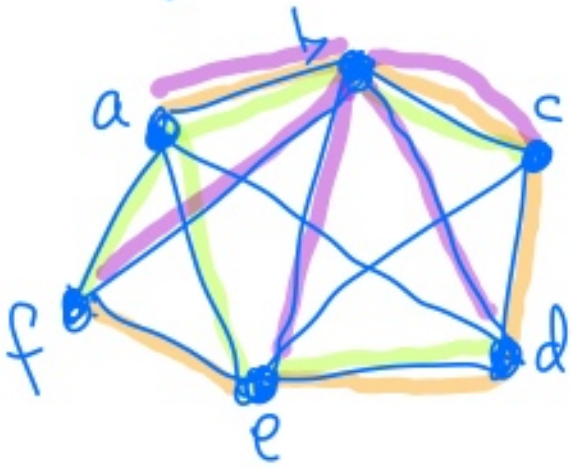
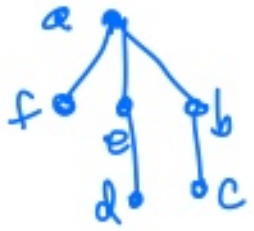
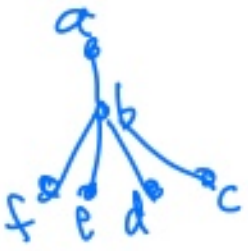
A spanning tree of a connected graph is a subgraph that is a tree that contains all the vertices of the graph.

Example: Find a spanning tree of this graph (starting with a).



a→b→c→d→e→f





Standard Algorithms for finding spanning trees

① Depth-First Search Algorithm.

- start at a given vertex
- go to an adjacent vertex
- keep doing this, making sure not to visit a vertex a second time.
- When this process stops, either all the vertices are in the tree, or not
- Back up one vertex, restart from that vertex to add more edges.

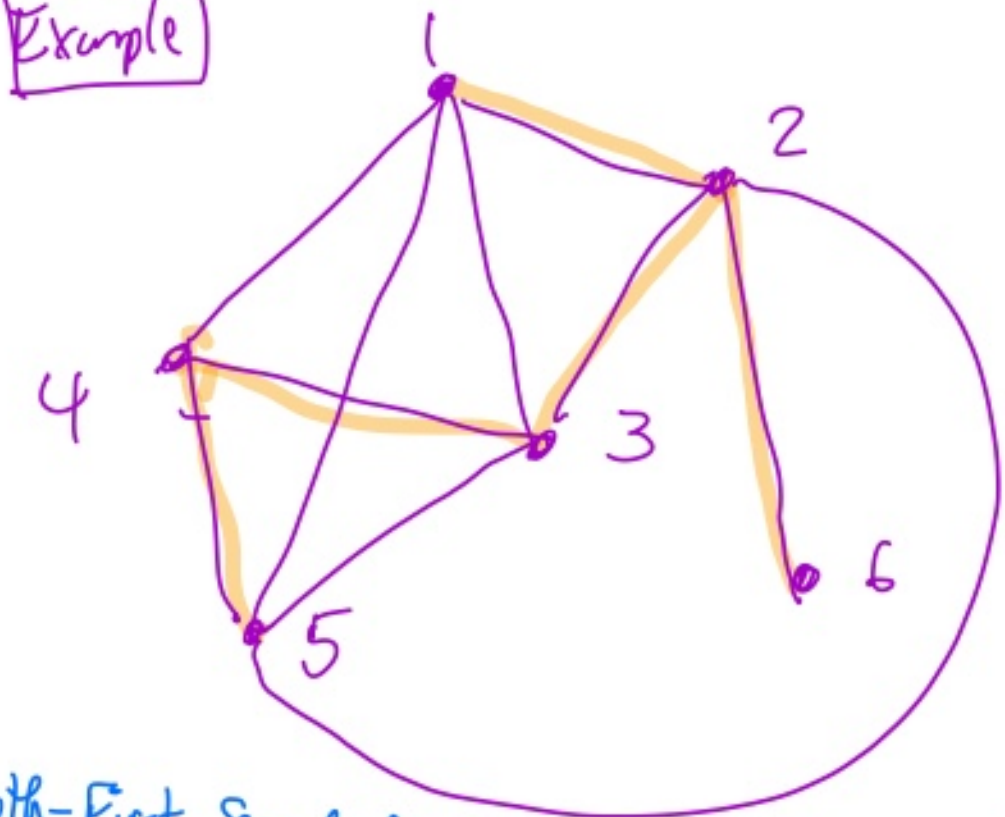
└ keep going until all vertices are visited.

