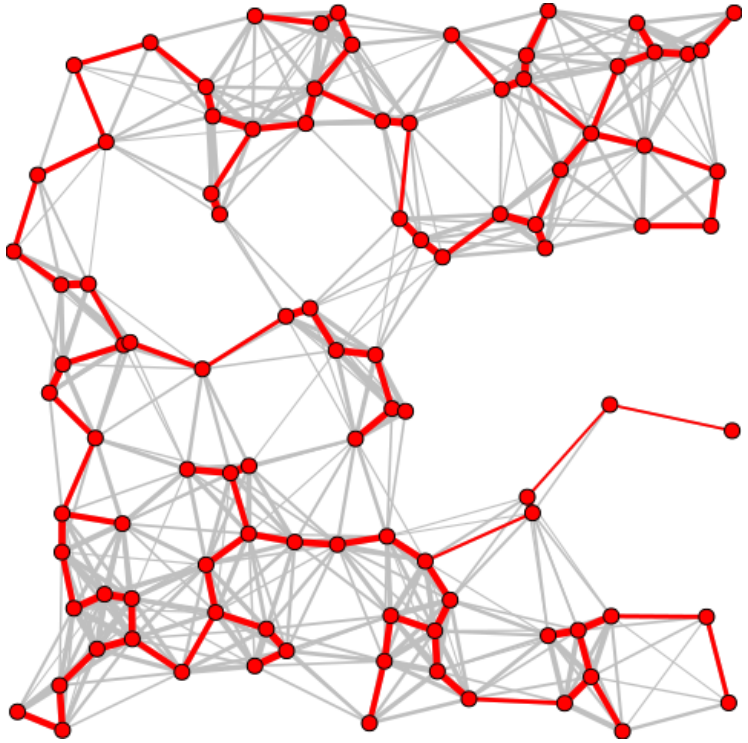


Algoritmer og Datastrukturer

Minimum udspændende træer (MST)
[CLRS, kapitel 23]

Kontaimu p. prof. Dr V. Bydžovskému v Hluboké 5.x.26. O. Borůvka.

ZVLÁŠTNÍ OTISK Z ČASOPISU „ELEKTROTECHNICKÝ OBZOR“
Roč. 15. Čís. 10. Praha III., Cihelná 102. 5. března 1926.

Dr. OTAKAR BORŮVKA:

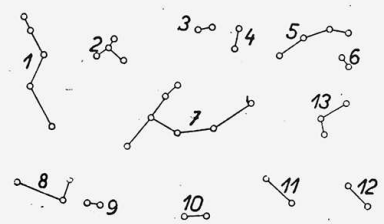
Příspěvek k řešení otázky ekonomické stavby elektrovodných sítí.

Ve své práci „O jistém problému minimálním“ *) odvodil jsem obecnou větu, jíž jest ve zvláštním případě řešena tato úloha:

V rovině (v prostoru) jest dáno n bodů, jejichž vzájemné vzdálenosti jsou vesměs různé. Jest je spojiti sítí tak, aby:

1. každé dva body byly spojeny buď přímo anebo prostřednictvím jiných,
2. celková délka sítě byla co nejmenší.

Obr. 1.



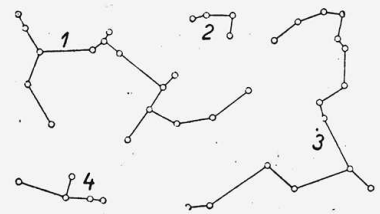
Obr. 2.

Jest zřejmé, že řešení této úlohy může míti v elektrotechnické praksi při návrzích plánů elektrovodných sítí jistou důležitost; z toho důvodu je zde stručně na příkladě vyložím. Čtenáře, jenž by se o věc blíže zajímal, odkazuji na citované pojednání.

*) Vyjde v nejbližší době v Pracích Moravské přírodovědecké společnosti.

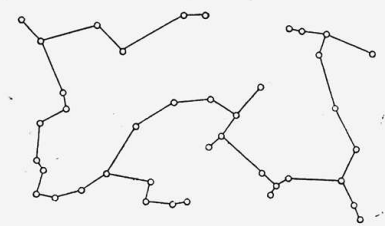
Řešení úlohy provedu v případě 40 bodů daných v obr. 1.

Každý z daných bodů spojím s bodem nejbližším. Tedy na př. bod 1 s bodem 2, bod 2 s bodem 3, bod 3 s bodem 4 (bod 4 s bodem 3), bod 5 s bodem 2, bod 6 s bodem 5, bod 7 s bodem 6, bod 8 s bodem 9, (bod 9 s bodem 8) atd. Obdržím řadu polygonálních tahů 1, 2, ... 13 (obr. 2).



Obr. 3.

Každý z nich spojím nejkratším způsobem s tahem nejbližším. Tedy na př. 1 s tahem 2, (tah 2 s tahem 1), tah 3 s tahem 4, (tah 4 s tahem 3) atd. Obdržím řadu polygonálních tahů 1, 2, ... 4 (obr. 3).



Obr. 4.

