CSCE 750
11/14/2023



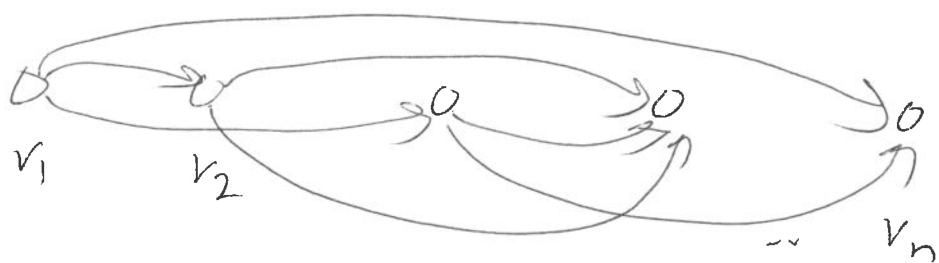
# Topological Sort (app of DFS)
# Minimum Spanning Tree

Def: Given a directed acyclic graph (dag) G, a topological sort of G is an arrangement

$v_1, \dots, v_n$ of the vertices of G

such that, for any edge $(v_i, v_j)$ in G, must have $i < j$.

all edges go left to right




Linear-time algo for Top sort:

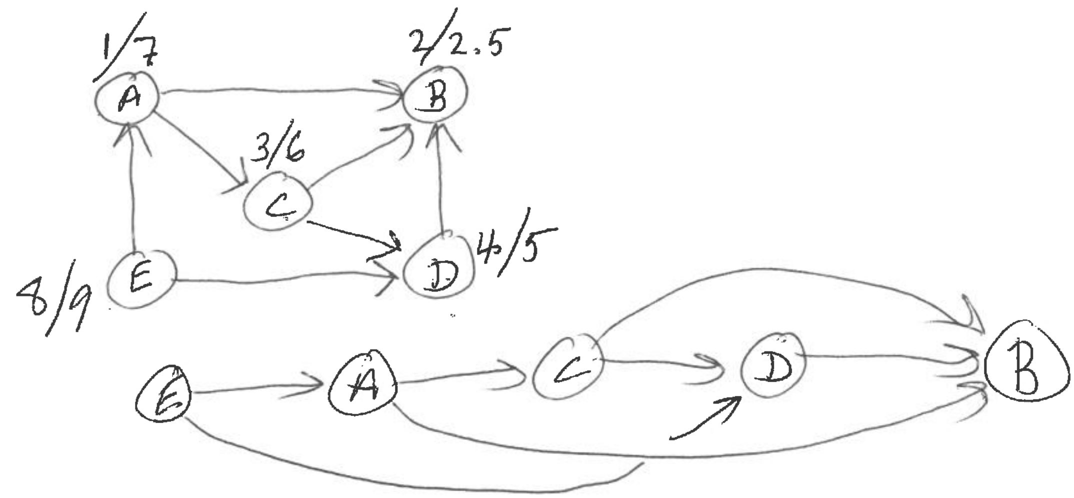
- Run DFS on G;
- when a vertex v is finished, add v to the front of a linked list.
- When done, list is top. sort of the vertices.

Correct?

Suppose list $v_1, \dots, v_n$ is not top sorted.

Then $\exists i > j$ such that $(v_i, v_j) \in G.E$

Then $v_j$ must finish before $v_i$ ↯ //




---

Minimum Spanning Tree (MST)

$G$ is a connected, undirected graph with edge weight function $w : G.E \to \mathbb{R}$

**Def:** A spanning tree of $G$ is a connected acyclic subgraph of $G$, (using all of $G$'s vertices).

**Fact:** Every spanning tree of $G$ has exactly $|G.V| - 1$ edges.

(Also Every subgraph of $G$ with $\geq |G.V|$ edges has a cycle.)

**Goal:** Find a MST for $G$, i.e., a spanning tree with min total weight.

# Generic MST algorithm

$T := \emptyset$

// Invariant: $T$ is a subset of an MST of $G$.