(Thm: This works.)

Question: If the graph has n vertices, how many ~~steps are there~~ edges have we chosen? $(n-1)$

(Because every tree with n vertices has $n-1$ edges.)

Example



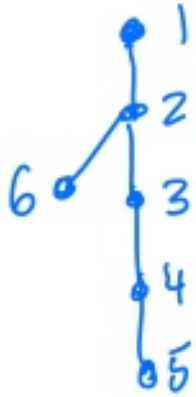
Depth-First Search (and, we'll always pick the smallest # if there is more than one choice).

$1 \to 2 \to 3 \to 4 \to 5$ stop

4 stop

3 stop

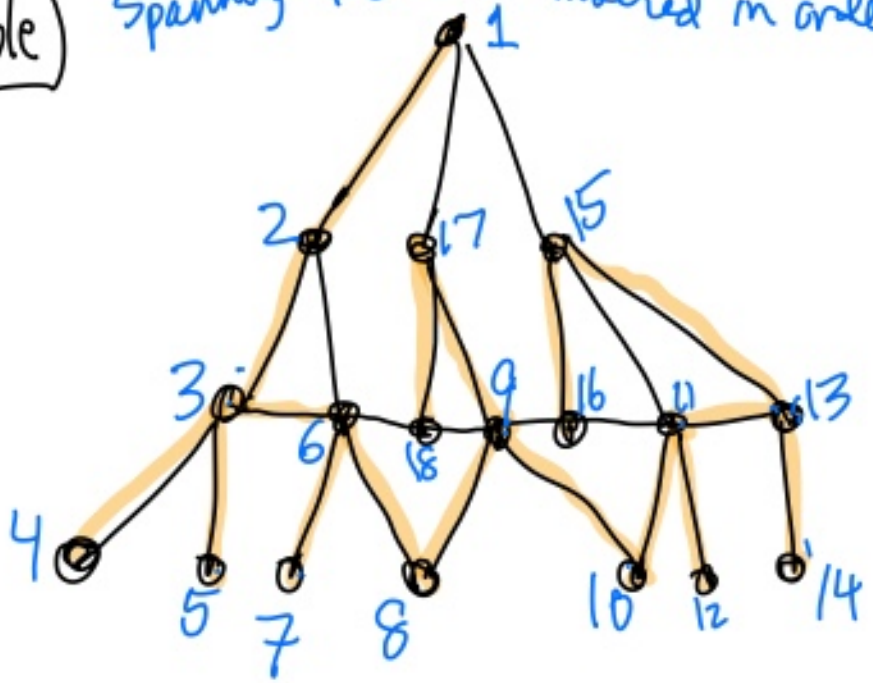
$2 \to 6$

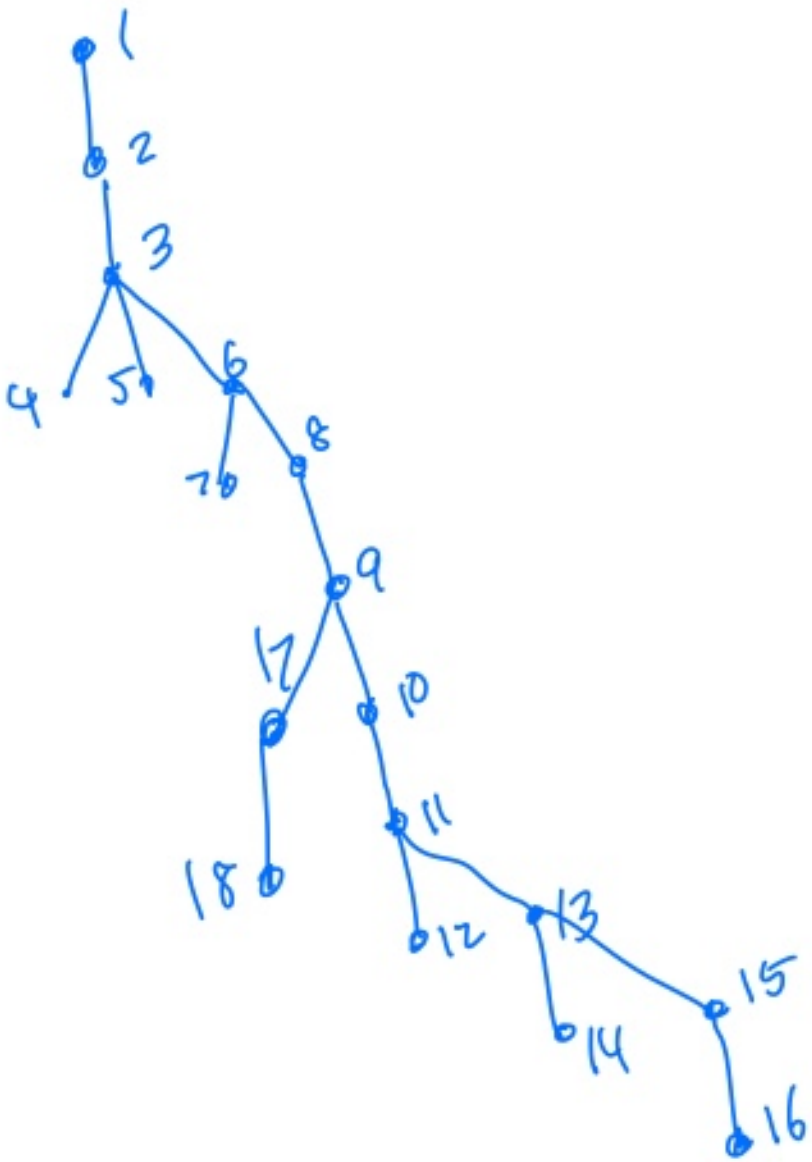


our spanning tree.

Example

Spanning tree — numbered in order.