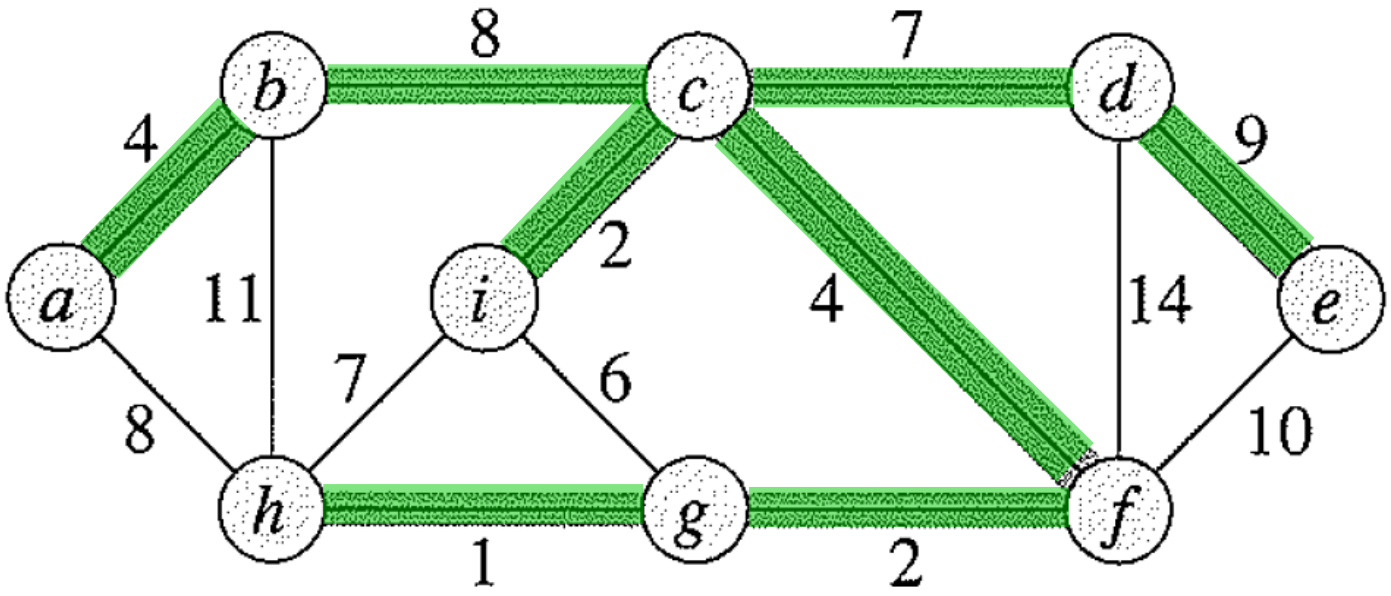
Orazec převrácen.

Každý z nich spojím nejkratším způsobem s tahem nejbližším. Tedy tah 1 s tahem 3, tah 2 s tahem 3 (tah 3 s tahem 1), tah 4 s tahem 1. Obdržím konečně jediný polygonální tah (obr. 4), jenž řeší danou úlohu.

Matematický ústav Masarykovy university v Brně,
v lednu 1926.

Otakar Borůvka. *Příspěvek k řešení otázky ekonomické stavby elektrovodních sítí*. Elektronický Obzor, 15:153–154 (1926)

Minimum Udspændende Træ (MST)

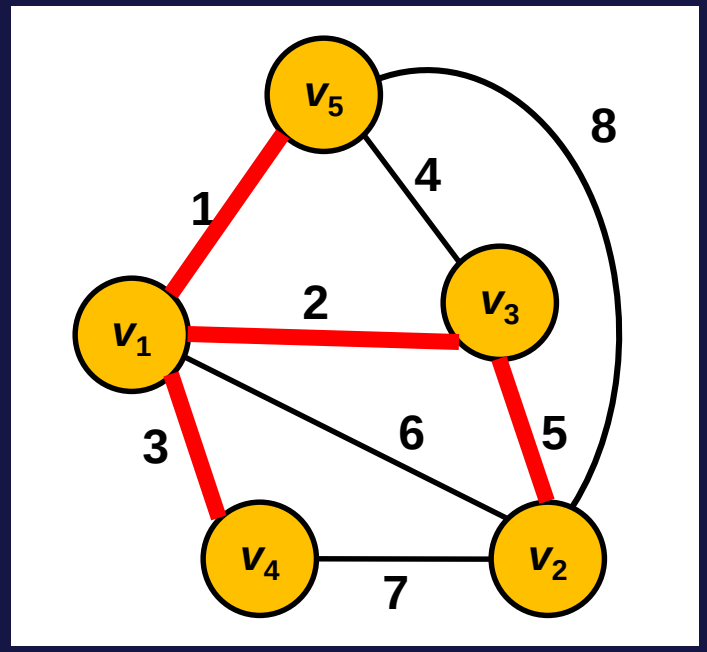


Problem

Find et udspændende træ for en sammenhængende uorienteret vægtet graf således at summen af kanterne er mindst mulig

