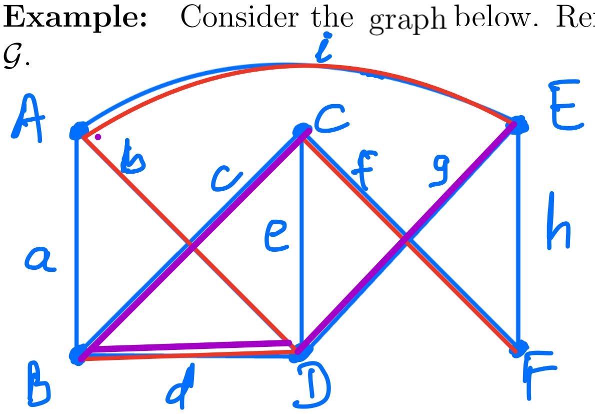
What if we have lots of edges but want to use a minimal number of edges to be able to reach everywhere?

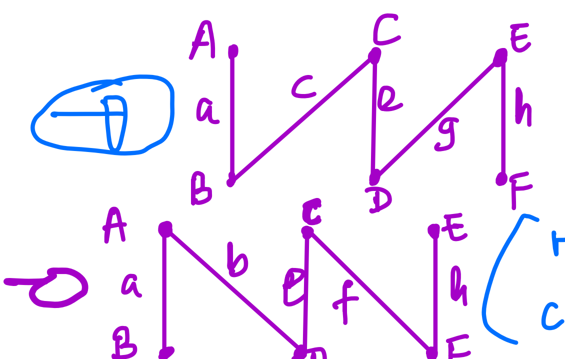
We can remove edges until we have a (spanning) tree, since trees are connected and non-redundant (no cycles).

Definition:

> A <u>spanning tree</u> of a graph $\mathcal{G}$ is a tree formed by using edges and vertices of $\mathcal{G}$ containing all vertices of $\mathcal{G}$.

Example: Consider the graph below. Remove edges as appropriate to make a spanning tree of $\mathcal{G}$.



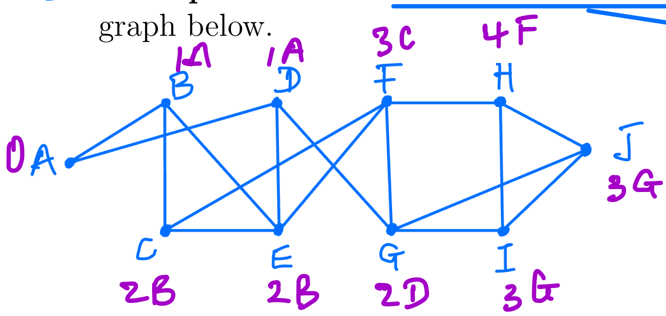


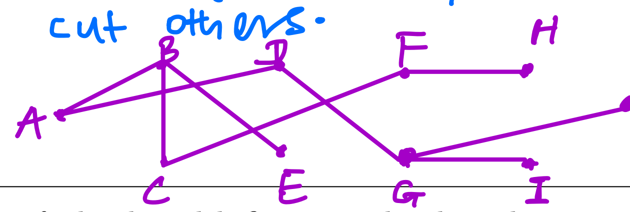
(remove b, d, f, i)

(remove c, g, d, i)

Example: Use the <u>Breadth-First Search Algorithm</u> to help you create a spanning tree for the graph below.

tells you distance from 1 vertex to others and tell predecessor
- here we will keep the edges used to make predecessor and cut others.





Definition:

> A spanning tree constructed by means of the breadth-first search algorithm is called a <u>shortest path</u> tree.

Theorem 5.6:

> A graph is connected if and only if it has a spanning tree.

Why should this make sense?

- If there is a spanning tree, then there must be a way to reach all vertices (trees are connected)
- If the graph is connected, then using Breadth-First Search Algorithm permits creation of a spanning tree (tree containing all vertices).

