30.09.2024

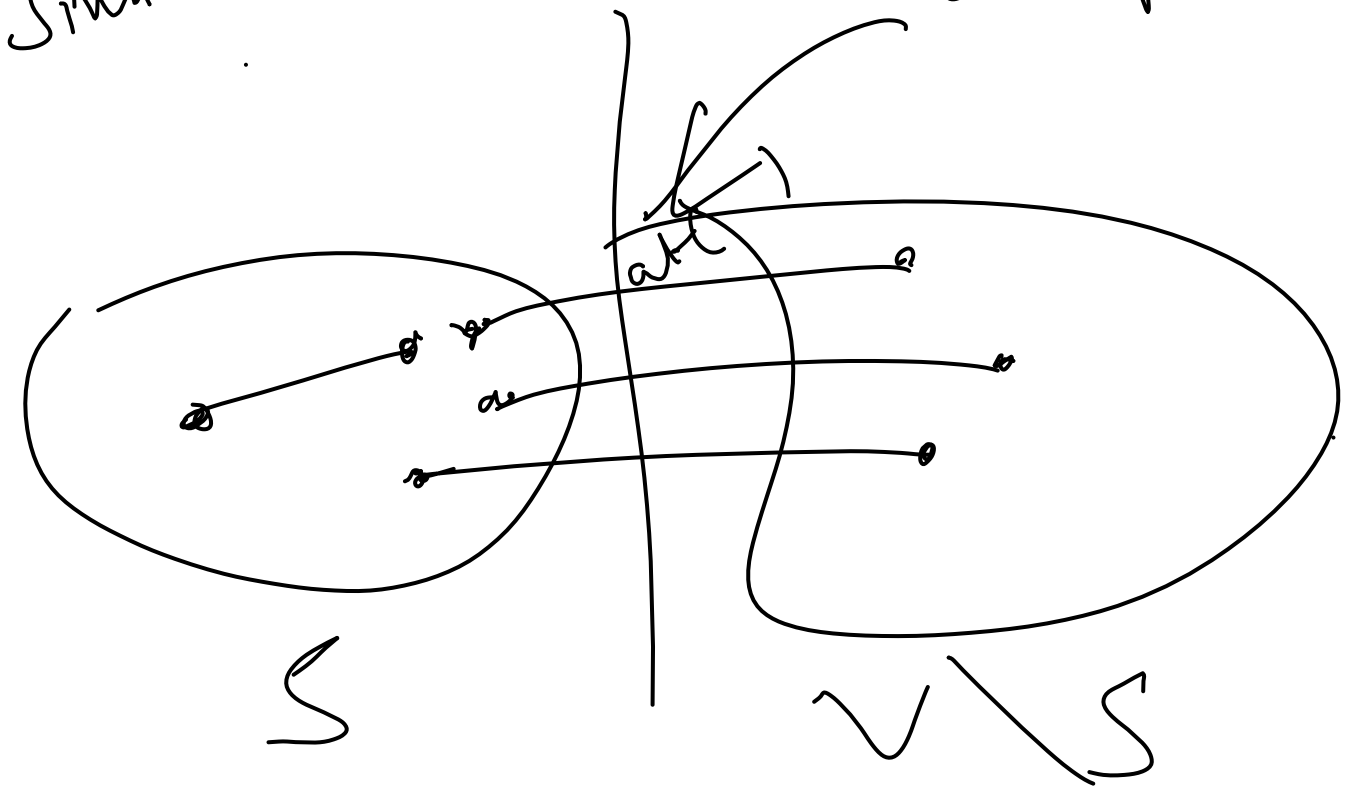
MST:

- Kruskal's algo: picks next "available" edge (in sorted order) + doesn't create cycle

- Prim's algo: picks next "available" neighbor. (in sorted order).

Similar. dijkstra's algorithm

cheapest edge.

