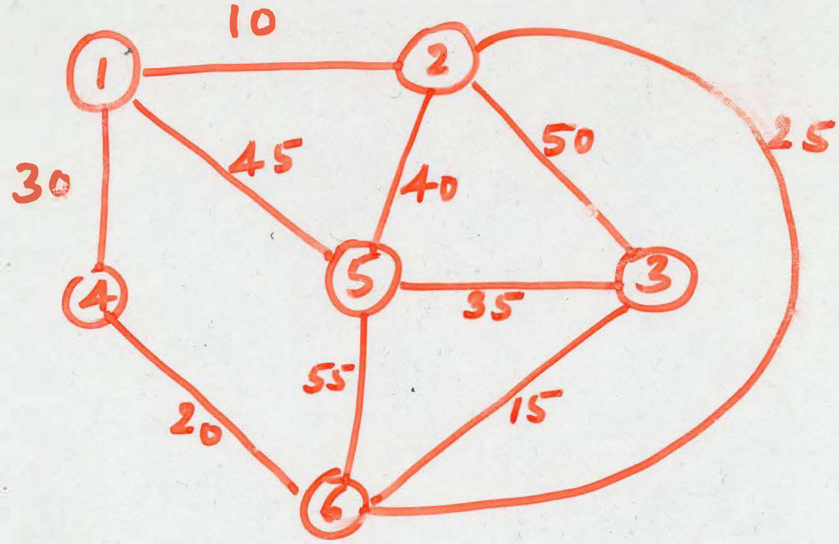
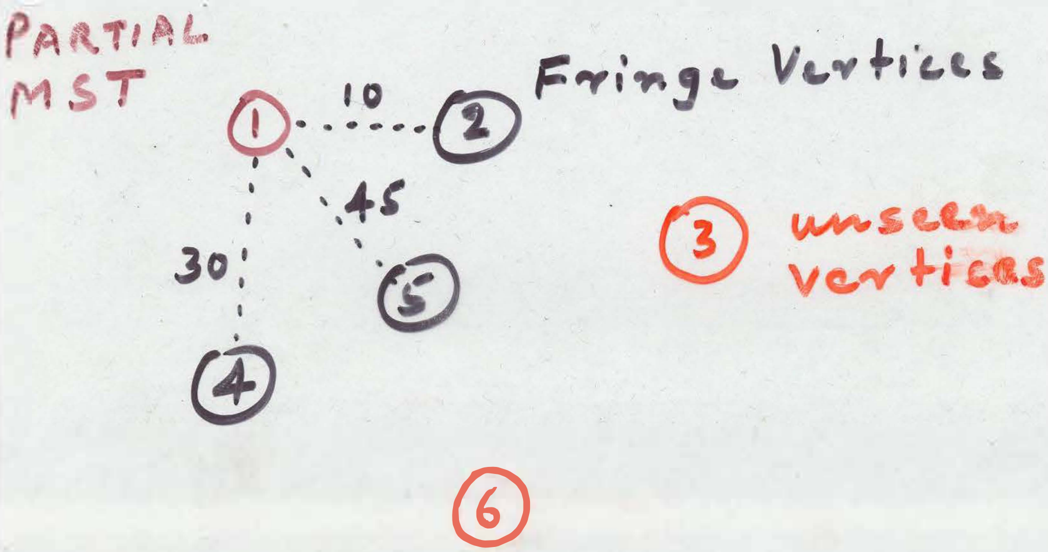
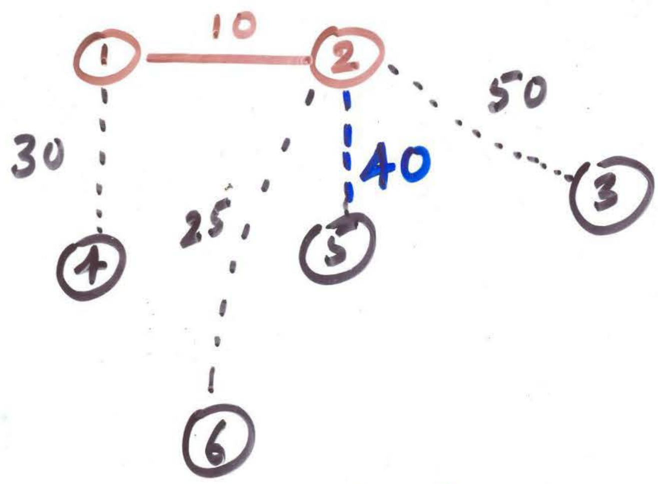
PRIM-DIJKSTRA'S MST ALGORITHM