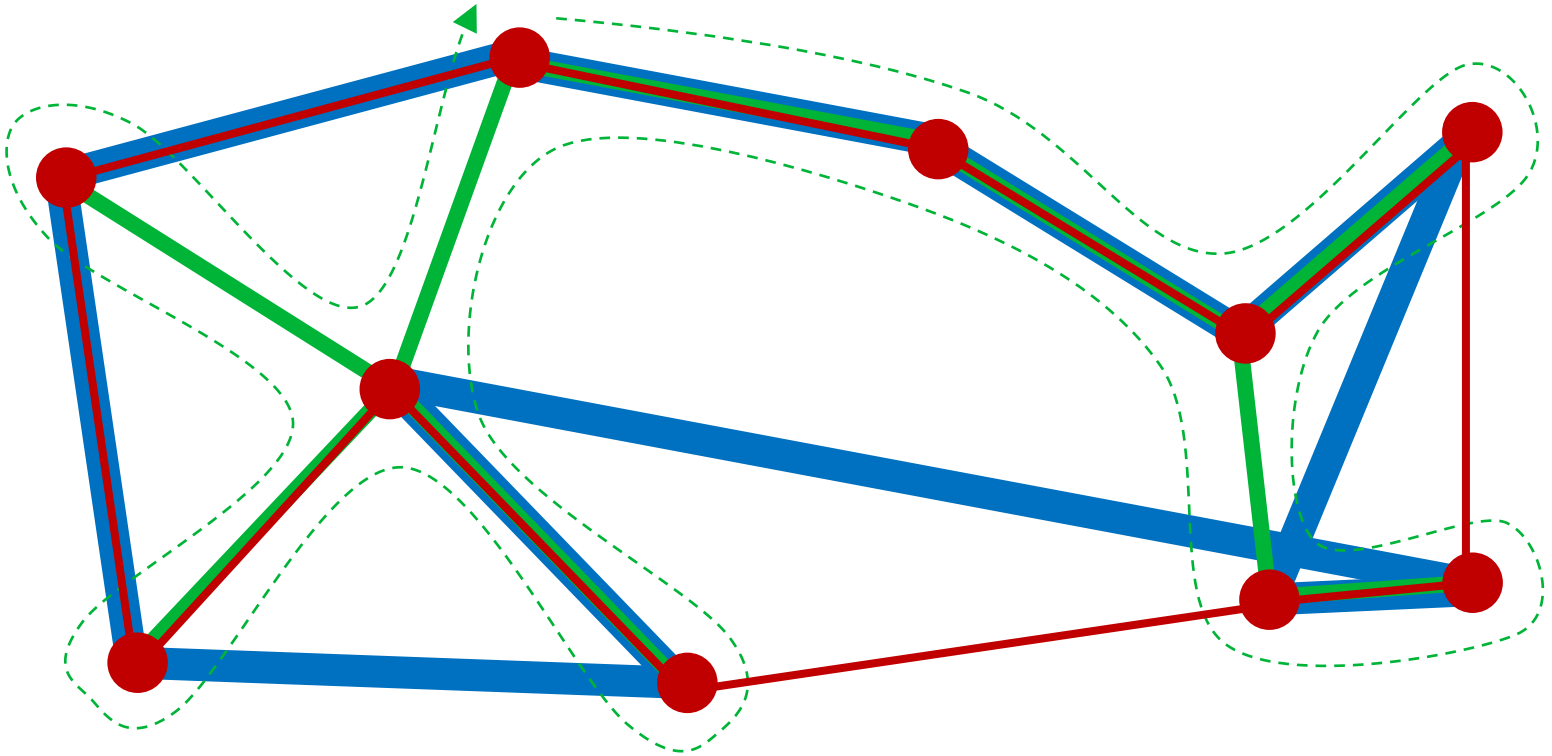
Vægten af et minimum udspændende træ?

a) 10
b) 11
c) 12
d) 13
e) 14
f) 15
g) 16
h) Ved ikke



IKKE PENSUM

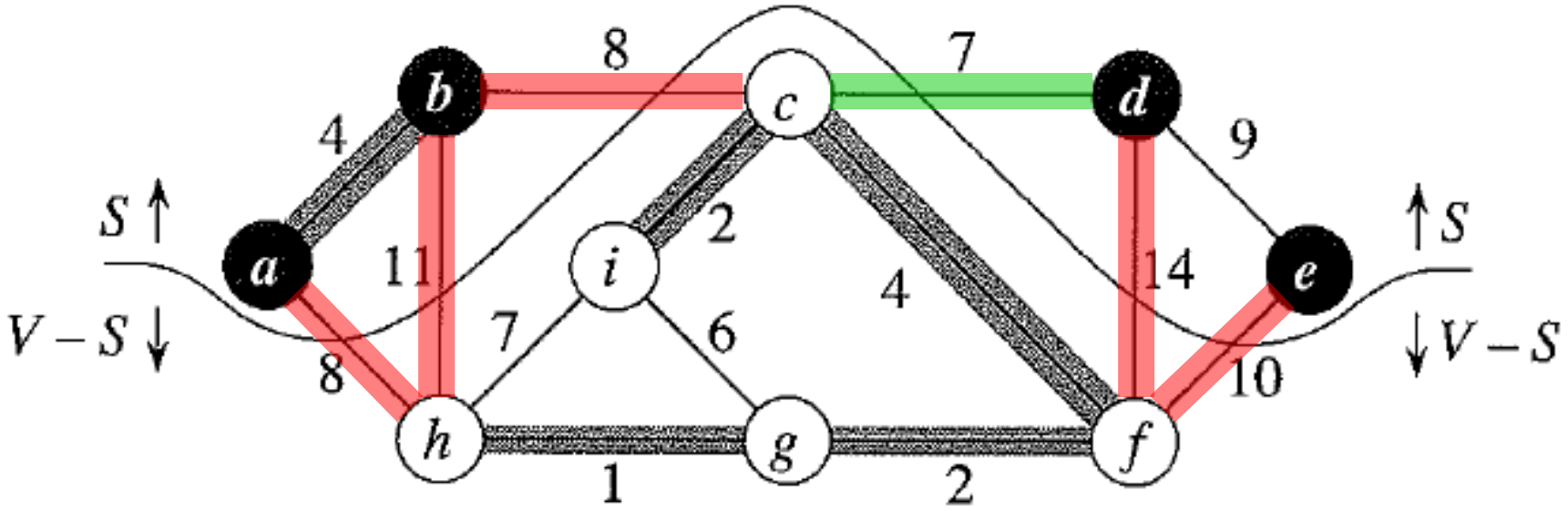
En anvendelse af MST :
(Euklidisk) Korteste Hamiltonske Cykel