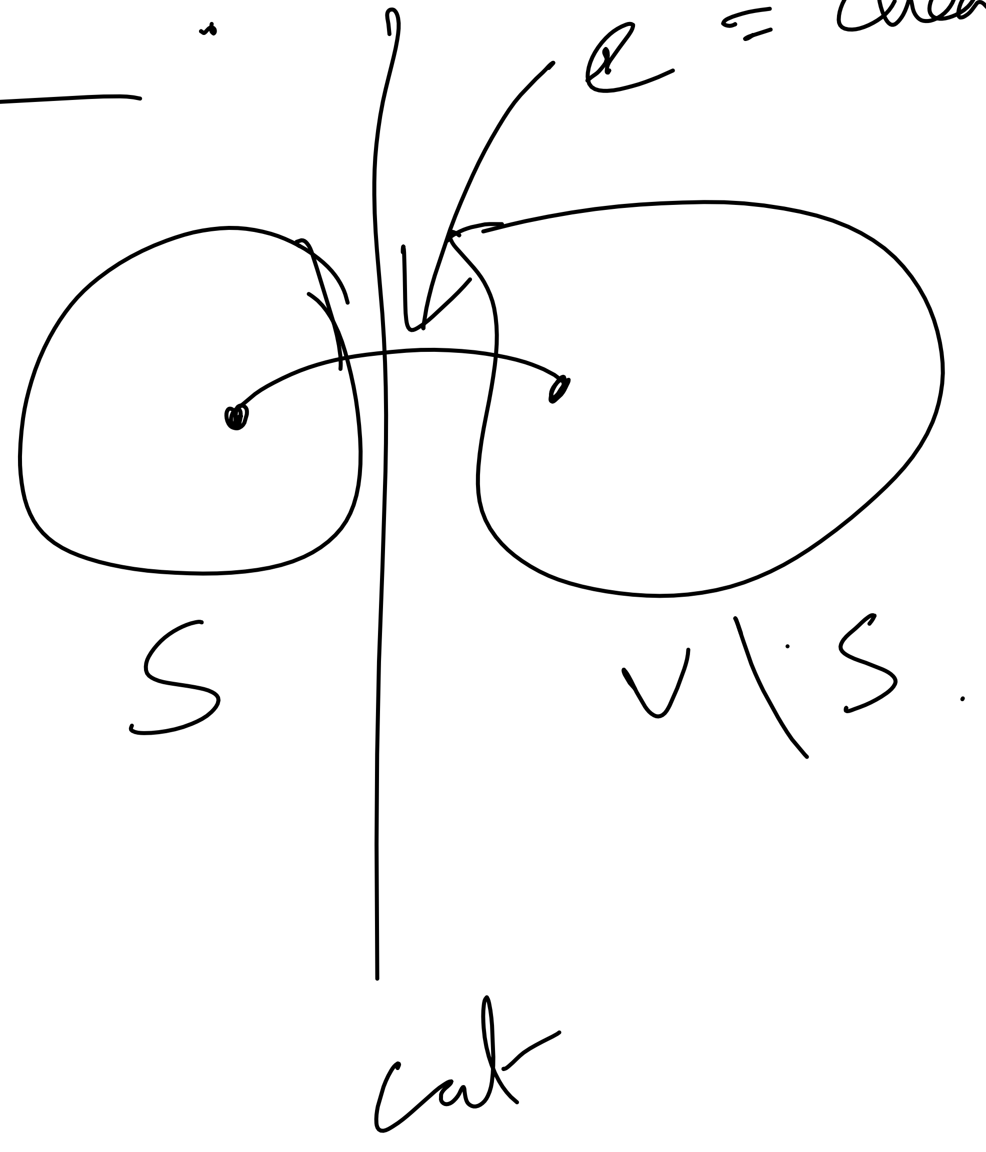


cut.

- undirected graph
- weight

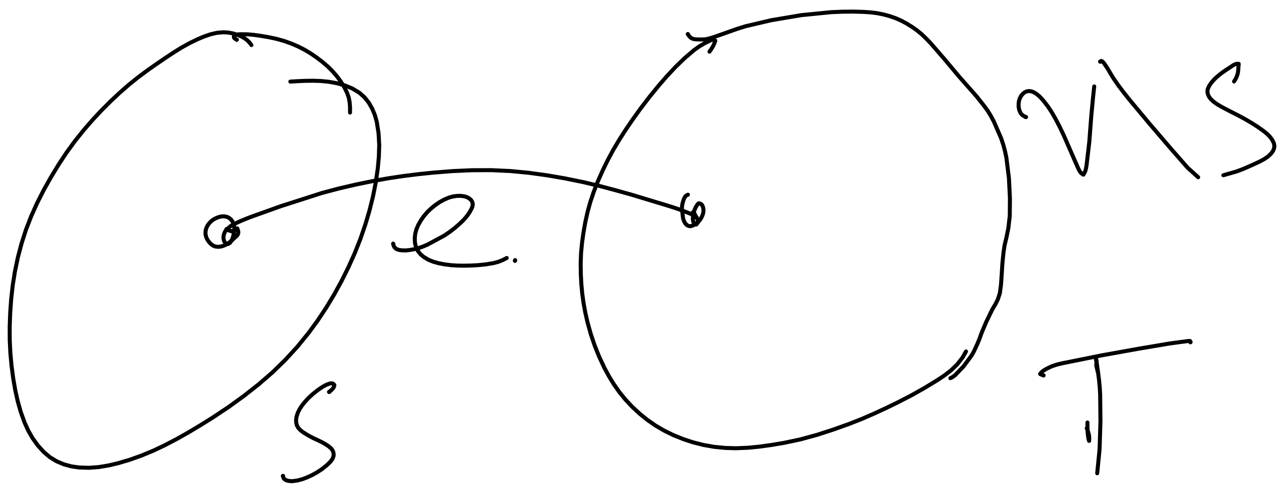
Cut Property of MST :

e = cheapest weight edges.