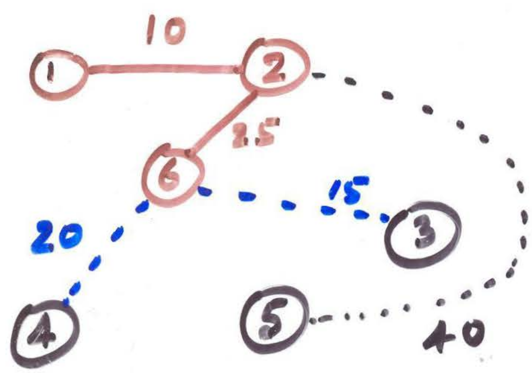




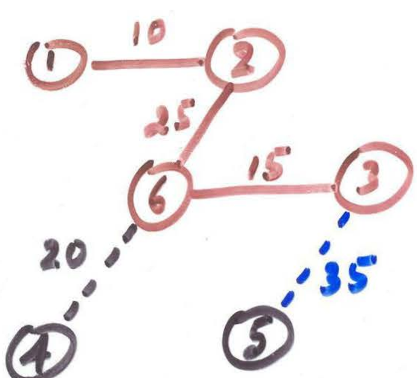
Choose node 2



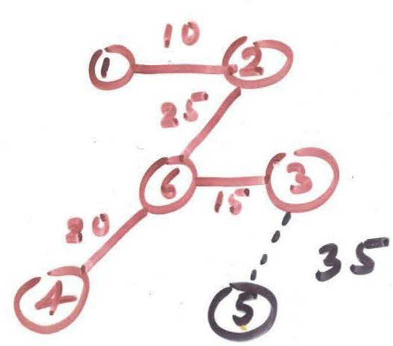
choose node 6



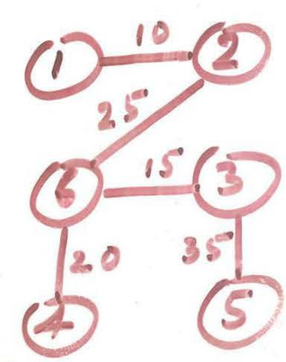
choose node 3



choose 4



choose 5



MST

w = 105

PRIM-DIJKSTRA MST ALGORITHM

Input: $G = (V, E, w)$

Output: $T = (V_T, E_T)$

$E_T = \emptyset$

Select a vertex $v \in V$ and move v to V_T ; $V_T = \{v\}$, $V = V - \{v\}$

For $i = 1$ to $n-1$ do

$)(n^2)$ Let $\{v, w\}$ be an edge such that $v \in V_T$, $w \in V$, and for all such edges, $\{v, w\}$ has the minimum weight.