Sætning : Touren fundet vha. et **MST**
er en 2-approximation til en **korteste cykel**

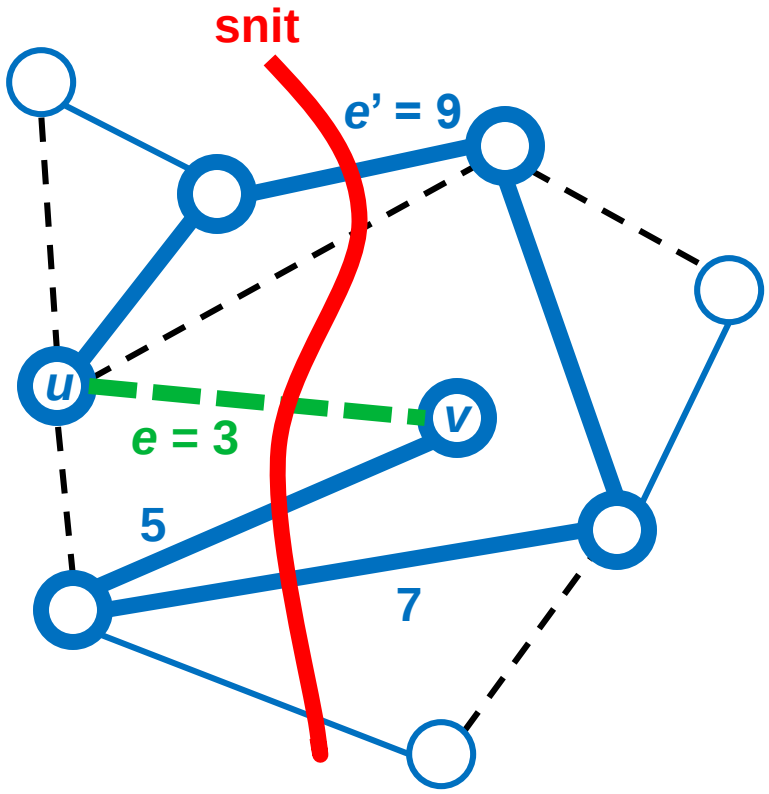
(|**Touren**| ≤ 2·|**MST**| og |**MST**| ≤ |**korteste cykel**|)

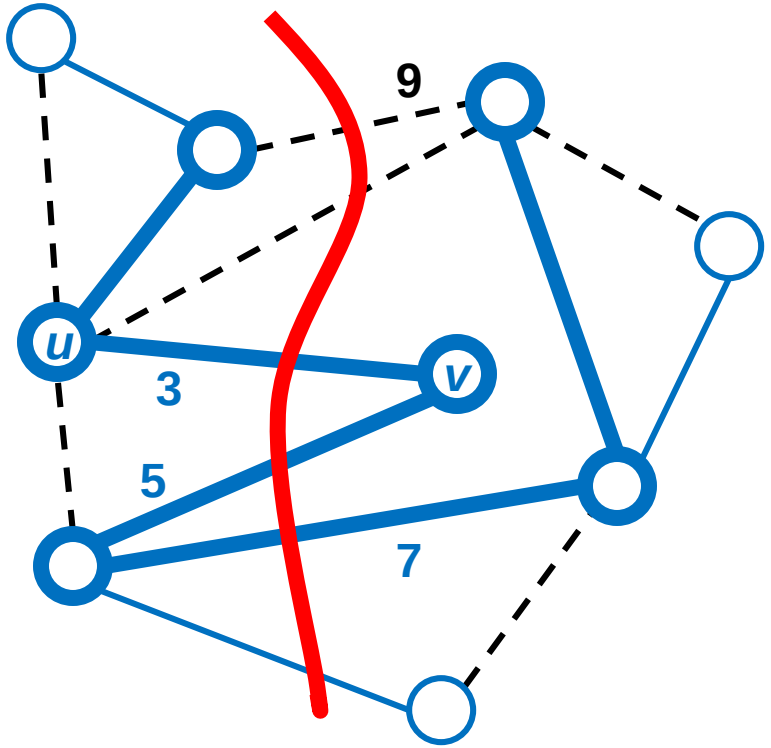
Minimum Udspændende Træer: Snit



Sætning

Hvis alle vægte er forskellige, så gælder der for *ethvert* **snit (*S*,*V*-*S*)** at den **letteste kant** der krydser snittet er med i et minimum udspændende træ





Antag modsætningsvis et MST med et snit hvor mindste kant *e* ikke er med i MST

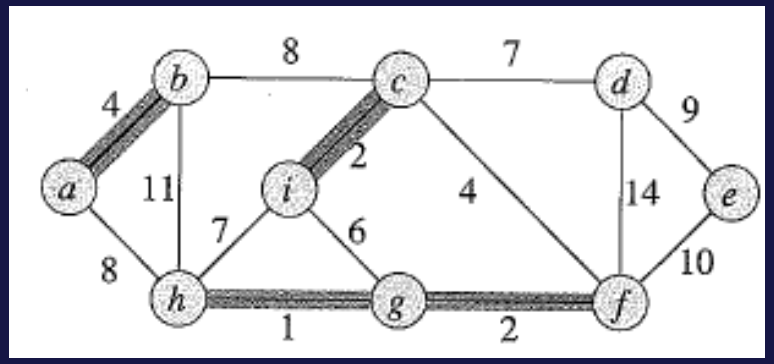
Nyt udspændende træ med mindre vægt

Sætning

Hvis alle vægte er forskellige, så gælder der for *ethvert* **snit (*S*,*V*-*S*)** at den **letteste kant** der krydser snittet er med i et minimum udspændende træ