$G = (V, E)$

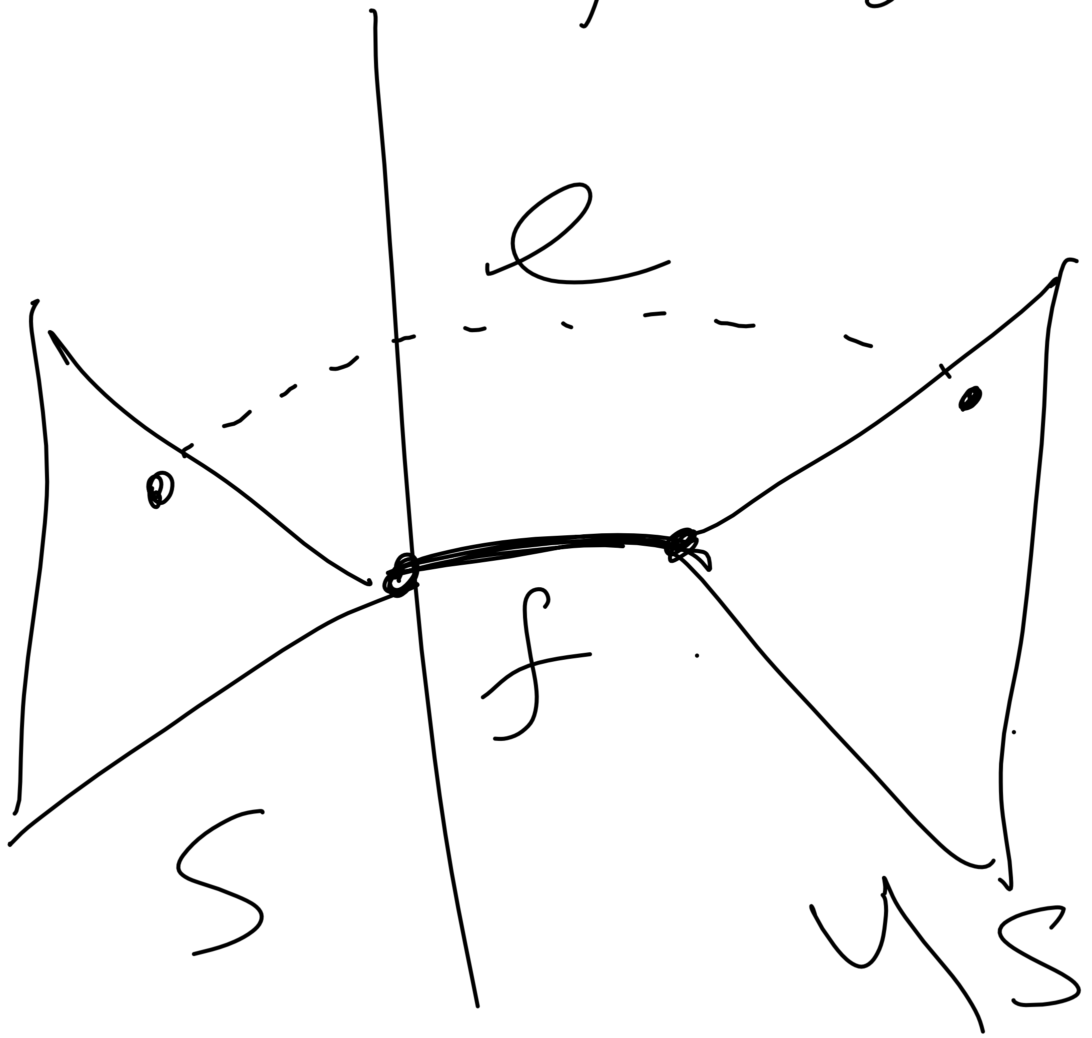
$\exists$ an MST T of G s.t. T includes e.



exchange argument.

T' be some mst.

T' :



$= \left((T' \cup \{e\}) \setminus \{f\} \right)$

$(n-1)$ edges, connected $\Rightarrow$ spanning tree.

Algo.

(1) pick any $v \in V$.