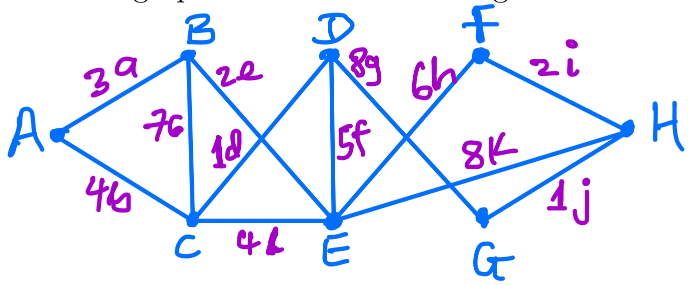
Definition:

In a weighted graph, the weight of a tree is the sum of the weights of the edges in the tree. A minimal spanning tree in a weighted graph is a spanning tree for which the weight of the tree is as small as possible. A maximal spanning tree in a weighted graph is a spanning tree for which the weight of the tree is as large as possible.

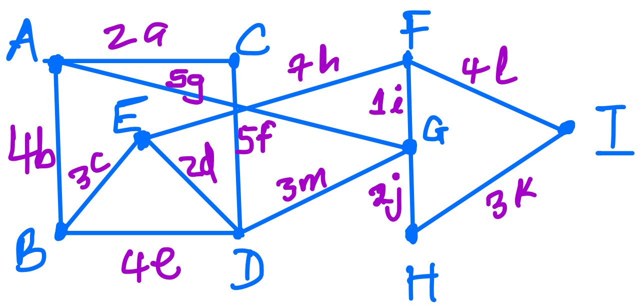
How can we approach creating a minimal spanning tree or maximal spanning tree? Prim's algorithm

summary for minimal: choose a starting vertex, pick an edge of of smallest weight with one vertex used and one not used, add the edge and vertex to the used pile, repeat until not possible to choose new vertices. If you don't reach all vertices, the graph is not connected/no minimal spanning tree exists. (for maximal, replace least weight with most weight)

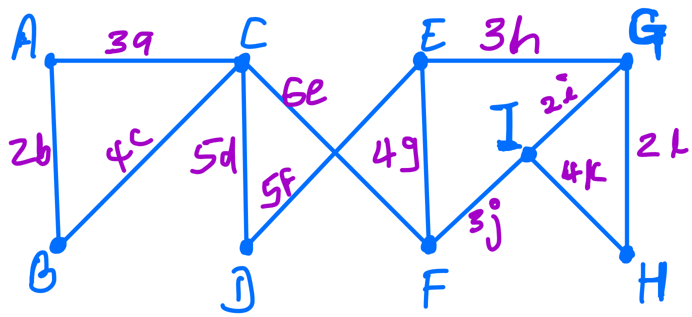
Example: Use Prim's algorithm to find a minimal spanning tree and maximal spanning tree for the graph below. Give the weight of each. (starting from A)



Example: Use Prim's algorithm to find a minimal spanning tree and maximal spanning tree for the graph below. Give the weight of each. (starting from A)



Example: Use Prim's algorithm to find a minimal spanning tree and maximal spanning tree for the graph below. Give the weight of each. (starting from A)

