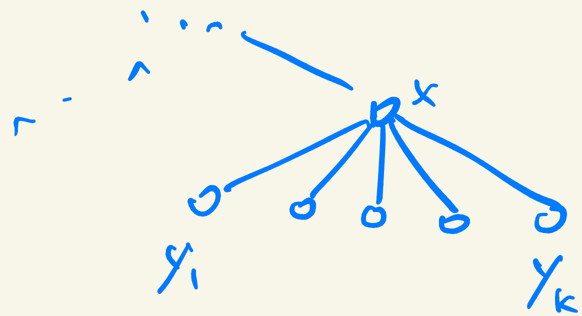
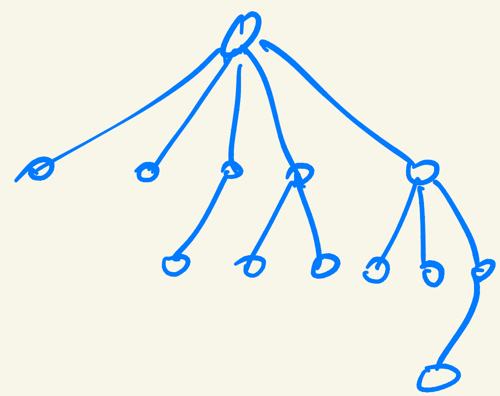
FREDMAN-TARJAN : Fibonacci heap

FT tree



rule :

\# children of child y_i is $\geq i-2$

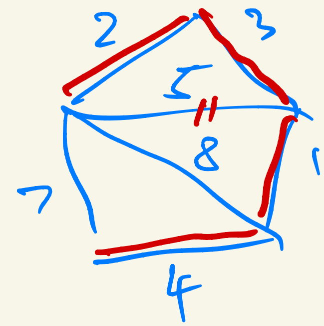


Ex. $ft(k)$: min # nodes of FT tree with root having k children

$ft(k) = ?$

$\therefore$ depth $= O(\log n)$

min-weight spanning tree

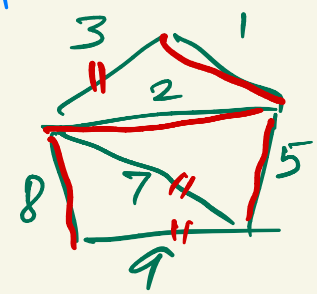


INPUT

G graph (undirected)
connected

$w: E \to \mathbb{R}$

GREEDY ALGORITHM



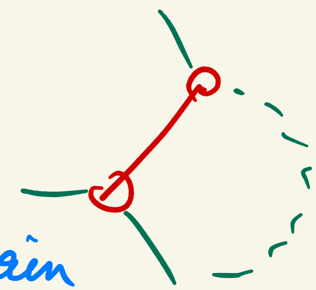
$O(m \log n)$ Sort edges by weight $e_1, \dots, e_m$

$F := \emptyset$ (set of edges of sp. tree)

for $i=1$ **to** m

  if $F \cup \{e_i\}$ does not have a cycle

  then $F \leftarrow F \cup \{e_i\}$



need to maintain connected components

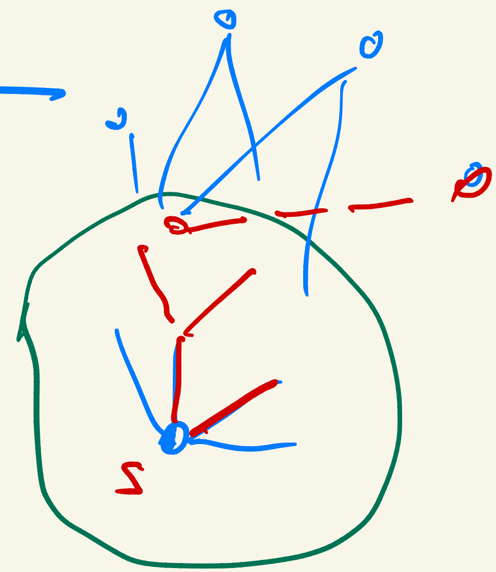
KRUSKAL'S ALGORITHM

← KRUSKAL'S ALGORITHM

"pure greedy"

