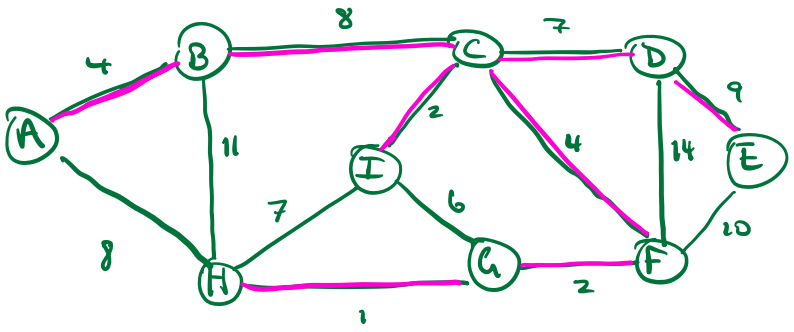
Minimum Spanning Trees



MST wl cost 37.

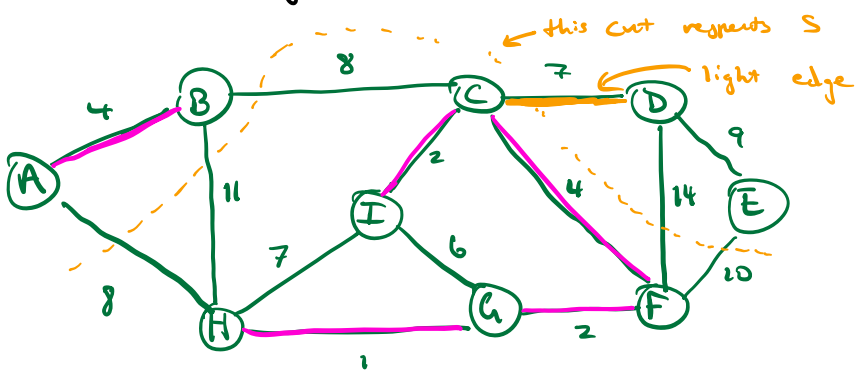
Def. A spanning tree is a tree that connects all the vertices in an undirected graph.

A minimum spanning tree is a spanning tree with min. cost.

Cuts in graphs.

Def. a cut of a graph is a partition of the graph into two sets of vertices (no requirement on connectedness).

Def. Let S be a set of edges in G; a cut <u>respects</u> S if no edges in S cross a cut



Def. an edge crossing the cut is called light if it has the smallest weight of any edge crossing the cut

Finding an MST.

Lemma: Let S be a set of edges and consider a cut that respects S

Suppose there is an MST containing S

Let $\{u, v\}$ be a light edge

Then there is an MST containing $S \cup \{u, v\}$

Prim's Algorithm

Greedily add shortest edges we can to grow tree

Implementation 1:

```
slow Prim (G = (V, E), starting vertex s):
    Let (s,u) be lightest edge out of s
    MST = {(s,u)}
    visited = {s, u}
    while |visited| < |V|:        <- |V| iters
        find lightest edge {x, v} ∈ E   <- |E| iters
            where x ∈ visited, v ∉ visited
        add {x, v} to MST
        add v to visited
    return MST
```

Runtime: $O(|V|\,|E|)$

Implementation 2: Every vertex has key + parent <- like Dijkstra's

Until all vertices are reached:

Activate unreached vertex u w/ smallest key

For each of u's unreached neighbors v:

$k[v] = \min(k[v], w(u,v))$

if $k[v]$ updated, $p[v] = u$

Mark u as reached, add $\{p[u], u\}$ to MST

Runtime: $O(|E| + |V| \log |V|)$ using Fib-Heap.