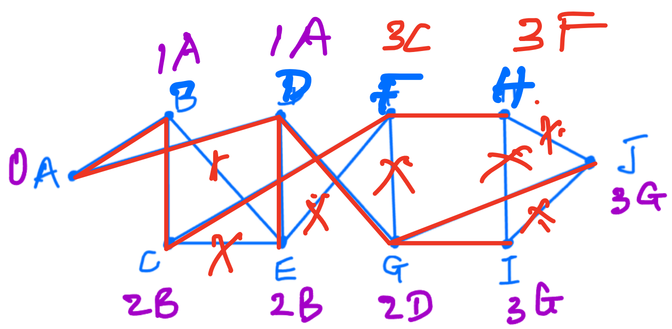


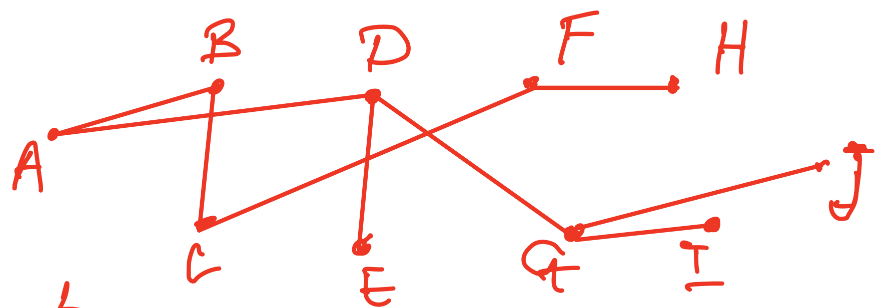
FRIDAY

DECEMBER 15

8:00 — 9:50AM

$\boxed{\text{A100}}$

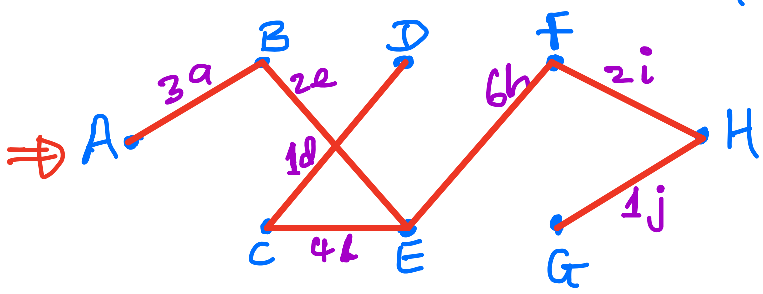


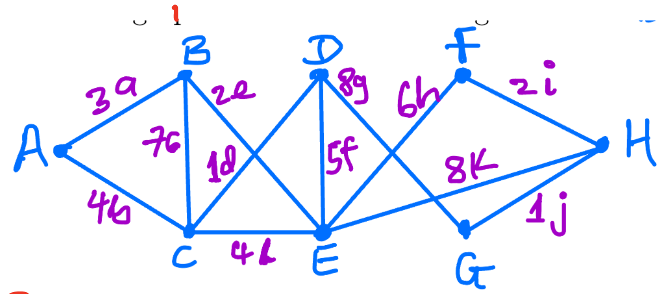


Breadth-first search

Prim's algorithm

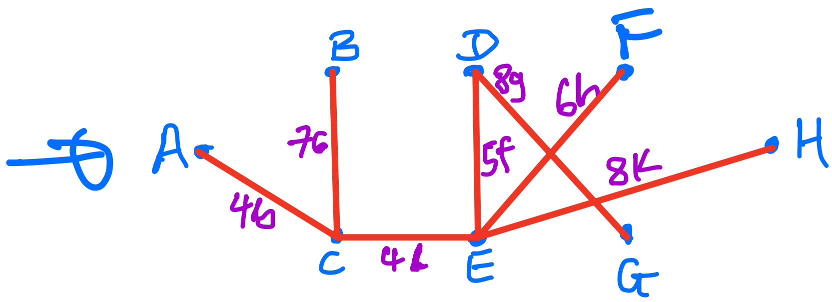
PRIM's ALGO.





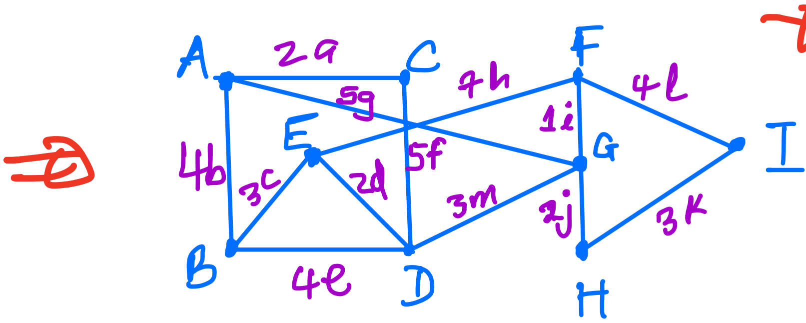
Minimal spanning tree

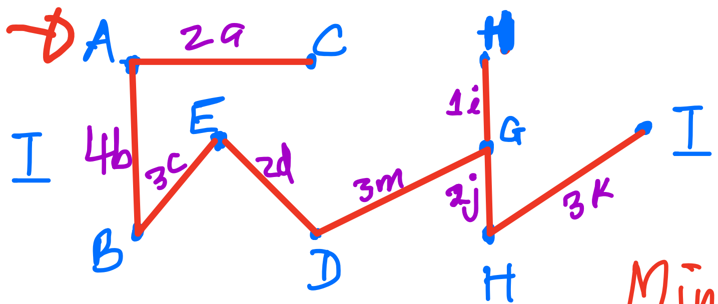
$\underline{3+2}+\underline{4+1}+\underline{6+2+1} = 19$