PRIM's alg (1957)

JARNÍK (1930)



growing tree T from root s

keep adding vertices

current set of vertices B

add lightest edge from B to V \ B

pseudocode identical with Dijkstra's 1956/59

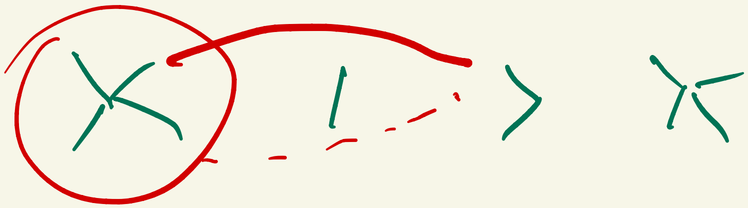
EX. except for RELAX

EDSGER DIJKSTRA

Dutch

BORŮVKA's algorithm 1926

Otakar Borůvka
Vojtěch Jarník } Czech



simultaneously for all components
select lightest edge
<u>leaving</u> that component

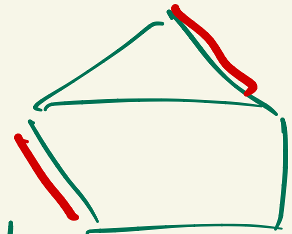
GREEDY: "optimist's algorithm"

REVERSE GREEDY: "pessimist's algorithm"
keep deleting heaviest edge
as long as deletion does not <u>disconnect</u> graph

<u>THEOREM</u> All these methods produce
min-weight spanning tree

G (undirected) graph

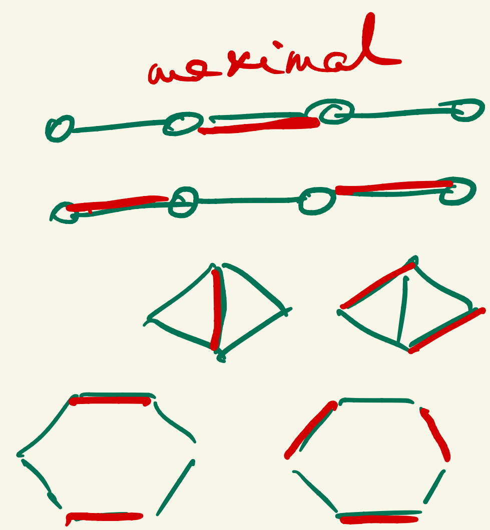
DEF MATCHING : set of disjoint edges



MAXIMUM matching : max # disjoint edges

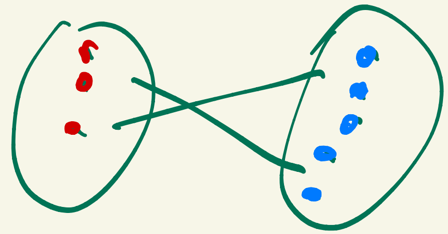
MAXIMAL " : no edge can be added

find G and maximal matching that is not maximum

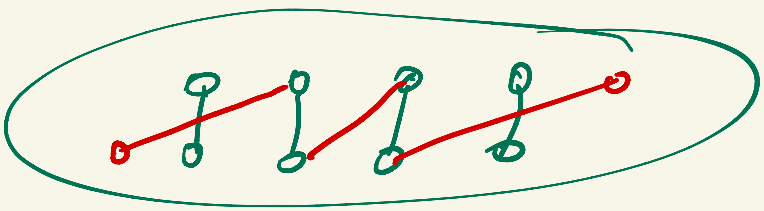


GREEDY produces maximal

EX maximal $\geq \frac{1}{2}$ maximum for bipartite graphs — marriage problem



Solved by DÉNES KŐNIG ~ 1930 (Hungary)



FIND **AUGMENTING PATH**

~BFS

Theorem If $\nexists$ augmenting path
then current matching maximum

---

for general graphs solved by JACK EDMONDS (Canada)

~1970 invented "polynomial time"

this algorithm: illustrates the power of — " — concept

first major success of asymptotic analysis of algorithms