ***Teorema***. Tarkime, kad G=(V,U) jungusis pilnasis grafas ir visų jo briaunų ilgiai (svoriai) skirtingi.
- Tada egzistuoja vienintelis trumpiausias dengiantis medis, kuris konstruojamas taip: “iš likusių grafo *G* briaunų randame trumpiausią briauną ir ją įtraukiame į medį, jei ši briauna neiššaukia ciklo su anksčiau paimtom medžio briaunom”.

Įrodymas I

- Prielaida. $H=(V,U_H)$ yra dengiantis medis, sukonstruotas teoremoje nurodytu metodu., bet ne trumpiausias.
- Tai reiškia , kad egzistuoja $T=(V,U_T)$ ir $U_T \neq U_H$.
- u_k – pirmoji viršūnė iš U_H, nepriklausanti $U_{T.}$

Įrodymas II

- Panagrinėkime briaunų aibę $U_T \cup \{u_k\}$.
- aibė turės vienintelį ciklą, kuriame bus bent viena briauna $u_0 \notin U_H$.
- Priešingu atveju medyje (V, U_H) yra ciklas.

Įrodymas III

- Grafas (V,W), $W = U_T \cup \{u_k\} \setminus \{u_0\}$
- Medis, nes jis turi n-1 briauną ir neturi ciklų.
- $l(u_0) > l(u_k)$
- medžio (V,W) briaunų ilgių suma yra mažesnė paties trumpiausio dengiančio medžio $T = (V, U_T)$ briaunų ilgių suma.
- Prieštara. Medis $H = (V, U_H)$ yra trumpiausias dengiantis medis.
- Teorema galioja, jei grafas yra jungusis, bet nepilnasis, ir kai kurių briaunų ilgiai yra lygūs.

Kraskalo metodo algoritmas

- Duota:
  - n – grafo viršūnių skaičius,
  - m – grafo briaunų skaičius,
  - b [1..2, 1..m] – jungiojo grafo briaunų matrica,
  - c [1..m] –briaunų ilgių masyvas:
  - c [j] yra briaunos (b [1, j], b [2, j]) ilgis.
- Rasti:
  - Trumpiausią dengiantį medį, t.y. medžio briaunų matricą
    t [1..2, 1..n-1] ir jų ilgių masyvą d [1..n-1].

- Vidiniai darbo masyvai:
- *s* [1..*n*] – viršūnių jungiųjų komponenčių masyvas,
- *p* [1..*m*] – požymių masyvas;

$$p[i] = \begin{cases} 0, & \text{jei } i \text{ - oji briauna nenagrinėta,} \\ 1, & \text{priešingu atveju.} \end{cases}$$

- Žodžiai “briauna nagrinėta” reiškia, kad ji yra arba medžio briauna, arba ją bandėme įtraukti į medį.

Procedūra kraskalas

- *var i, j, k, l, u, v, x, y* : *integer*;
  - *sum, min* : *real*;
  - *s, p* : *mas*;
- *begin*
  - *for i* := 1 *to n do s* [*i*] := *i*;
  - *sum* := 1;
  - *for i* := 1 *to m do*
  - *begin*
    - *p* [*i*] := 0;
    - *sum* :=*sum* + *c* [*i*];
  - *end*;
  - *k* := 0; { *Į medį įtrauktų briaunų skaičius* }

Procedūra kraskalas I

- *while k* < *n* – 1 *do*
- *Begin* { *Rasti trumpiausią briauną iš likusių* }
  - *min* := *sum*;
  - *for i* := 1 *to m do*
    - *if* (*p* [*i*] = 0) *and* (*min* > *c* [*i*]) *then*
    - *begin*
      - » *min* := *c* [*i*];
      - » *l* := *i*;
    - *end*;
  - *if min = sum then*
  - *begin*
    - *writeln* ('*Grafas – nejungusis*');
    - *exit*;
  - *end*;

Procedūra kraskalas II

- *p* [*l*] := 1; { *l-oji briauna – nagrinėta* }
- *u* := *b* [1, *l*];
- *v* := *b* [2, *l*];
- *if s* [*u*] <> *s* [*v*] *then*
- *Begin*
  - *k* := *k* + 1;
  - *t* [1, *k*] := *u*; *t* [2 ,*k*] := *v*;
  - *d* [*k*] := *c* [*l*];
  - *x* := *s* [*u*]; *y* := *s* [*v*];
  - *for i* := 1 *to n do*
  - *if s* [*i*] = *y then s* [*i*] := *x*;
- *end*;