$V_T = V_T \cup \{w\}$

$E_T = E_T \cup \{\{v, w\}\}$

$V = V - \{w\}$

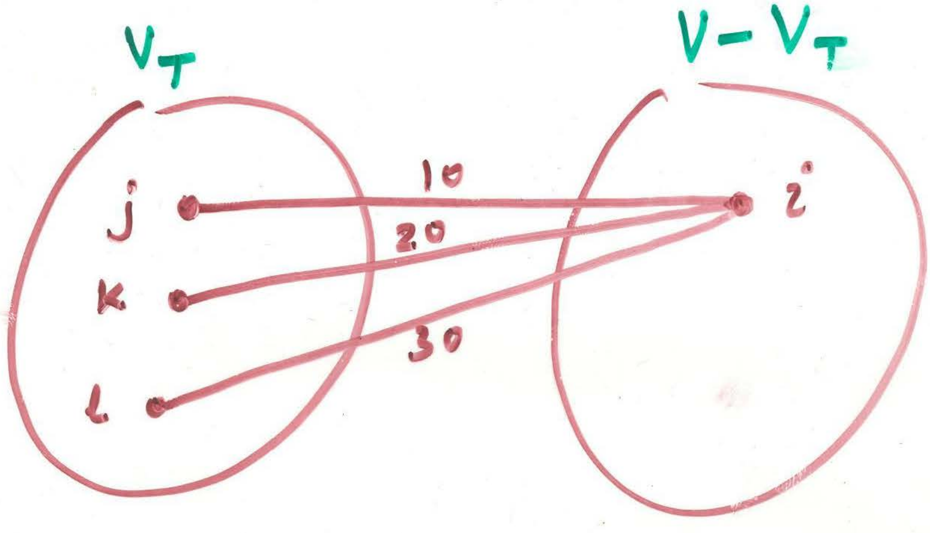
endfor

$W(n) = O(n^3)$

DATA STRUCTURE FOR PRIM'S ALGORITHM

NEAR[1 . . n]

$$NEAR[i] = \begin{cases} 0 & \text{if } i \in V_T \\ j & \text{if } w(i,j) \text{ is minimum among all } j \in V_T \end{cases}$$



NEAR[i] = j

NEAR[j] = 0

NEAR[l] = 0

MODIFIED PRIM'S ALGORITHM

Input $G = (V, E, w)$, a connected weighted graph

Output $T = (V_T, E_T)$, a MST.

1. $E_T = \phi$
2. NEAR [1] = 0 /* $V_T = \{1\}$ */
   NEAR [2..n] = 1 /* $V = V - \{1\}$ */

3. For $i = 1$ to $n-1$ do

O(n)
O(log n)

FIND j such that NEAR[j] $\neq 0$
AND $w(j, \text{NEAR}[j])$ is min.

NEAR[j] = 0 /* $V_T = V_T \cup \{j\}$ */

$E_T = E_T \cup \{\{j, \text{NEAR}[j]\}\}$.

/* update NEAR[1..n] */

for k = 1 to n do

O(n log n)

if NEAR[k] $\neq 0$ AND
$w(k, \text{NEAR}(k)) > w(k, j)$

DECREASE-KEY

then NEAR[k] = j

endfor

endfor.

$W(n) = O(n^2)$

Thm Let $G = (V, E, w)$ be a weighted connected graph and $T = (V, E_T)$ be a MST of G. Let $T' = (V', E')$ is a subtree of T. If $\{x, y\}$ is the minimum weight edge such that