Maximal spanning tree

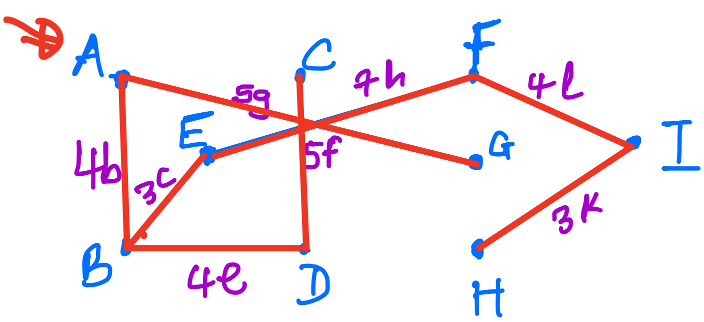
$\underline{4+7}+\underline{4+5}+\underline{8+8}+\underline{6} = 42$





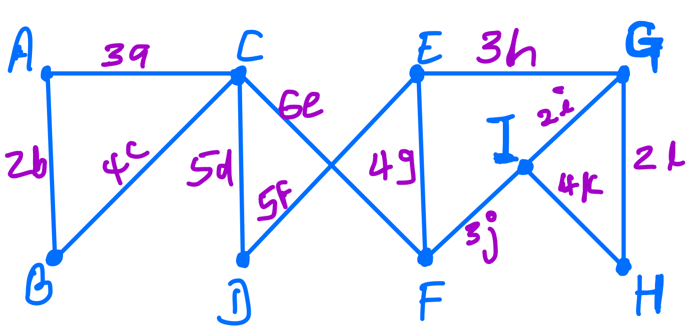
$\underline{2+4+3+2+3+1+2+3}$

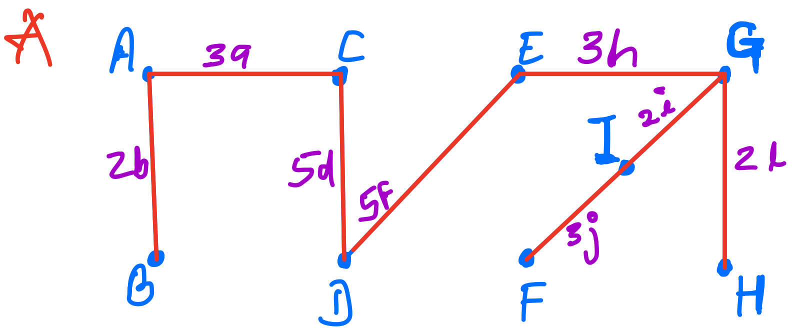
Minimal spanning tree



maximal spanning treee

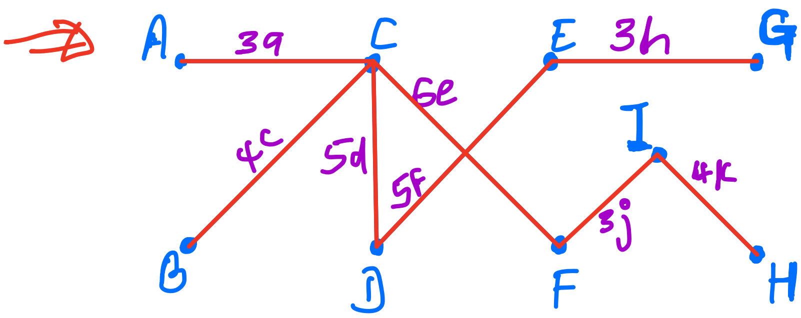
$\underline{4+4+3+5+5+7+4+3}$





minimal spanning tree

$2+3+5+5+3+2+2+3$



maximal spanning tree

$3+4+5+5+6+3+3+4$