– *end*;

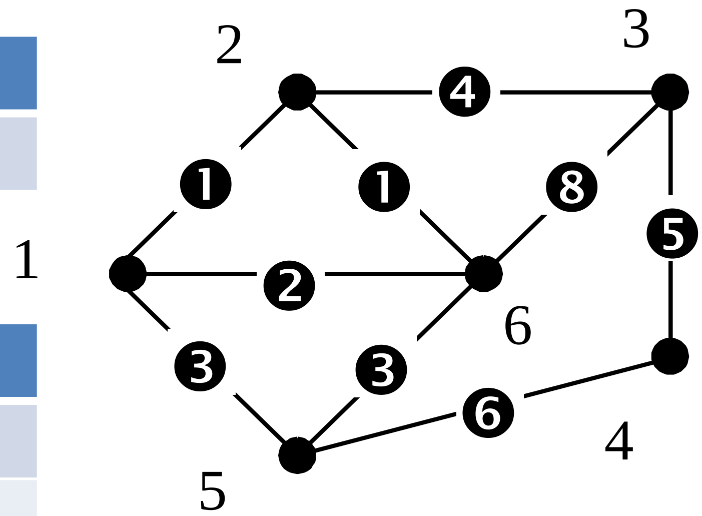
- *end*;

Pavyzdys

	1	2	3	4	5	6	7	8	9
b 1	1	1	1	2	2	3	3	4	5
b 2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	0	0	0	0	0	0	0	0	0

	1	2	3	4	5	6
s	1	2	3	4	5	6

	1	2	3	4	5
t1					
t2					
d					



a)

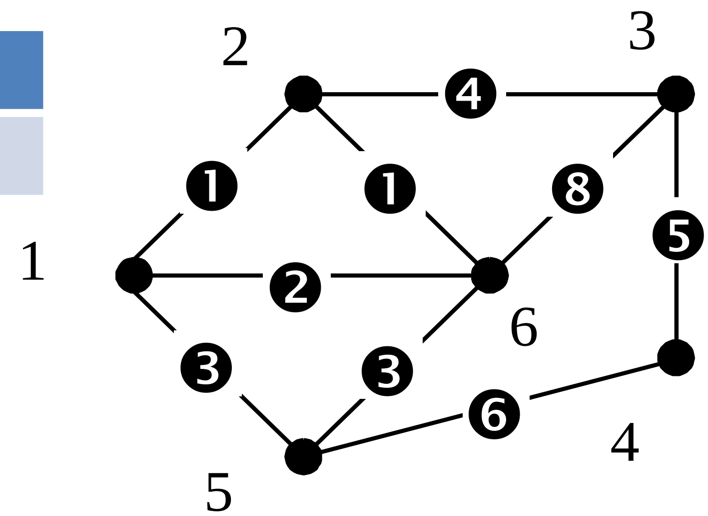
1 iteracija

- Min:=1; l=1;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	0	0	0	0	0	0	0	0

	1	2	3	4	5	6
s	1	1	3	4	5	6

	1	2	3	4	5
t1	1				
t2	2				
d	1				



a)

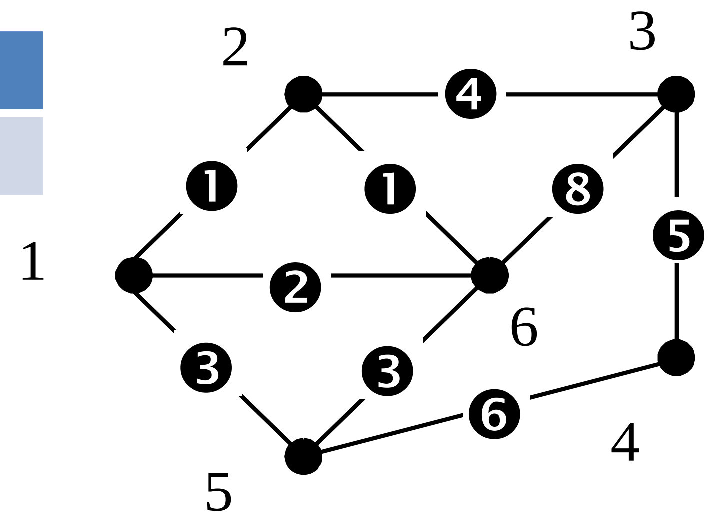
2 iteracija

- Min:=1; l=5;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	0	0	0	1	0	0	0	0