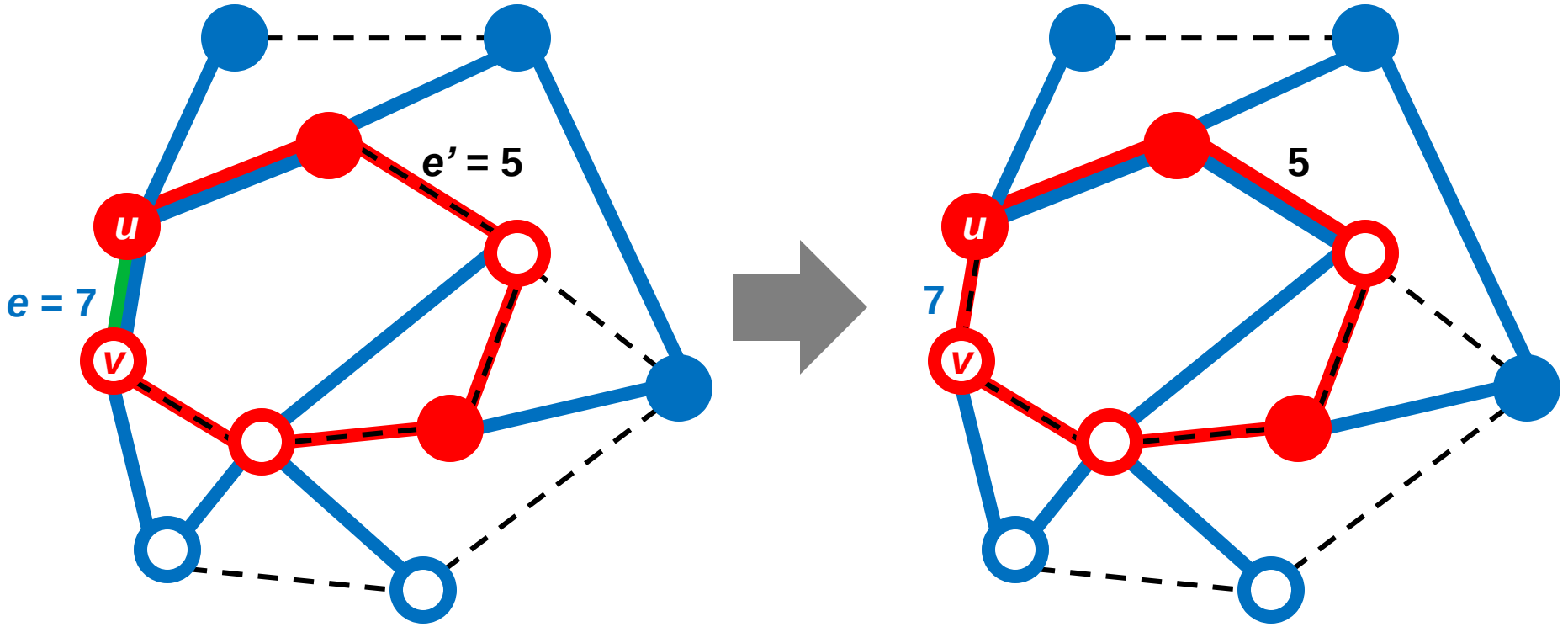
Hvis man for en sammenhængende graf endnu ikke har fundet tilstrækkeligt mange kanter til at de udgør et MST, kan man så altid anvende snit sætningen ?

a) Ja
b) Nej
c) Ved ikke



Sætning
Hvis alle vægte er forskellige, så gælder der for *ethvert* **snit (*S*,*V*-*S*)** at den **letteste kant** der krydser snittet er med i et minimum udspændende træ

Sætning
Hvis alle vægte er forskellige, så gælder der for *enhver* **cykel** at den **tungeste kant i cyklen** ikke er med i et minimum udspændende træ



Antag modsætningsvis et MST og cykel
hvor tungeste kant *e* er med i MST

Nyt udspændende træ
med mindre vægt

Minimum Udspændende Træer: Grådig generel algoritme

```
GENERIC-MST(G, w)
1  A = ∅
2  while A does not form a spanning tree
3      find an edge (u, v) that is safe for A
4      A = A ∪ {(u, v)}
5  return A
```

En letteste kant over et snit (som ikke allerede indeholder kanter fra *A*)

Kruskall's Algoritme

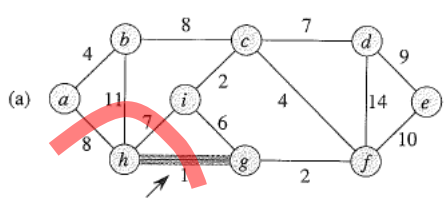
```
MST-KRUSKAL(G, w)
1  A = ∅
2  for each vertex v ∈ G.V
3       MAKE-SET(v)
4  sort the edges of G.E into nondecreasing order by weight w
5  for each edge (u, v) ∈ G.E, taken in nondecreasing order by weight
6       if FIND-SET(u) ≠ FIND-SET(v)
7            A = A ∪ {(u, v)}
8            UNION(u, v)
9  return A
```