(2) $S = \{v\}$, $T = \emptyset$

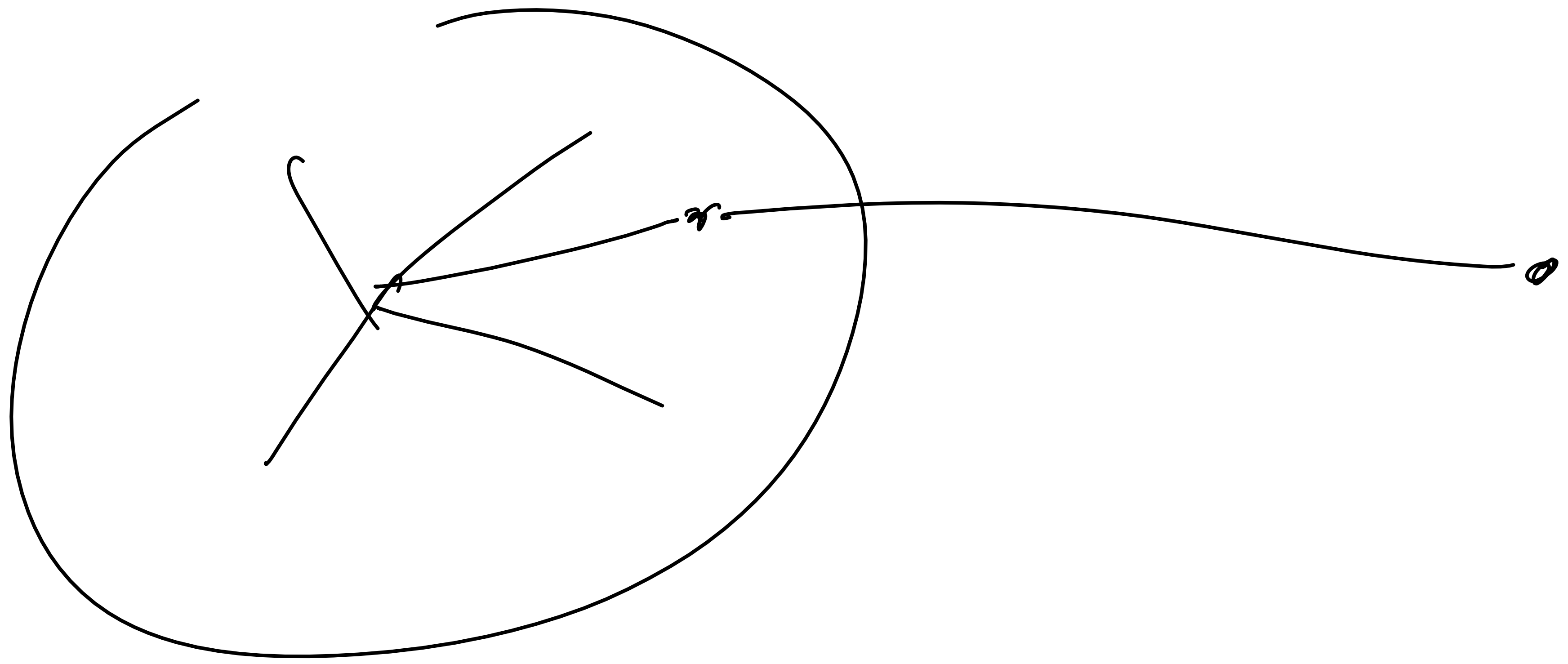
(3) pick lightest edge from $S \times \{V \setminus S\}$.

$\downarrow$

(x, y)

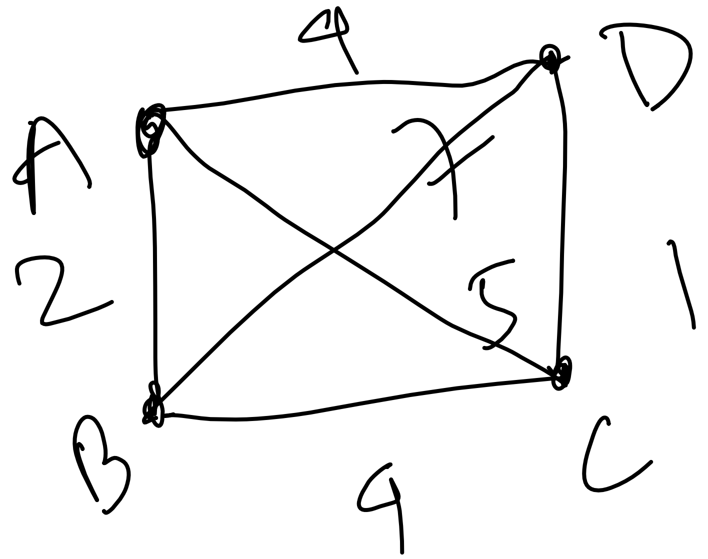
(4) $S \leftarrow S \cup \{x\}$, $T \leftarrow T \cup \{(x, y)\}$.

(5) go back to step 3 until $(n-1)$ edges are selected.



- connected
- $(n-1)$

$\Rightarrow$ ST.