	1	2	3	4	5	6
s	1	1	3	4	5	1

	1	2	3	4	5
t1	1	2			
t2	2	6			
d	1	3			



a)

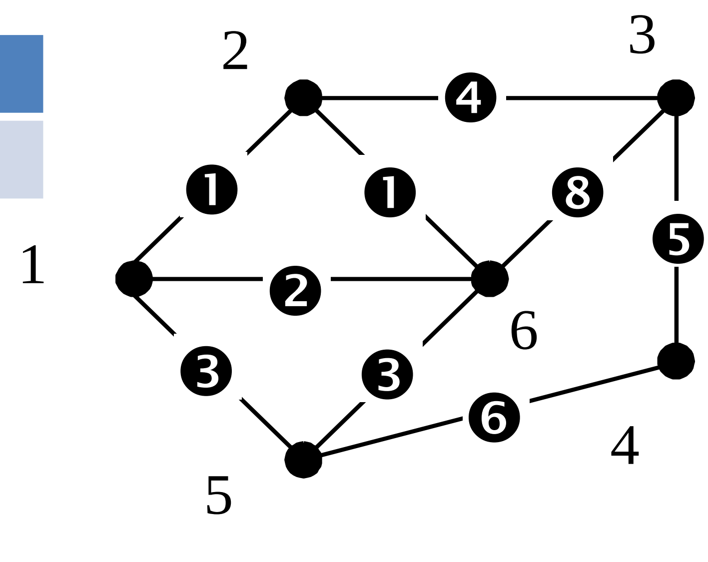
3 iteracija

- Min:=2; l=2;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	1	0	0	1	0	0	0	0

	1	2	3	4	5	6
s	1	1	3	4	5	1

	1	2	3	4	5
t1	1	2			
t2	2	6			
d	1	3			



a)

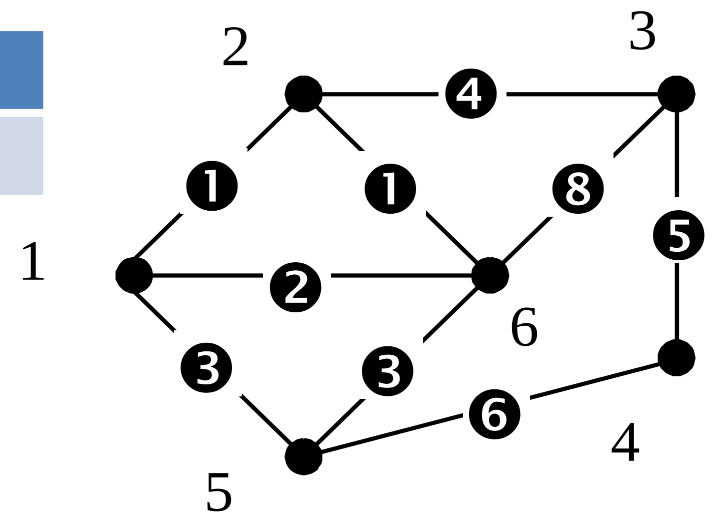
4 iteracija

- Min:=3; l=3;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	1	1	0	1	0	0	0	0

	1	2	3	4	5	6
s	1	1	3	4	1	1

	1	2	3	4	5
t1	1	2	1		
t2	2	6	5		
d	1	3	3		



a)

