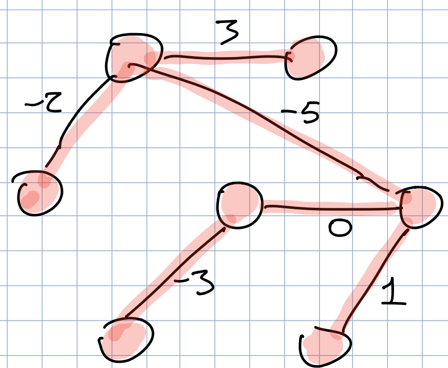
Midterm 2 — next Mon

Today's lecture won't be on the exam.

Minimum Spanning Trees



Assumption:

$w(e) \neq w(e')$ iff $e \neq e'$

$\Rightarrow$ MST is unique

Strategy:

Forest $F \leftarrow (V, \emptyset)$

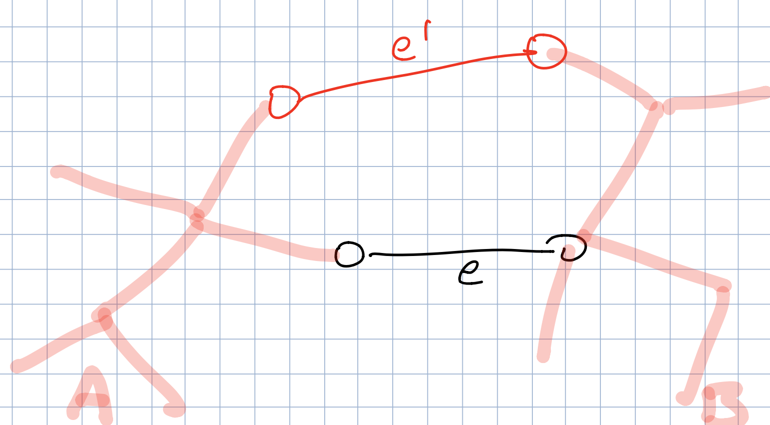
while F is disconnected

either
  add a safe edge to F
or
  delete a useless edge from G

safe = min wt edge with one end in some component of F.

useless = both ends in same component of F

Lemma: Every safe edge is in the MST



e is lightest edge leaving A

Spose $e \notin$ MST but e' is.

e is safe for A

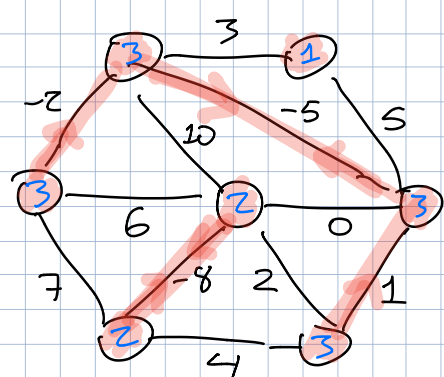
$w(e) < w(e')$

T = actual MST $\qquad T' = T - e' + e$

another sp. tree. smaller wt ⨳

Borůvka 1926

1. Add all safe edges to F $\Leftarrow$ $O(V+E)$ time
2. Recurse.



Label components of F in $O(V)$ time by WFS

For each edge uv $\quad + O(E)$

if $label(u) \neq label(v)$

if $w(uv) \leq w(Best[label(u)])$

$Best[label(u)] \leftarrow uv$

sim for v

for $i \leftarrow 1$ to #comps of F

add $Best[i]$ to F

How many rounds?

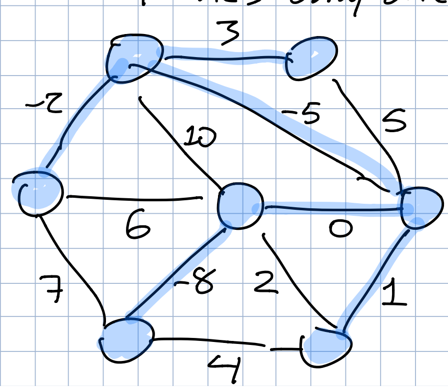
$O(\log V)$!

Every round divides #comps by 2! Or better!!

$\Rightarrow O((V+E)\log V) = O(E \log V)$

Jarník (1936) $\longrightarrow$ Prim (1956)

F has only one interesting component.



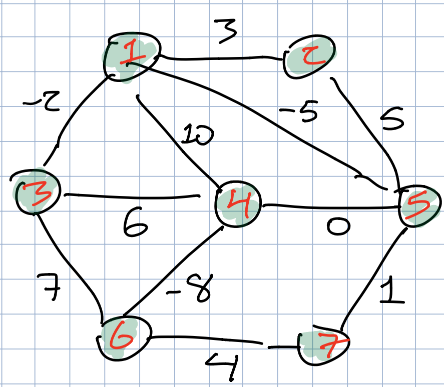
Best-First search

WFS with a priority queue where $priority(uv) = w(uv)$

$\boxed{O(E \log V)}$

Fibonacci heap: $O(E + V \log V)$

Kruskal (1956)



$O(E \log E) = O(E \log V)$

Sort E by increasing wt

For all edges $e \in E$

if e is not useless

add e to F

$\to O(E + V \log V)$

Maintain disjointset DS.

When add uv to F

→ relabel smaller of two components.

Every node is relabeled $O(\log V)$ times

Work harder $V \log V \longrightarrow \boxed{V\alpha(V)}$

$\log n$ = # reps of $x/2$ to get from n to 1

$\log^* n$ = ——— log ———

$\log^{**} n$ $\quad$ $\log^*$

$\log^{***} n$ $\quad$ $\log^{**}$

$\alpha(n)$ # stars s.t. $\log^{**\cdots*} n \le 2$