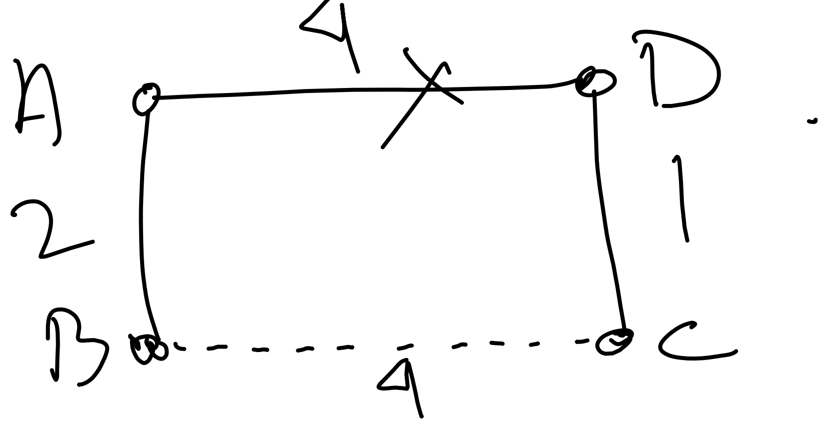


$e = (B, C) \quad f = (A, D)$

$S = \{A, B\}$

$V \setminus S = \{C, D\}$.

T



$\rightarrow$

① $S = \{A\}$, $V \setminus S = \{B, C, D\}$

$S = \{A, B\}$, $T = \{(A, B)\}$.

② . $S = \{A, B\}$, $V \setminus S = \{C, D\}$.

$S = \{A, B, D\}$, $T = \{(A, B), (A, D)\}$

③ . $S = \{A, B, D\}$, $V \setminus S = \{C\}$

$S = \{A, B, C, D\}$ $\boxed{T = \{(A, B)\ (A, D), (C, D)\}}$

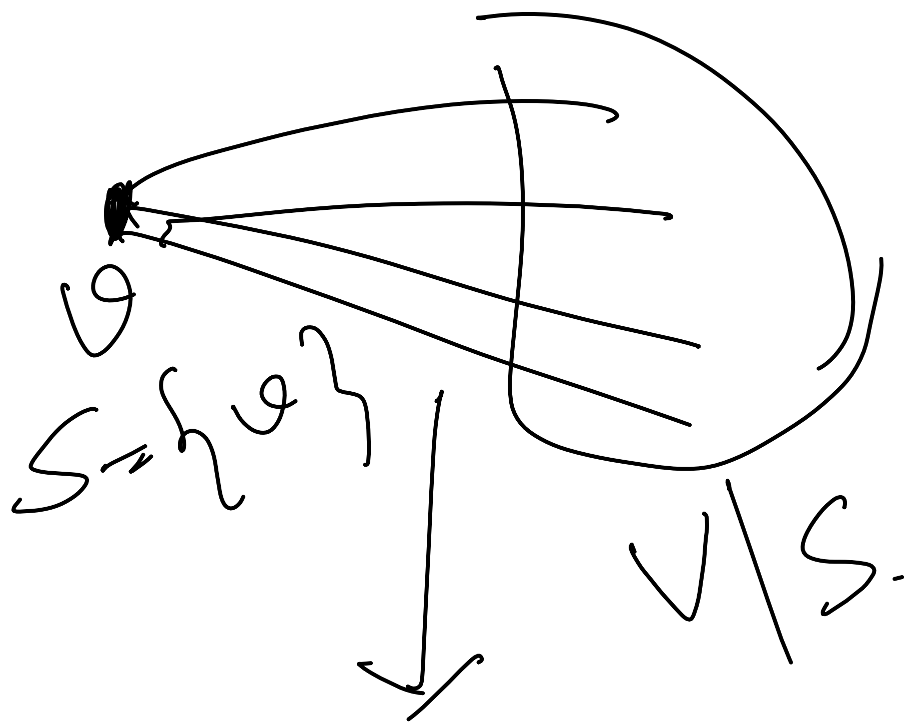
Correctness:

Spanning tree.

