Minimum Cost Spanning Tree

$G = (V, E)$

Tree: $T = (V', E')$; $V' = V$; $|E'| = |V| - 1$

$\downarrow$

induced by the graph G

T has no cycles

$T - e + e'$ is also a Tree

$\underset{\in E'}{e}$, $\underset{\in E}{e'}$

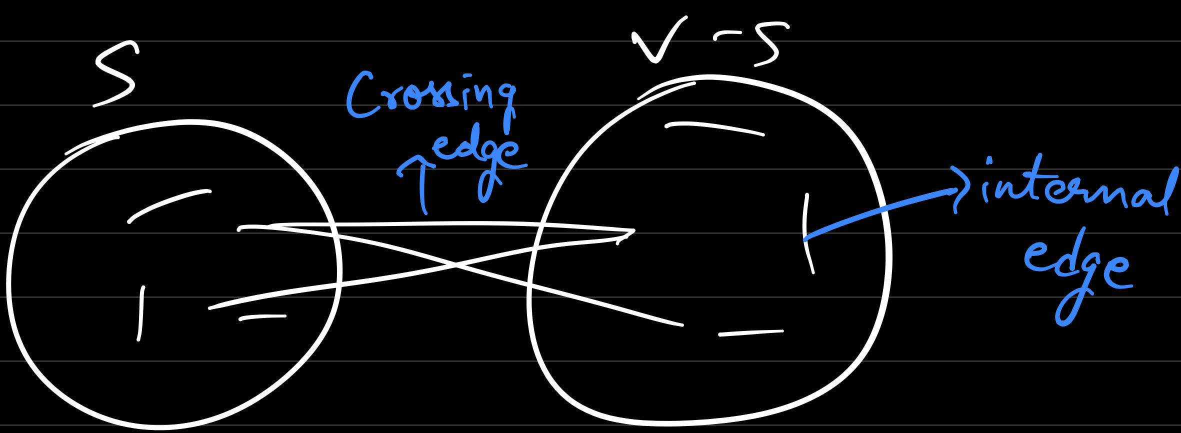
Vertex Set partition in a Graph:

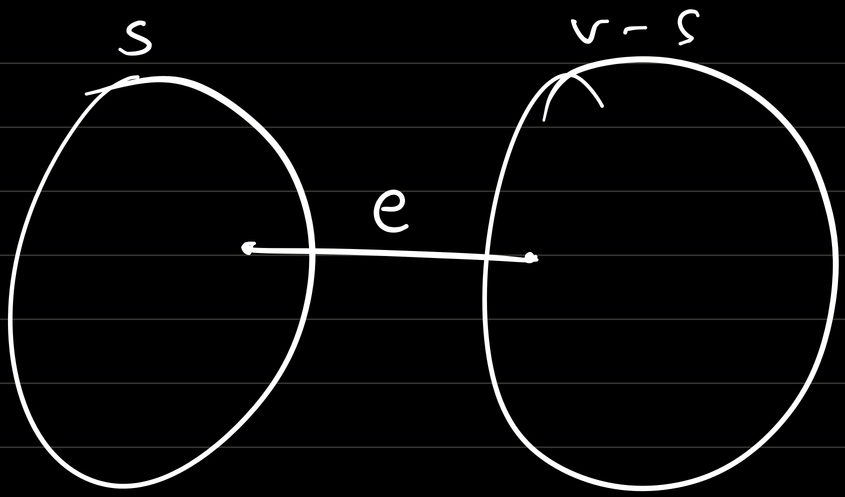
$(S, V-S)$ is a partition and

Internal edge $\Rightarrow$ (u, v) and u and v, both belong to exactly one of the partitions

Crossing edge $\Rightarrow$ (u, v) and $u \in S$ and $v \in V-S$ (or) $u \in V-S$ and $v \in S$

Let $A \subseteq E$. We say a cut $(S, V-S)$ is respecting A iff all the edges in A are internal w.r.t the cut, $(S, V-S)$.





Let e be a crossing edge and e be of minimum weight among the crossing edges existing w.r.t cut $(S, V-S)$.

$\Rightarrow$ $A \cup \{e\} \subseteq$ Some Spanning tree

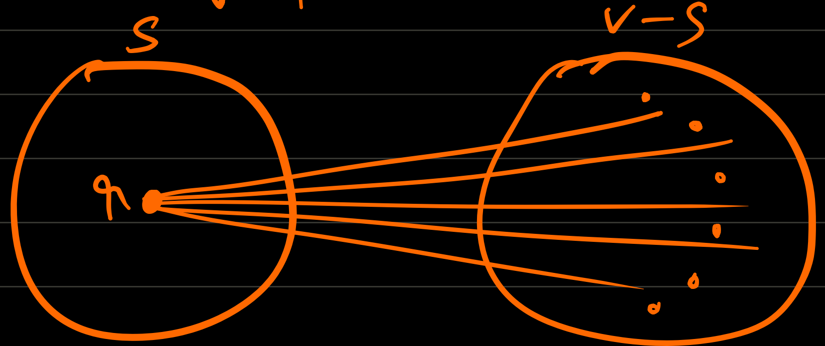
A $\subseteq$ Some Spanning Tree

e is a min. weight crossing edge of a cut respecting A.