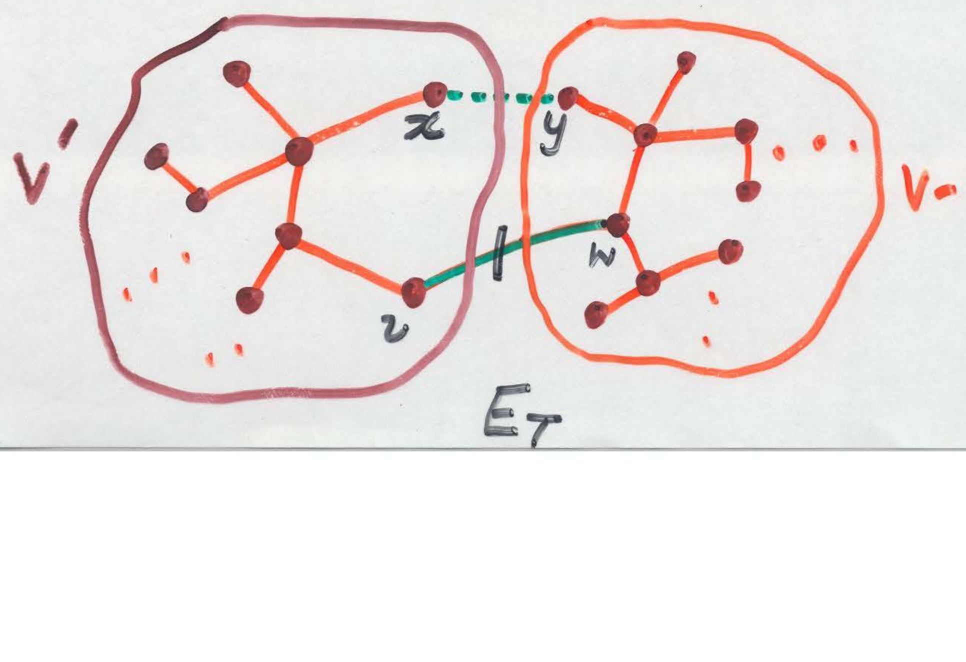
$$x \in V' \text{ and } y \in V - V'$$

then $T'' = (V' \cup \{y\}, E' \cup \{\{x, y\}\})$ is a subtree of a MST of G.

Proof

1. If $\{x, y\} \in E_T$, done.
2. Let $\{x, y\} \notin E_T$.

   Then $E_T \cup \{\{x, y\}\}$ has a cycle:



E_T

By the choice of $\{x, y\}$

$$w(\{x, y\}) \leq w(\{v, w\})$$

Consider $E_T \cup \{\{x, y\}\} - \{\{v, w\}\}$

Its weight is no more than the weight of T and it is a spanning tree.

$\Rightarrow E' \cup \{\{x, y\}\}$ is a subtree of a MST of G.

$\square$

Kruskal's Algorithm

input: $G = (V, E, W)$, a connected graph

output: $T = (V, E_T)$, a MST of G.

$T \leftarrow \emptyset$

while $|T| < n-1$ do

Let $\{v, w\}$ be the least cost edge in E.

$E \leftarrow E - \{\{v, w\}\}$

if $\{v, w\}$ does not create a cycle in T

then add $\{v, w\}$ to T

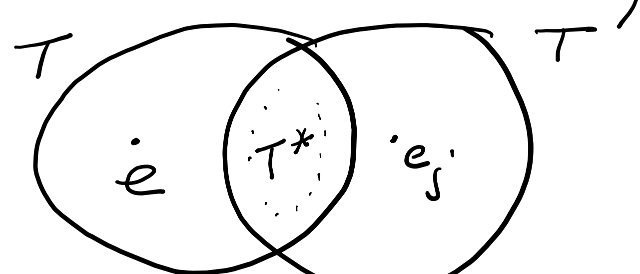
end while

Kruskal's MST Algorithm
Proof of Correctness

Th.: Let T be the spanning tree for G generated by kruskal's algorithm. Let T' be a minimum cost spanning tree for G. Show that

$w(T) \leq w(T')$, and T is a MST.

Proof:



Let $e \in T - T'$ of min Weight.

$\{e\} \cup T'$ has a cycle.

Let $e_j \in T' - T$ in this cycle.

$w(e_j) \geq w(e)$ else $e_j \in T$ by kruskal's

{ Why? e_j would have been considered before e for inclusion into a subset $T^* \subseteq T \cap T'$.