Claim: Let T_j be the grown tree at some intermediate iteration. (j-th iteration)

$\Rightarrow$ $\exists$ an MST of G that includes T_j

Proof: $j = 1$



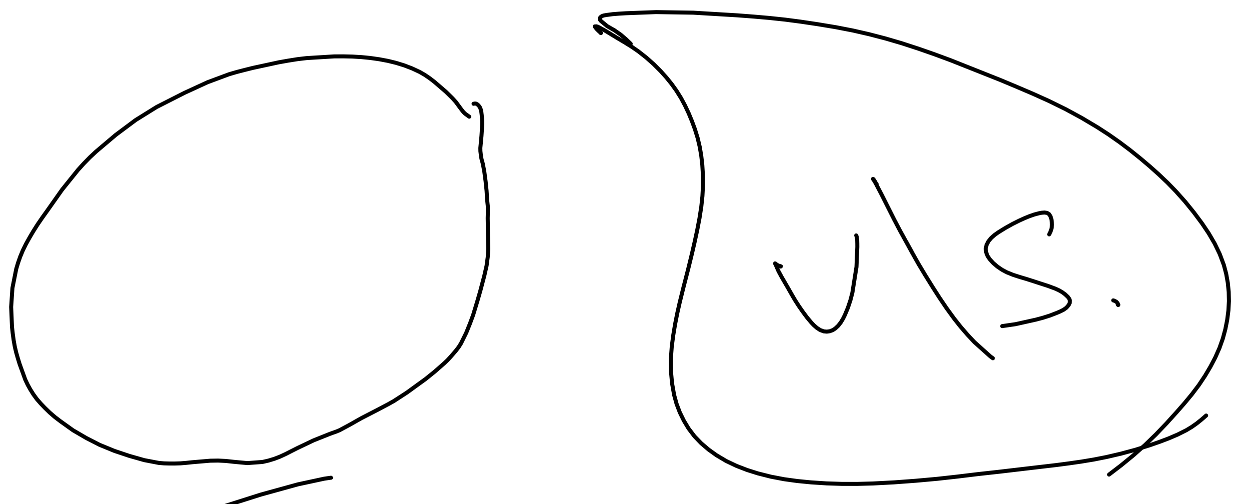
$\exists$ an MST that includes e.

e = cheapest weight edge selected.

$j=1$ is true from earlier discussion by choice $S = \{v\}$.

I.H.: T_j is indeed a **subtree** of ~~the~~ some MST.

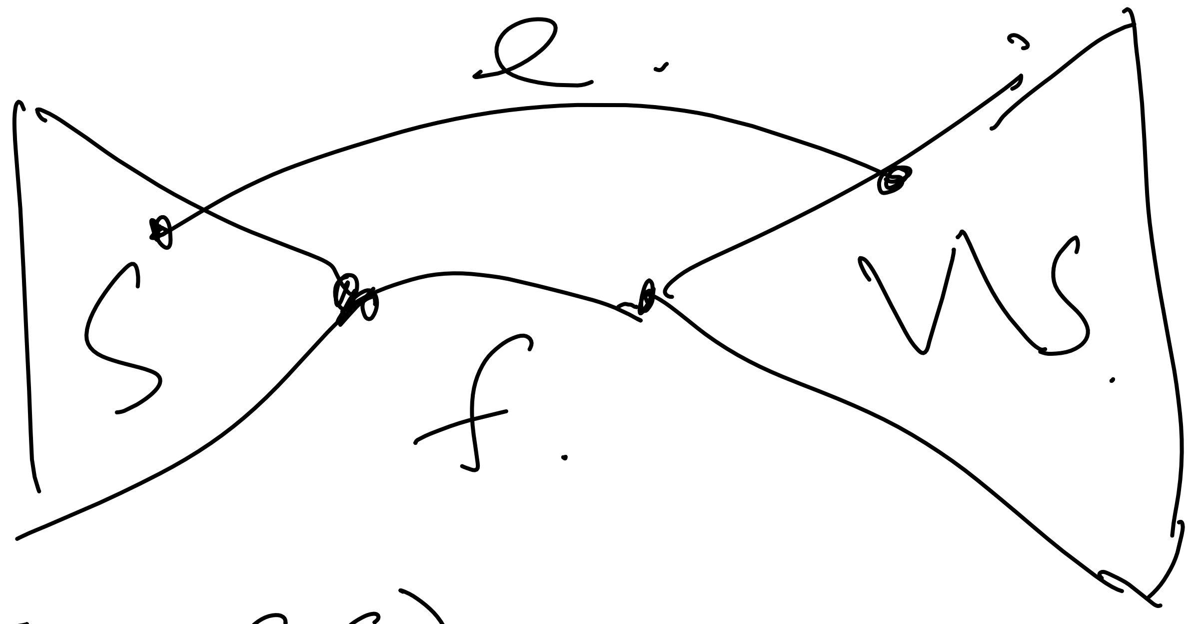
I.S. prove for $(j+1)$.



$S = T_j$ rest

Let T be the final min. spanning tree that includes T_j as a subtree.

T:



$\rightarrow$ $\boxed{T \setminus \{f\} \cup \{e\}}$, valid min. spanning tree.