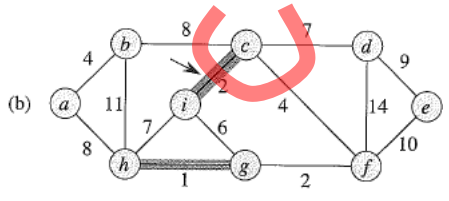
flaskehals i algoritmen

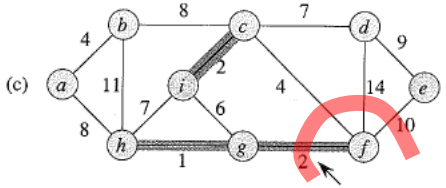
Union-Find datastruktur

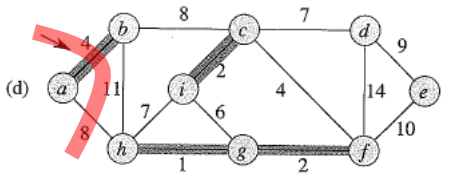
Tid $O(m \cdot \log n)$

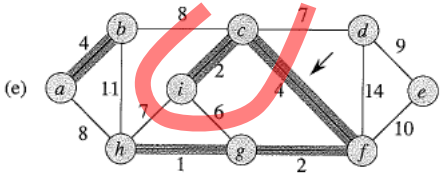
Kruskall's Algoritme: Eksempel

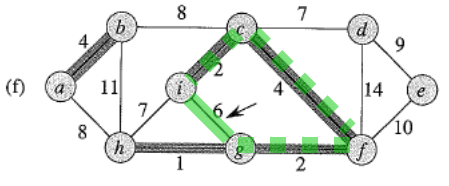


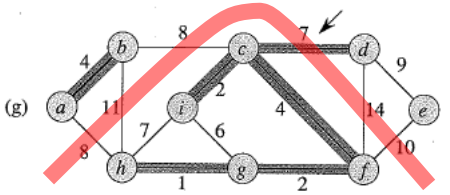


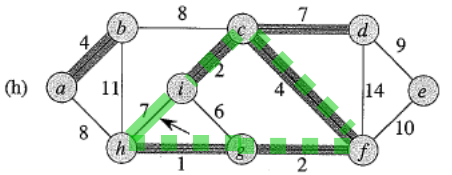


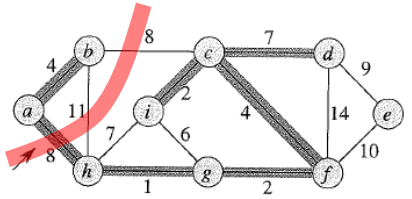


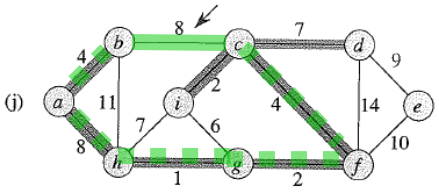


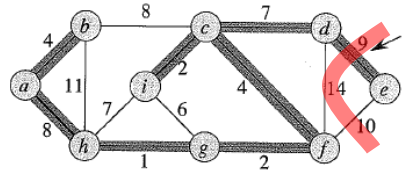


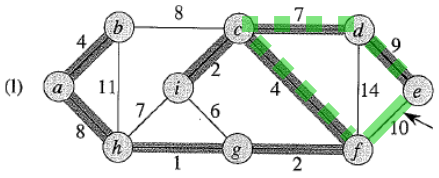


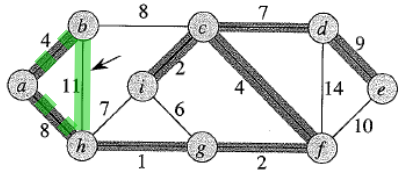


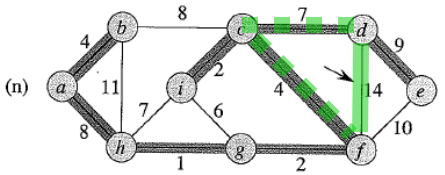










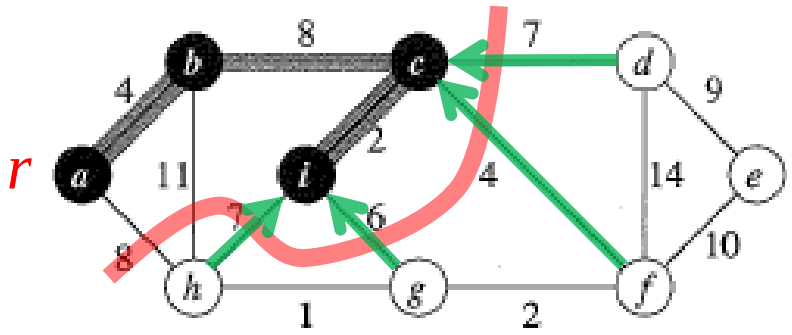


Kantene betrages efter stigende vægte (angivet med ↗)

For hver kant er angivet snittet hvor den er en letteste kant eller cyklen hvor den er en tungeste kant