Then, $A \cup \{e\} \subseteq$ Some Spanning Tree

$A = \phi$ (initialization)

So, initially, the cut can have any vertex in the graph in the vertex set S.



$S = \{r\}$ $\qquad$ $V - S = V - \{r\}$

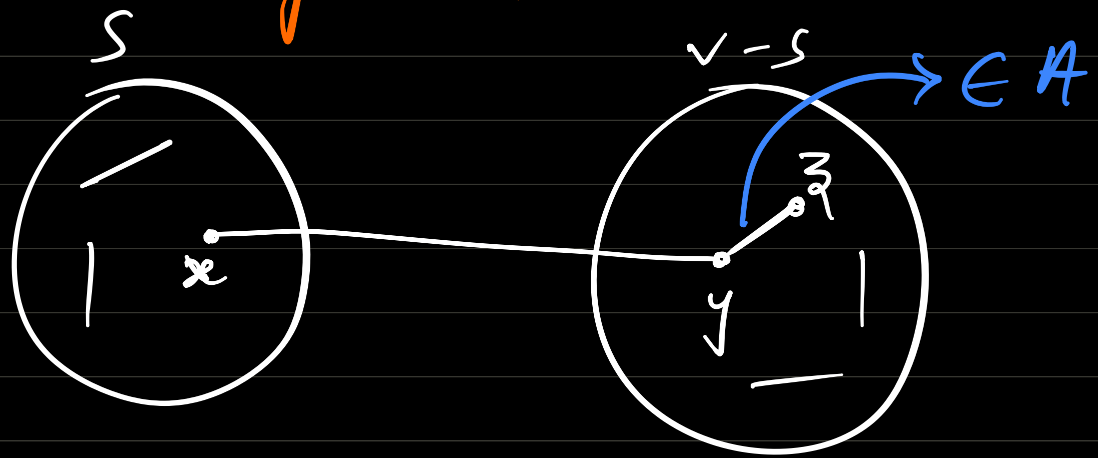
Look at the adjList(r) and get the minimum edge weight

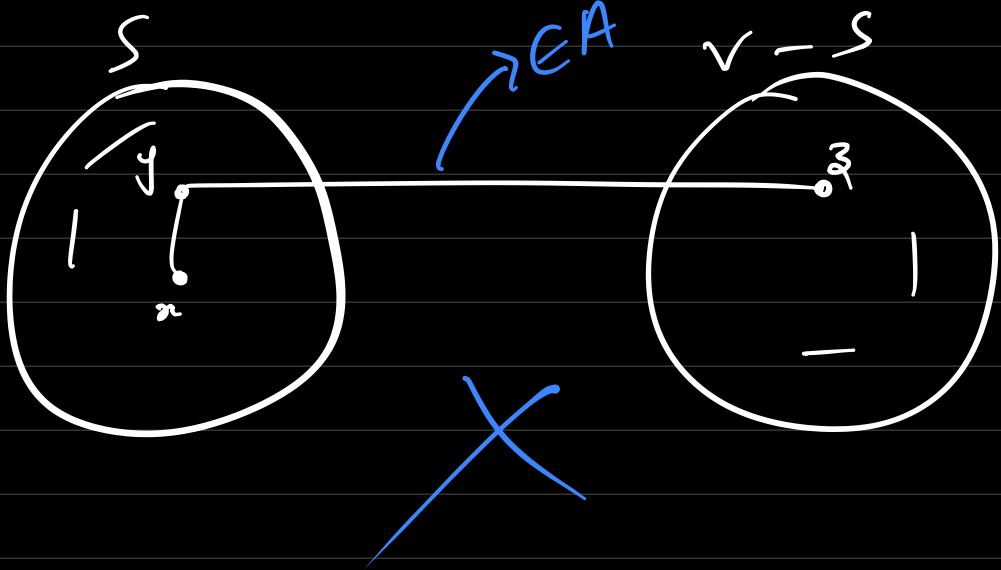
min. Cost Crossing edge is an edge (u, v) Such that
$w(u, v) \leq w(u, x) \quad \forall x \in adjList(u)$

Update : $A \leftarrow A \cup \{(u,v)\} = \phi \cup \{(u,v)\}$
$= \{(u,v)\}$

move : $S \leftarrow S \cup \{v\}$
$V-S \leftarrow V-S$

✗ an internal edge will become a crossing edge when we move v to S.