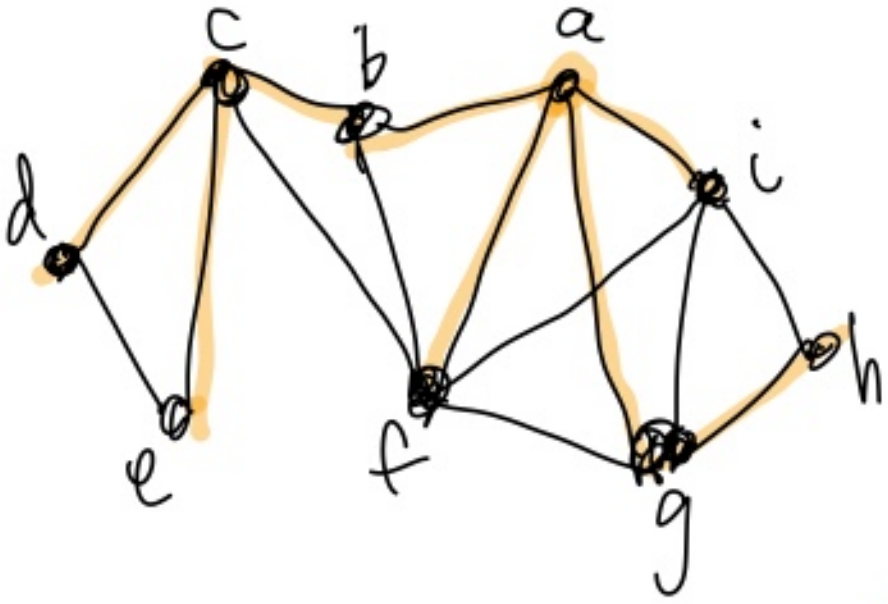




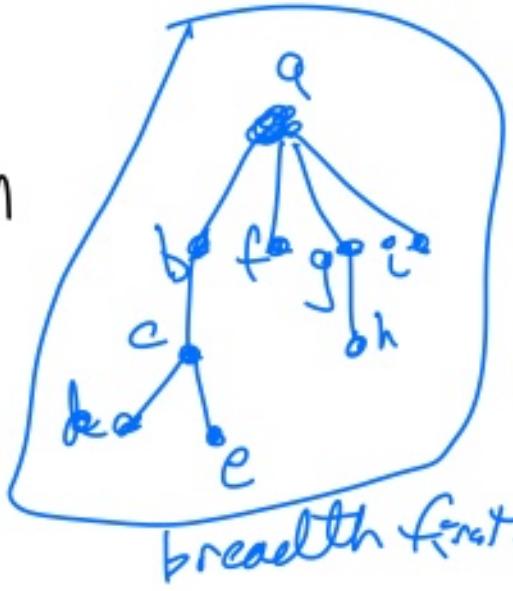
(2) Breadth-First Algorithm.

- Start at a vertex
- add all possible edges adjacent to this (making sure no circuits are created - vertices are not visited a second time.)
- Keep going with the vertices that have just been added.

(it's a theorem that this algorithm creates a spanning tree.)



Starting with a:



breadth first.

Weighted Graphs: Searching for minimal spanning trees

↑ a spanning tree that minimizes the sum of the weights.

Prim's Algorithm.