$wt(e) \leq wt(f)$

$wt(e) \le wt(f)$

$\Rightarrow \underbrace{wt(T \cup \{e\} \setminus \{f\})}_{\text{mst also.}} \le \underbrace{wt(T)}_{\text{min wt}}$

$=$

$T_{j+1} = T_j \cup \{e\}.$

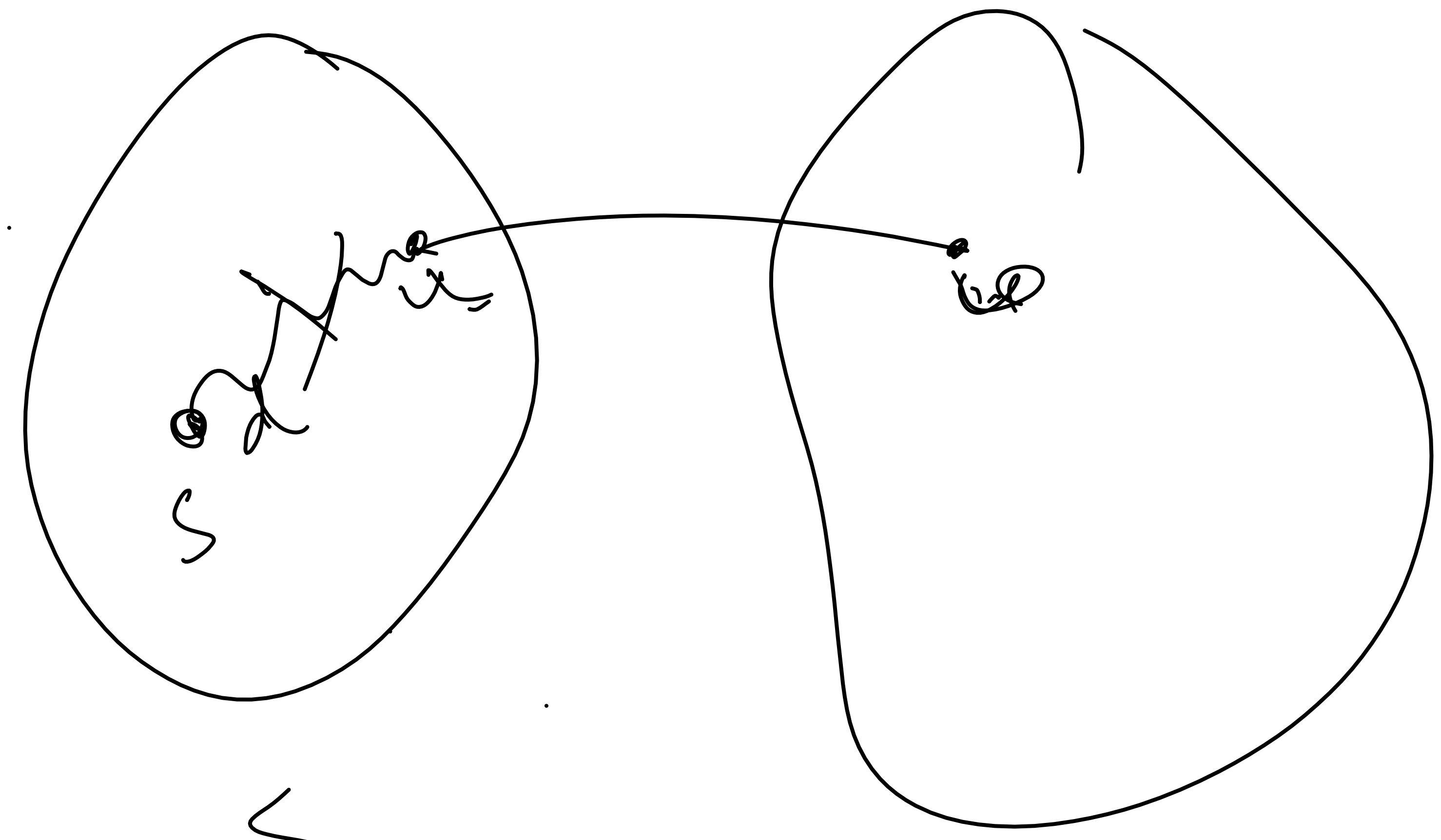
$T \cup \{e\} \setminus \{f\}$ — MST

— includes T_{j+1} as a subtree.

$\boxtimes$

Time Complexity

$$\Theta(V) \cdot \Theta(E) = \Theta(V \cdot E)$$



S $\qquad$ $V \setminus S$

$$\cancel{d(u)} + \boxed{wt(u,v)}$$

Pseudocode PrimMST(G)

// G =(V,E) is an weighted undirected graph.

PQ ← ~~empty~~ priority queue ~~of~~ of size |V|
with all keys = ∞.

T ← empty tree

v ← any node in V.

~~PQ~~ Decrease Key (PQ, v, 0).