$\Rightarrow \{e_j\} \cup T^*$ would not form cycle because $T^* \subset T'$.

$$\Rightarrow w(T') \geq w(\underbrace{T' - \{e_j\} \cup \{e\}}_{T''}) \geq w(T)$$

Repeat for each $e \in T - T'$ $\Rightarrow$ T' becomes T

$\Rightarrow$ T is a MST. $\square$

Kruskal's Algorithm (Refined)

OPERATIONS

1. UNION(i, j), of sets i & j, contains elements of sets i and j.

2. FIND$(v) = i$ iff $v \in$ set i.

a) Construct a min-heap E of edges in E.

b) Each $v \in V$ forms singleton set by itself, such that FIND$(v) = v$.

c) $E_T = \phi$ {Tree is empty}

d) while $|E_T| < n-1$ do

log m Delete the root edge $\{v, w\}$ from min-heap E and restore heap E.

$\boxed{O(1)}$ O(log n) if FIND$(v) \neq$ FIND(w)

{ $E_T \cup \{\{v, w\}\}$ has no cycle }

then

O(1) $E_T = E_T \cup \{\{v, w\}\}$

/* now, combine the components of E_T joined by $\{v, w\}$ into one */

O(1) UNION (FIND(v), FIND(w))

endif

endwhile O(m log m)

1. UNION (1, 2)

2. UNION (2, 3)

⋮

$n/2$. UNION ($n/2$, $n/2+1$)

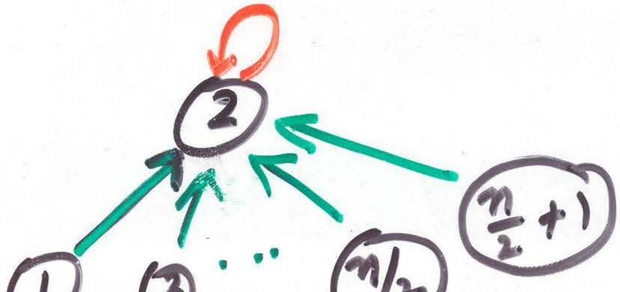
$\frac{n}{2}+1$ FIND (1)

FIND (1)

⋮

n FIND (1)

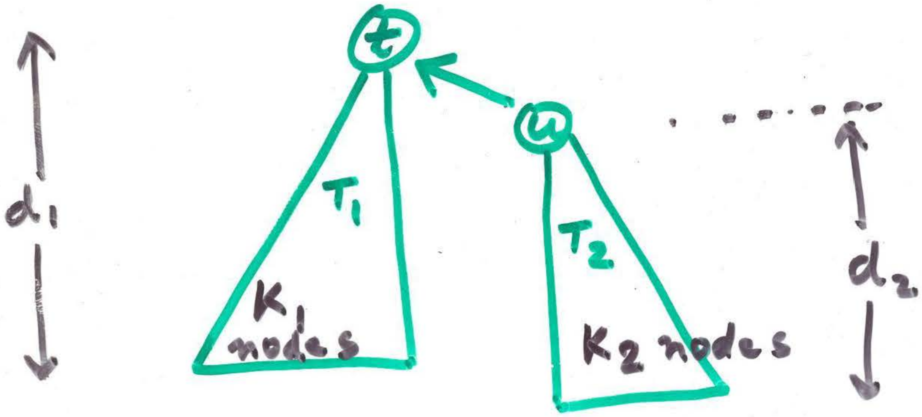
TREE WITH UNWEIGTED UNION



TREE WITH WEIGTED UNION

Lemma With W-UNION, ANY TREE THAT HAS k NODES HAS DEPTH AT MOST $\lfloor \lg k \rfloor$.