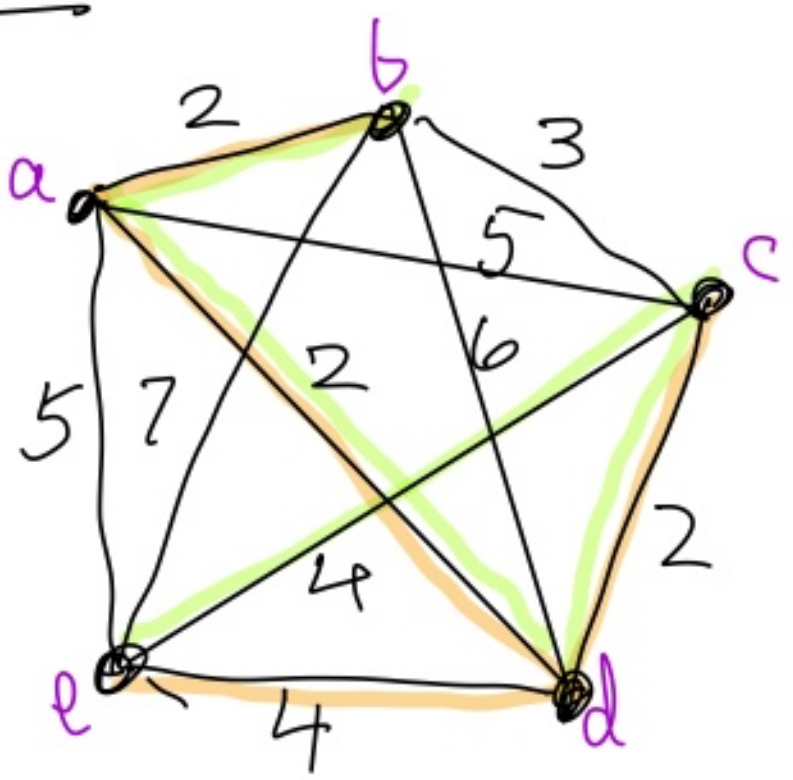
1. Select edge of minimum weight.
2. Attach an incident edge to our tree (that does not create a circuit, has minimum weight among all possibilities.)
3. Keep doing this – stop after we have $(n-1)$ edges.

Kruskal's algorithm:

Same as Prim's, but don't worry about the edge being incident to the tree we are making.

Both algorithms produce an optimal solution!

Example



Prim: $\{a, b\}, \{a, d\}, \{c, d\}, \{d, e\}$,

sum = 10

Kruskal: $\{c, d\}, \{a, b\}, \{a, d\}, \{e, e\}$

sum = 10