Prim's Algoritme

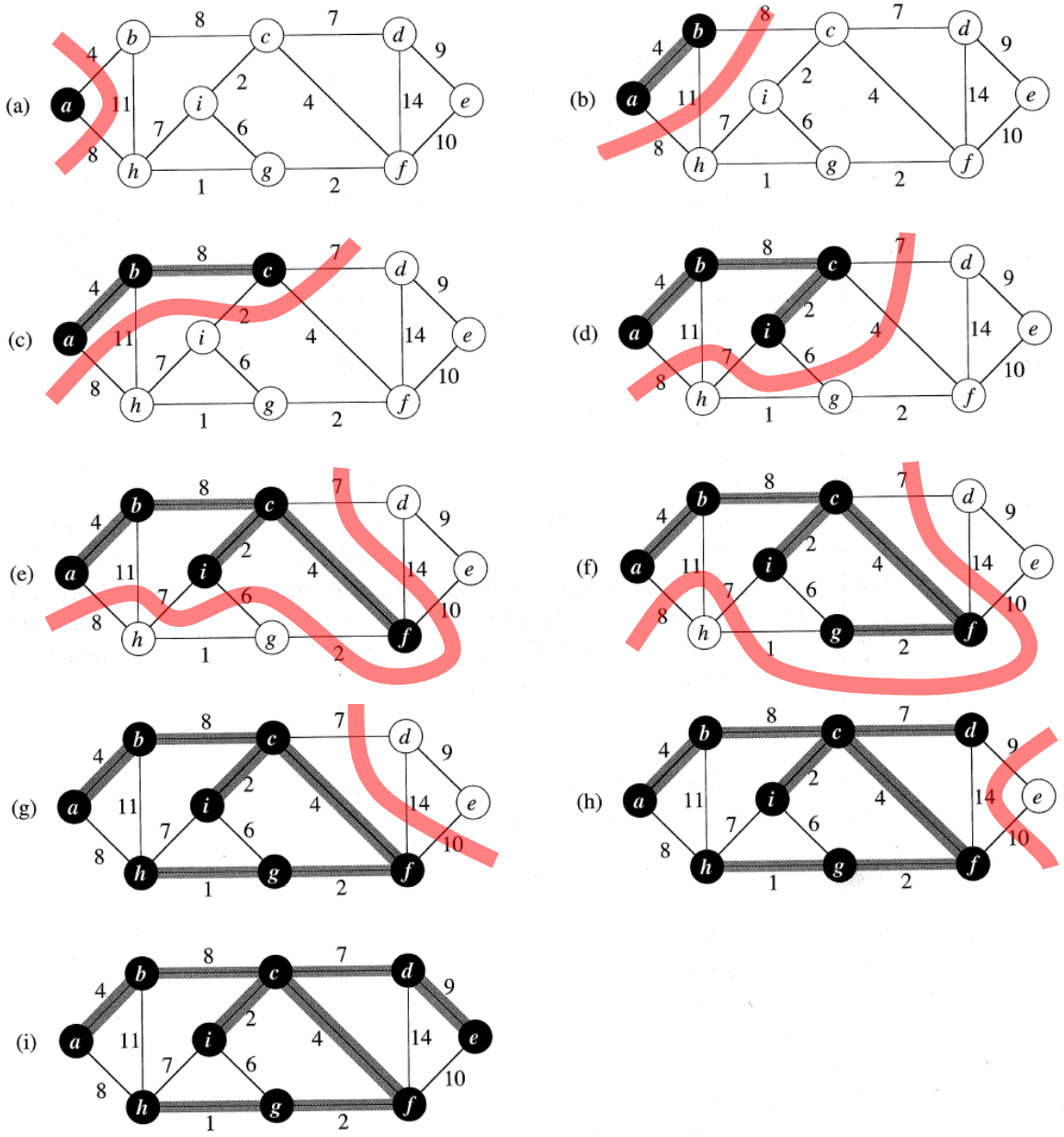
```
MST-PRIM(G, w, r)
 1  for each u ∈ G.V
 2      u.key = ∞
 3      u.π = NIL
 4  r.key = 0
 5  Q = G.V
 6  while Q ≠ ∅
 7      u = EXTRACT-MIN(Q)
 8      for each v ∈ G.Adj[u]
 9          if v ∈ Q and w(u, v) < v.key
10              v.π = u
11              v.key = w(u, v)
```



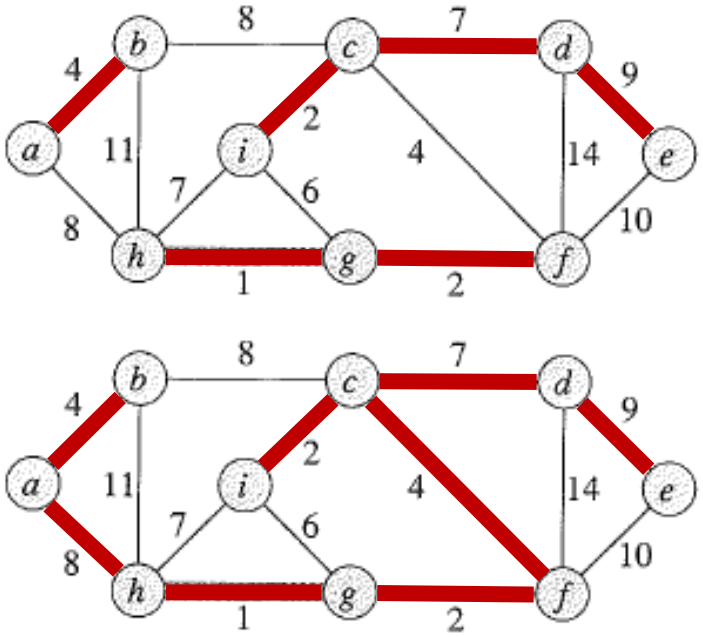
flaskehals i algoritmen
- prioritetskø

Tid $O(m \cdot \log n)$

Prim's Algoritme: Eksempel



Borůvka's Algoritme



MST-BORUVKA(G, w)
1 $A = \emptyset$
2 **for** each vertex $v \in G.V$
3 MAKE-SET(v)
3 **while** $|A| < n - 1$
4 $B = \emptyset$
5 **for** each set S
6 let (u, v) be lightest edge incident to S, i.e. $u \in S$ and $v \notin S$
7 **if** such (u, v) exists
8 $B = B \cup \{(u, v)\}$
8 **for** each edge $(u, v) \in B$
9 **if** FIND-SET(u) $\neq$ FIND-SET(v)
10 $A = A \cup \{(u, v)\}$
11 UNION(u, v)
12 **return** A

