PRIM-Dijkstra's MST Algorithm

Partial MST

Fringe Vertices

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

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.

\[ V_T = V_T \cup \{w\} \]
\[ E_T = E_T \cup \{\{v, w\}\} \]
\[ V = V - \{w\} \]

end for

\( w(n) = O(n^3) \)
DATA STRUCTURE FOR PRIM'S ALGORITHM

\[ \text{NEAR}[1 \ldots n] \]
\[ \text{NEAR}[i] = \begin{cases} 
0 & \text{if } i \in V_T \\
\text{min} & \text{if } w(i,j) \text{ is minimum among all } j \in V_T
\end{cases} \]

\[ \text{NEAR}[i] = j \]
\[ \text{NEAR}[j] = 0 \]
\[ \text{NEAR}[l] = 0 \]
Modified Prim's Algorithm

Input $G = (V, E, w)$, a connected weighted graph
Output $T = (V_T, E_T)$, a MST.

1. $E_T = \emptyset$
2. $\text{NEAR}[1] = 0$ /* $V_T = \{1\}$ */
   $\text{NEAR}[2..n] = 1$ /* $V = V - \{1\}$ */
3. For $i = 1$ to $n-1$ do
   Find $j$ such that $\text{NEAR}[j] \neq 0$
   and $w(j, \text{NEAR}[j])$ is min.
   $\text{NEAR}[j'] = 0$ /* $V_T = V_T \cup \{j\}$ */
   $E_T = E_T \cup \{(j, \text{NEAR}[j])\}.$
   /* Update $\text{NEAR}[1..n]$ */
   For $k = 1$ to $n$ do
   if $\text{NEAR}[k] \neq 0$ and $w(k, \text{NEAR}(k)) > w(k, j)$
   then $\text{NEAR}[k] = j$

$T(n) = O(n \log n)$

$w(n) = O(n^2)$
Theorem: 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 $x \in V'$ and $y \in V - V'$, then $T'' = (V' \cup \{y\}, E' \cup \{\{x, y\}\})$ is a subtree of the 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.
By the choice of \( \{x,y\} \)
\[ w(\{x,y\}) \leq w(\{y,w\}) \]

Consider \( E' \cup \{\{x,y\}\} - \{\{y,w\}\} \)

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

\[ E' \cup \{\{x,y\}\} \text{ is a subtree of a MST of } G. \]
Kruskal's Algorithm

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

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

$T \leftarrow \emptyset$

while $|T| < n-1$ do

let $\{u, w\}$ be the least cost edge in $E$.

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

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

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

end while

1. Characteristic Vector

2. Linked List
**Kruskal's Algorithm (Refined)**

**OPERATIONS**

1. **\textsc{union}(i,j)**, of sets \(i \& j\), contains elements of sets \(i\) and \(j\).
2. **\textsc{find}(v) = i** iff \(v \in \text{set } i\).

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

b) Each \(v \in V\) forms singleton set by itself, such that **\textsc{find}(v) = v**.

c) \(E_T = \emptyset\) [Tree is empty]

d) \textbf{while} \(|E_T| < n-1\) \textbf{do}

\[\log m\] Delete the root edge \([v,w]\) from min-heap \(E\) and restore heap \(E\).

\[O(\log n)\] \textbf{if} \(\textsc{find}(v) \neq \textsc{find}(w)\) \[\{E_T \cup \{v,w\}\} \text{has no cycle}\)

\[O(1)\] \textbf{then}

\[E_T = E_T \cup \{v,w\}\]

\[O(1)\] \textbf{end if}

\[O(1)\] \textbf{end while}

\[O(m \log m)\]
1. UNION (1, 2)
2. UNION (2, 3)
   .
  \[ \frac{n}{2} \]
  UNION (\[ \frac{n}{2}, \frac{n}{2} + 1 \])
\[ \frac{n}{2} + 1 \]
FIND (1)
FIND (1)
. . .
FIND (1)

TREE WITH UNWEIGHTED UNION

TREE WITH WEIGHTED UNION
Lemma with \textit{W-union}, any tree that has $k$ nodes has depth at most $\lceil \log k \rceil$.

![Diagram of two rooted trees connected by W-union](image)

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

$$d = d_1 \text{ if } d_1 > d_2$$
$$d = d_2 + 1 \text{ if } d_1 \leq d_2$$

**Basis:** $k = 1$, $\lceil \log 1 \rceil = 0$

**Hypothesis:** for $k' < k$, depth $\leq \lceil \log k' \rceil$

**Induction**

I. $d = d_1 \leq \lceil \log k_1 \rceil \leq \lceil \log (k_1 + k_2) \rceil = \lceil \log k \rceil$

II. $d = d_2 + 1 \leq \lceil \log k_2 \rceil + 1 = \lceil \log 2k_2 \rceil \leq \lceil \log (k_1 + k_2) \rceil$
Lemma: A union-find program of size $n$ does $\Theta(n \log n)$ link operations in the worst case if the weighted union is used.

Before $C$-FIND$(u)$

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 \( \Theta(n \log(n)) \).

Tanjun & Hopcroft

\[ G(n) = \log^* n \]

= Smallest \( i \) such that \( \log^{(i)} n \leq 1 \)

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

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

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

\[ \log 2^{16} = 16, \quad \log 16 = 4, \quad \log 4 = 2, \quad \log 2 = 1 \]

\[ G(2^{65536}) = 5. \quad \Rightarrow \quad G(n) \leq 5 \text{ for all reasonable } n. \]
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>( O(n) )</th>
<th>( O(\frac{n^2}{\log n}) )</th>
<th>( O(n^2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>Kruskal's</td>
<td>( O(n \log n) )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td></td>
<td>Prim's</td>
<td>( O(n^2) )</td>
<td>( O(n^2) )</td>
</tr>
</tbody>
</table>

Kruskal's:

\[
O(\text{m log m}) = O(n \log n)
\]

Prim's:

\[
O(n^2)
\]

Time Complexity of Prim's:

- With heap: \( O(n \log n + m \log n) \)
- With Fibonacci Heap: \( O(n \log n + m) \)