π ← all NULL array.

while (PQ ≠ empty)

$u \leftarrow \text{ExtractMin}(PQ)$.

for all $(u, w) \in E$

  if $(wt(u,w) < \text{Key}(w))$.

    DecreaseKey $(PQ, w, wt(u,w))$.

    $\pi(w) \leftarrow u$.

end for.

$T = T \cup \{(u, \pi(u))\}$.

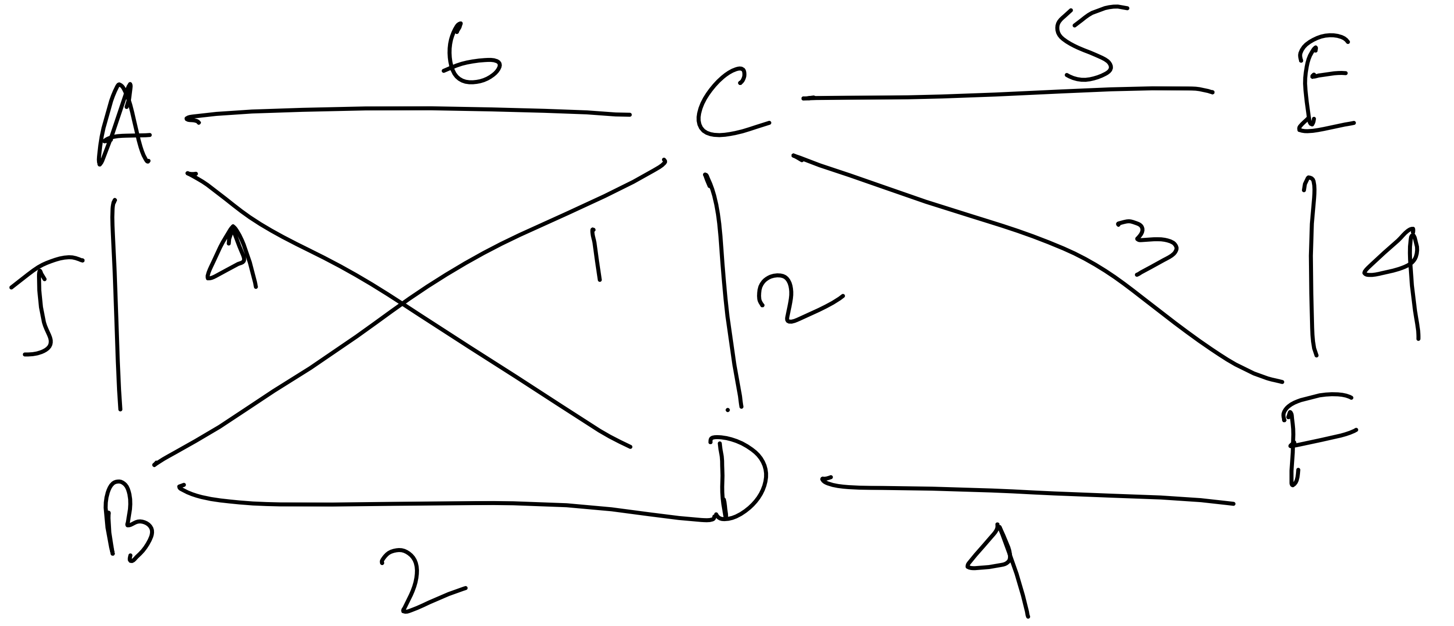
end while.

Return π.

Time Complexity : $O\left((m+n)\log n\right)$.

$\log n$ → Binary Min Heap.

Example :



	A	B	C	D	E	F		
π	NULL	NULL	NULL	NULL	NULL	NULL		
Key	0	∞	∞	∞	∞	∞		

- After Iteration 1 :

A is extracted.

B, C, D will get update.

	B	C	D	E	F	
Key	5	6	4	∞	∞	
π	A	A	A	NULL ~~∞~~	NULL	

- After Iteration 2 extract D.

$T\{(A,D)\}$.

	B	C	E	F
Key	2	2	∞	4
π	D	D	NULL	D

- <u>After Iteration 3</u>
  - extract B.
- $T = \{(A,D), (D,B)\}$

	C	E	F
Key	1	∞	4
π	B	NULL	D

- **After Iteration 4**

extract C ; $T = \{(B,C), (A,D), (D,B)\}$.

	E	F
key	5	3
π	C	C

- **After Iteration 5**

extract F , $T = \{(B,C), (A,D), (D,B), (C,F)\}$.

	E
key	4
π	F

- After Iteration 6

E is extracted.

~~final~~ Tree: $\underbrace{\{ (B,C), (A,D), (D,B), (C,F), (E,F) \}}$

MST

verify by running **Kruskal**:

— min. wt.