Efter *i* iterationer af while **har hver mængde størrelse ≥ 2^i (eller er udspændende)**

Tid $O(m \cdot \log n)$

Eksamensopgave 3(c)
august 2005

Spørgsmål c: Beskriv en algoritme, der givet en vægtet graf G og en kant e i grafen, afgør om e er indeholdt i et minimum udspændende træ for G. Det antages at alle vægtene er forskellige. Udførselstiden for algoritmen skal være $O(m)$, hvor m er antal kanter i grafen. □

Minimum Udspænding Træer

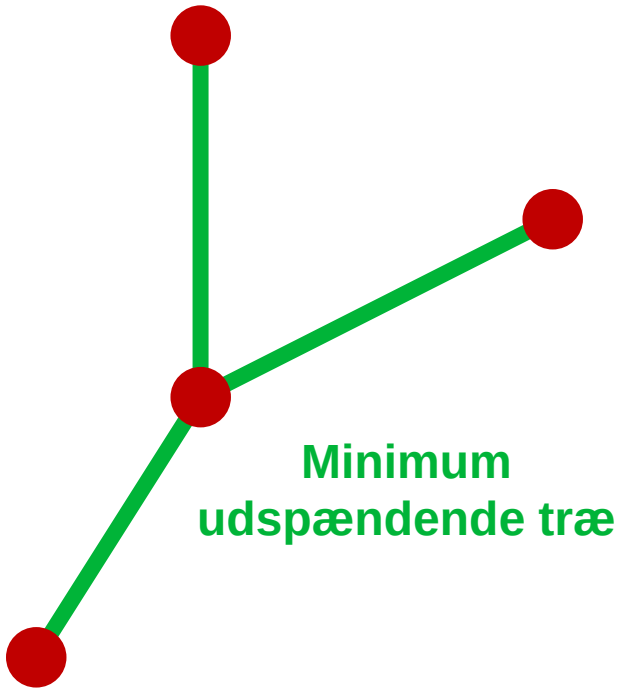
Kruskall (1956) (mange træer; sorterer kanterne)	$O(m \cdot \log n)$
Prim (1930) (et træ; prioritetskø over naboknuder)	$O(m \cdot \log n)$
	$O(m+n \cdot \log n)$ (Fibonnaci heaps [CLRS, kapitel 19] (1984))
Borůvka (1926) (mange træer samtidigt; kontraktion)	$O(m \cdot \log n)$
Fredman, Tarjan (1984) (Borůvka (1926) + Fibonnaci heaps)	$O(m \cdot \log^* n)$
Chazelle (1997)	$O(m \cdot \alpha(m,n))$
Pettie, Ramachandran (2000)	? (men optimal deterministisk sammenligningsbaseret)
Karger, Klein, Tarjan (1995) (Randomiseret) Fredman, Willard (1994) (RAM)	$O(m)$

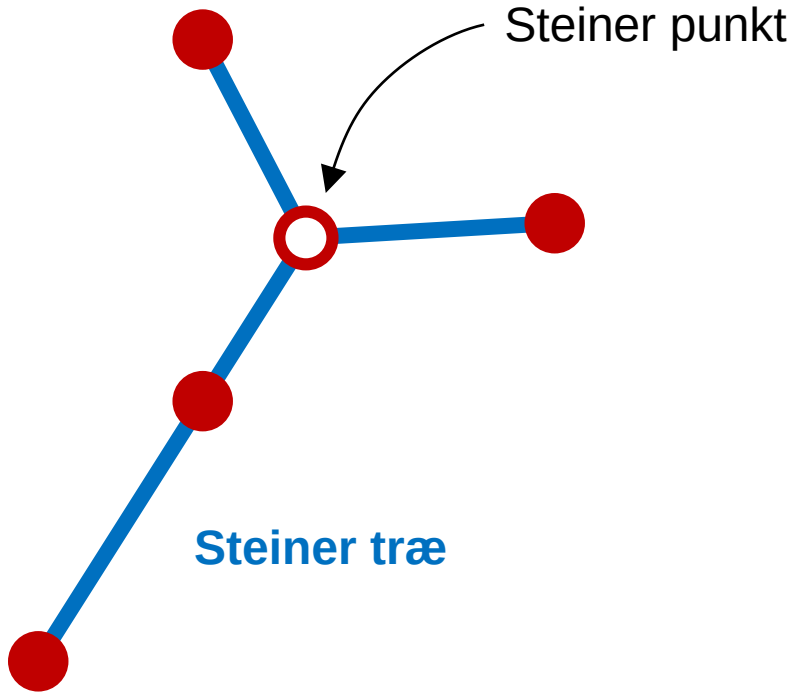
IKKE PENSUM

Jakob Steiner [1796-1863]

(Euklidiske) Steiner Træer

Givet en mængde punkter, find et træ med minimum vægt som forbinder alle knuder, muligvis ved at tilføje **nye punkter**





Conjecture (Steiner Ratio): $|\text{MST}| \leq \frac{2}{\sqrt{3}} \cdot |\text{Steiner Tree}|$

NP hårdt; polynomial-time approximation scheme (PTAS)