**while** there exists an edge $e \in G.E$ ($e \notin T$) that is *safe* to add to $T$, do

$T := T \cup \{e\}$

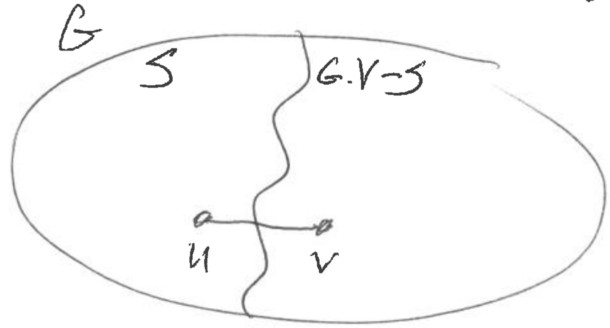
end-while

return $T$.

"Safe": $e$ is safe if $e \notin T$ and adding $e$ to $T$ maintains the invariant, i.e., $T \cup \{e\}$ is a subset of some MST.

Whole ~~the~~ trick: finding a safe edge.

**Def:** A *cut* in $G$ is a pair $(S, G.V - S)$ where $S \subseteq G.V$ is a nonempty proper subset of $G.V$.




An edge $e$ *crosses* cut $(S, G.V-S)$ if $e$ has one endpoint in $S$ and the other in $G.V-S$.

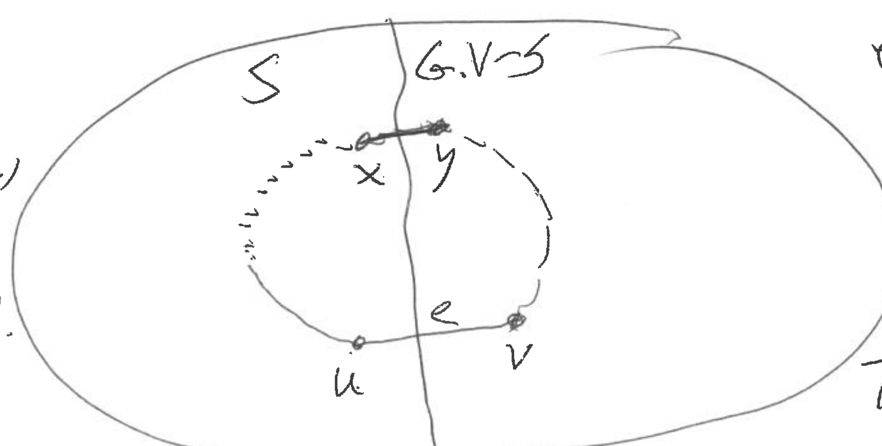
Say that a ~~set~~ cut respects a set of edges $T$ if no edge in $T$ crosses the cut.

Theorem: Let $T$ be a subset of some MST. Let $(S, G.V-S)$ be a cut respecting $T$, and let $e$ be an edge crossing $(S, G.V-S)$ of min weight (among edges crossing the cut). [Book: $e$ is a light edge]

Then $e$ is safe to add to $T$.

Proof: Let $T'$ be an MST such that $T \subseteq T'$. Let $(S, G.V-S)$ and $e$ be as above.
If $e \in T'$ then clearly $e$ is safe.
If $e \notin T'$: Let $e = (u,v)$. There is a unique path in $T'$ connecting $u$ with $v$.

There must exist an edge $(x,y) \in T'$ crossing $(S, G.V-S)$.




$e$ is a light edge, so $w(e) \leq w(x,y)$.
Let $T'' := (T' - \{(x,y)\}) \cup \{e\}$

Note: $T''$ is a spanning tree.

and $w(T'') = w(T') - w(x,y) + w(e) \leq w(T')$.

Since $T'$ is an MST, $T''$ is an MST (and $w(T'') = w(T')$)

But $e \in T''$ and $T \subseteq T''$ (because cut respects $T$)

$\therefore$ $e$ is safe. $\blacksquare$

2 MST Algos: Kruskal's algo and Prim's algo.

<u>Kruskal:</u>

1) Sort G.E is ascending order by weight.