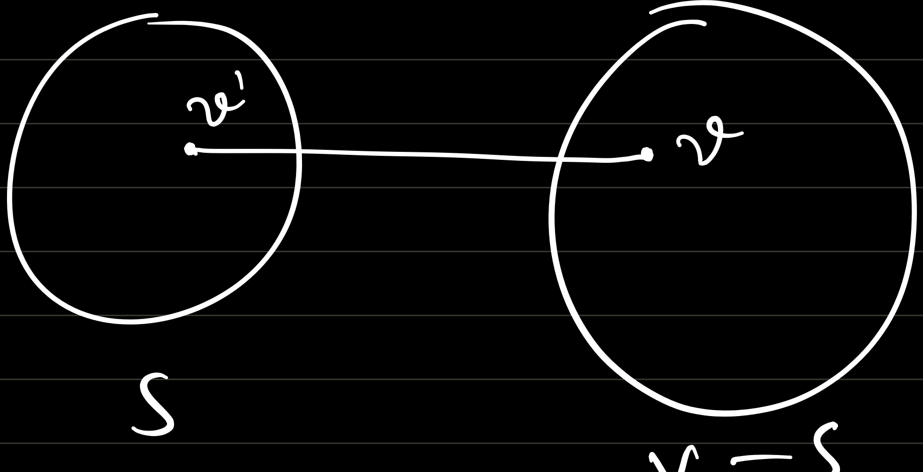
Idea: make Sure that no edge in A is in V-S.
i.e, we maintain all edges of A as internal to S.

Development of Prims Algorithm:

$$\frac{S}{\{r\}} \;\Big|\; \frac{V-S}{V-\{r\}}$$

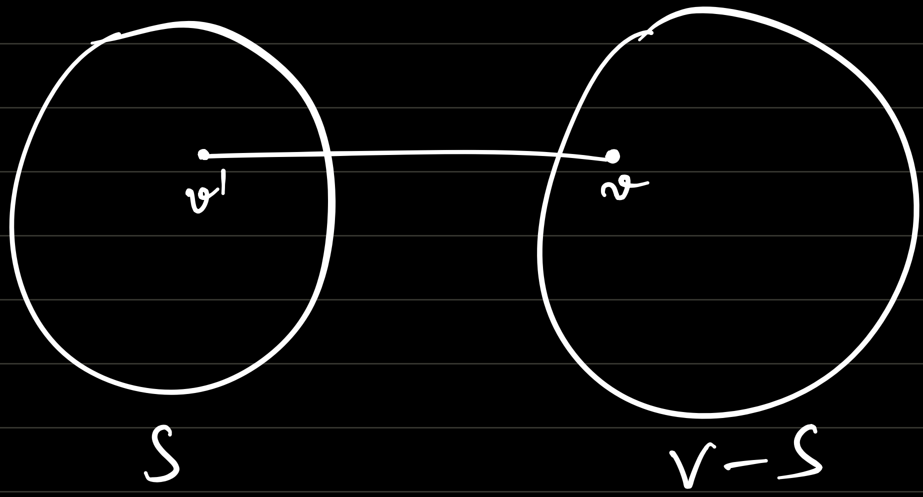
* for every $v \in V-S$, we maintain a minimum weight crossing edge (v, v') such that $v' \in S$.

  Then, the overall min. weight crossing edge can be identified by only looking at the type of edges (v, v') $\forall$ $v \in V-S$.



$\Rightarrow$ By examining only $|V-S|$ edges, we identify the minimum crossing edge

$\because$ $|V-S| \leq n-1$, this is a significant improvement.



(v, v') : min. weight crossing edge incident on v.

$\Rightarrow \quad w(v, v') \leq w(v, u) \quad \forall u \in S$

i.e, $\forall$ Crossing edges (v, u)

$v' = p(v)$ (partner of v)

$c(v) = w(v, p(v))$

$\therefore$ for each $v \in V - S$, we maintain $p(v)$ and $c(v)$ arrays.

Let $v \in V - S$ such that

$c(v) \leq c(u) \quad \forall u \in V - S$

Then, clearly $(v, p(v))$ is the min. wt. crossing edge.

Hence add $(v, p(v))$ to A, move v to S and $p(u)$, $c(u)$ values must be updated $\forall u \in V - S$

$p(u)$ is updated only if $\exists\, (v, u) \in E$ and $w(v, u) < w(u, p(u))$

$p(u) \leftarrow v.$

High Level pseudocode

$A = \phi$

$V - S = V - \{r\}$

For all $v \in V$,
   $p(v) = \text{NULL}$,
   $c(v) = \infty$

For each $u \in V - S$
   if $(r, u) \in E$
      $p(u) = r;$
      $c(u) = w(r, u);$

while $(V - S \neq \phi)$
{
   Find a vertex $v \in V - S$
      with minimum $c(v)$
   move v to S and add $(v, p(v))$ to A

   Update $c(u)$ and $p(u)$ $\forall$ $u \in V - S$
}

Detailed Pseudocode for Prim's MST Algorithm

primMST((V, E))
{
$S = \phi$
$p(v) = NULL \quad \forall v \in V$
$c(v) = \infty \quad \forall v \in V$

$S = \{r\}$

For all $u \in V - S$
{
if $(u, v) \in E$
$p(u) = r$
$c(u) = w(u, r)$
}

while $(V - S \neq \phi)$
{
Find $v \in V - S$ such that
$c(v) \leq c(u) \quad \forall u \in V - S$

move v to S

For each $u \in V - S$
if $(u, v) \in E$ && $w(u, v) < c(u)$
$p(u) = v$
$c(u) = w(u, v)$
}

Return $p(v)$
}

// $T = (V, A)$ is an MST
// $A = \{ (v, p(v)), \ v \in V - \{r\} \}$