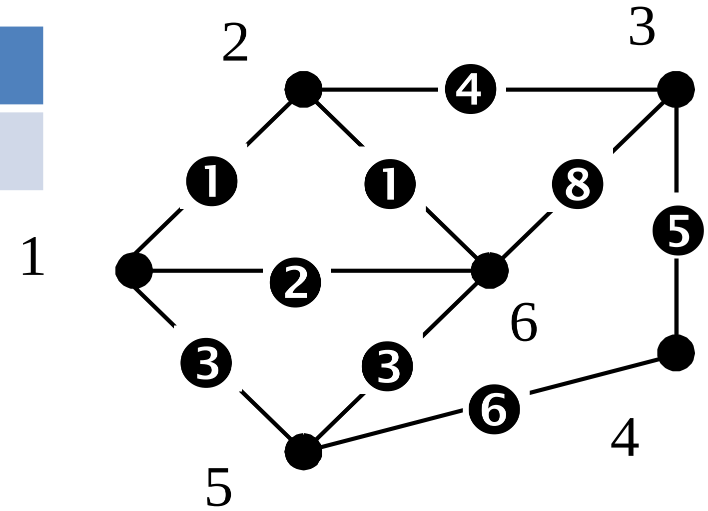
5 iteracija

- Min:=3; l=9;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	1	1	0	1	0	0	0	1

	1	2	3	4	5	6
s	1	1	3	4	1	1

	1	2	3	4	5
t1	1	2	1		
t2	2	6	5		
d	1	3	3		



a)

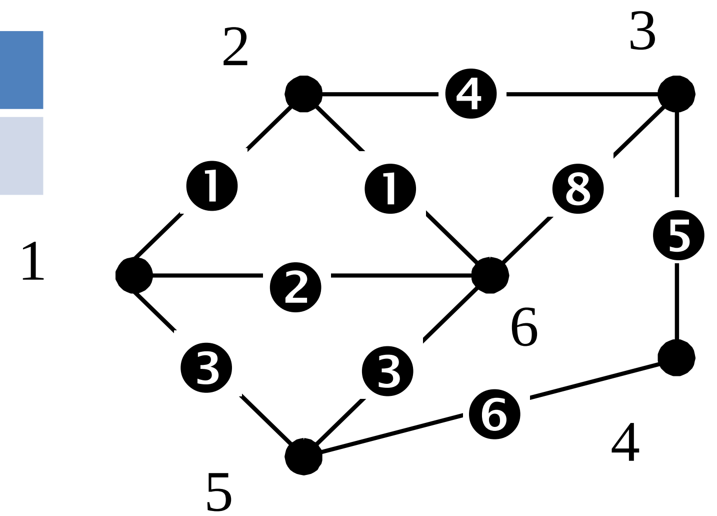
6 iteracija

- Min:=4; l=4;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	1	1	1	1	0	0	0	1

	1	2	3	4	5	6
s	1	1	1	4	1	1

	1	2	3	4	5
t1	1	2	1	2	
t2	2	6	5	3	
d	1	3	3	4	



a)

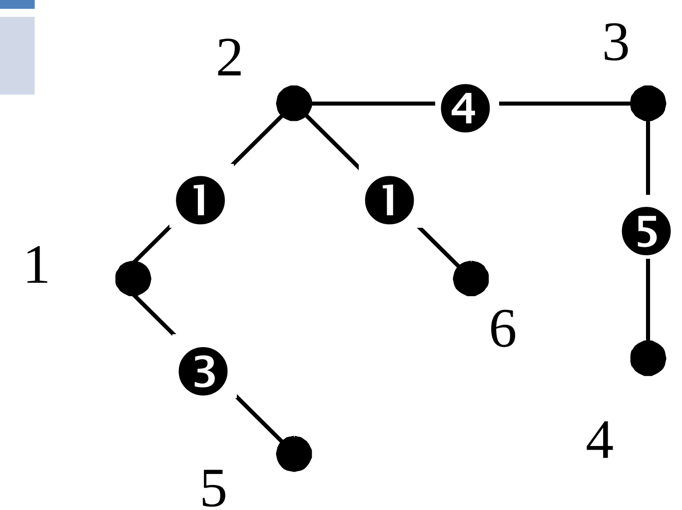
7 iteracija

- Min:=4; l=4;

	1	2	3	4	5	6	7	8	9
b1	1	1	1	2	2	3	3	4	5
b2	2	6	5	3	6	4	6	5	6
c	1	2	3	4	1	5	8	6	3
p	1	1	1	1	1	1	0	0	1

	1	2	3	4	5	6
s	1	1	1	1	1	1

	1	2	3	4	5
t1	1	2	1	2	3
t2	2	6	5	3	4
d	1	3	3	4	5



KLAUSIMAI?