1.5) $T := \emptyset$

2) repeat

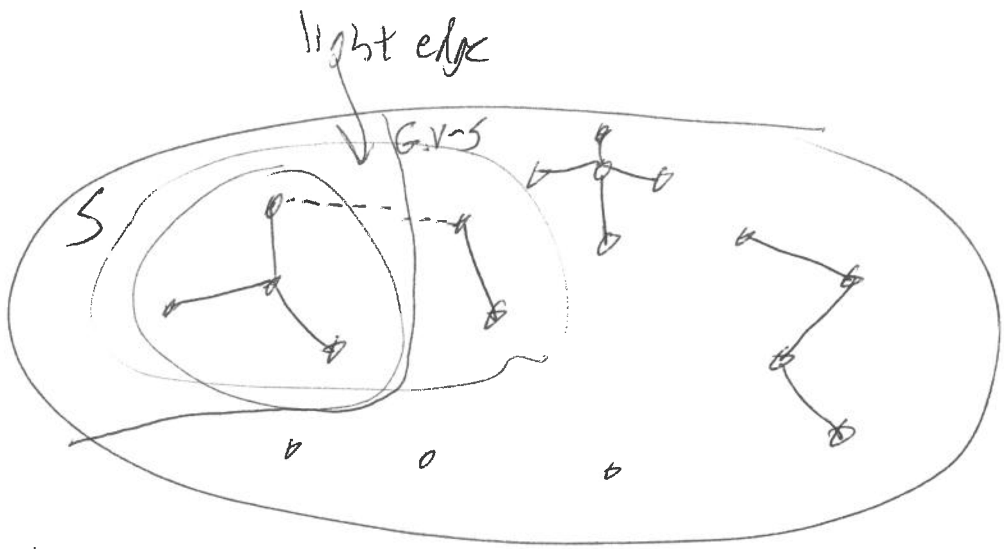
- let $e$ be the next edge in G.E is ascending order
- if adding $e$ to $T$ does not form a cycle, then

$$T := T \cup \{e\}$$

until $|G.V|-1$ many edges are added to $T$.

3) return $T$.

<u>Correctness:</u>




Doing the test: Use disjoint set system of vertices.
Sets are the connected components
deciding if two endpoints are in the same component:
2 Find operations
adding the edge merges the two connected components into: Union operation.

Run time:

$$V-1 \le E \le V^2$$

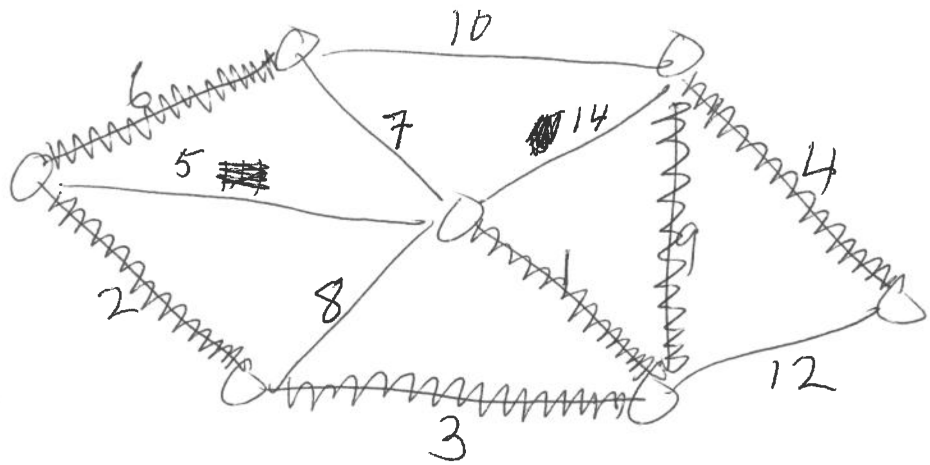
sorting the edges: $\Theta(E \lg E) = \Theta(E \lg V)$

do 2 Finds & a Union

| 2 E finds | V-1 many |
|---|---|
| $\Theta(E\alpha(V))$ | $\Theta(V\alpha(V))$ |
| $=O(E \lg V)$ | $=O(V \lg V) = O(E \lg V)$ |

Total: $\Theta(E \lg V)$

Example:




$w(T) = 1 + 2 + 3 + 6 + 9 + 4$
$= \boxed{25}$

---

Prim: start with a source vertex $s$ (arbitrary)