W-UNION(t, u)



Let $k_1 \geq k_2$, $k = k_1 + k_2$

$$\begin{aligned} d &= d_1 \quad \text{if} \quad d_1 > d_2 \\ &= d_2 + 1 \quad \text{if} \quad d_1 \leq d_2 \end{aligned}$$

BASIS: $k=1$ $\quad \lfloor \lg 1 \rfloor = 0$

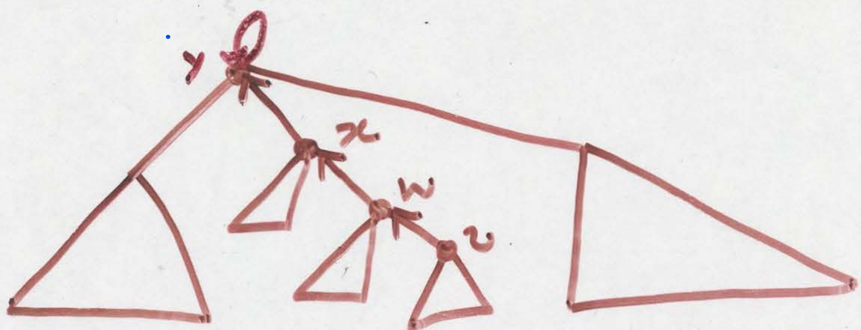
HYPOTHESIS: for $k' < k$, depth $\leq \lfloor \lg k' \rfloor$

INDUCTION

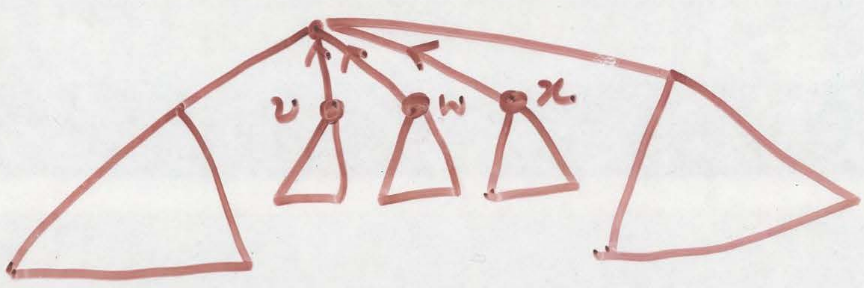
I. $d = d_1 \leq \lfloor \lg k_1 \rfloor \leq \lfloor \lg (k_1 + k_2) \rfloor = \lfloor \lg k \rfloor$

II. $d = d_2 + 1 \leq \lfloor \lg k_2 \rfloor + 1 = \lfloor \lg 2k_2 \rfloor \leq \lfloor \lg (k_1 + k_2) \rfloor$

Lemma A UNION-FIND PROGRAM OF SIZE n DOES $\Theta(n \lg n)$ link OPERATIONS IN THE WORST CASE IF THE WEIGTED UNION IS USED.



BEFORE C-FIND(v)
COMPRESSING-FIND(v)



AFTER C-FIND(v)

Lemma THE NUMBER OF LINK OPERATIONS DONE BY A UNION-FIND PROGRAM OF LENGTH n IMPLEMENTED WITH W-UNION AND C-FIND IS $\underline{O(nG(n))}$.

Tarjan & Hopcroft

$$G(n) = \log^* n$$

$=$ SMALLEST i SUCH THAT

$$\log^{(i)} n \le 1$$

where $\log^{(i)} n = \log(\log^{(i-1)} n)$

and $\log^{(0)} n = n$

$$G(65536) = \log^*(2^{16}) = 4$$

$\log 2^{16} = 16$, $\log 16 = 4$, $\log 4 = 2$, $\log 2 = 1$

$G(2^{65536}) = 5. \Rightarrow \underline{G(n) \le 5}$ for all reasonable n.

[pp. 498-508 chap 22]

COMPARISON

m	$O(n)$	$O(\frac{n^2}{\log n})$	$O(n^2)$
Kruskal's	$O(n\log n)$	$O(n^2)$	$O(n^2 \lg n)$
Prim's	$O(n^2)$	$O(n^2)$	$O(n^2)$

Kruskal's
$O(m \log m)$
$= O(m \log n)$

PRIM'S
$O(n^2)$

Time Complexity of Prim's

- with heap $O(n\log n + m \log n)$
- with Fibonacci Heap
  $O(n \log n + m)$