

**Minimum Spanning Trees**

- $G = (V, E)$  is an undirected graph with non-negative edge weights  $w: E \rightarrow \mathbb{Z}^+$
- We assume wlog that edge weights are distinct
- A **spanning tree** is a tree with  $V - 1$  edges, i.e. a tree that connects all the vertices.
- The total cost (weight) of a spanning tree  $T$  is defined as  $\sum_{e \in T} w(e)$
- A **minimum spanning tree** is a tree of minimum total weight.

Jan 24-10:59 AM

lem The min wt edge crossing any cut is in the MST (edge wts are unique)

**PF** Suppose not. Then there is a MST  $T$  and a cut  $(S, \bar{S})$  s.t. the min wt. edge crossing the cut is not in  $T$ .

$T' = T - (u,v) + (x,y)$  is a spanning tree

$(x,y)$  min wt edge crossing cut  
 $(u,v)$  another edge crossing cut on  $(x,y)$  cycle.

$w(T') = w(T) - w(u,v) + w(x,y) < w(T)$ , contradicts  $T$  being min.

Jan 24-11:21 AM

**Prim's Algorithm**

**Idea:** Grow the MST from one node going out

**Need:** a data structure to maintain the edges crossing the cut, and choose minimum. We will maintain, for each vertex, the minimum weight incident edge crossing the cut

- INSERT( $v$ ) puts  $v$  in the structure
- EXTRACT-MIN() finds and returns the node with minimum key value
- DECREASE-KEY( $v, w$ ) updates (decreases) the key of  $v$

```

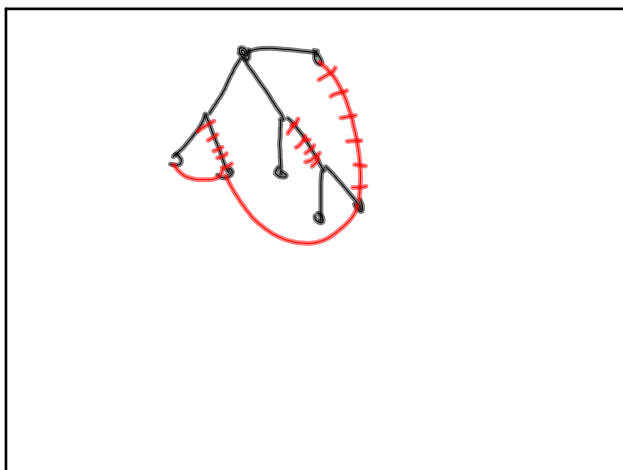
MST-Prim( $G, w, r$ )
1 for each  $u \in V[G]$ 
2   do  $key[u] \leftarrow \infty$ 
3
4  $key[r] \leftarrow 0$ 
5  $Q \leftarrow V[G]$ 
6 while  $Q \neq \emptyset$ 
7   do  $u \leftarrow$  EXTRACT-MIN( $Q$ )
8     for each  $v \in Adj[u]$ 
9       do if  $v \in Q$  and  $w(u,v) < key[v]$ 
10          then  $key[v] \leftarrow w(u,v)$ 
11
    
```

Handwritten notes in blue: n INSERT, n EXTRACT-MIN, m DECREASE-KEY

Jan 24-11:59 AM

**Example**

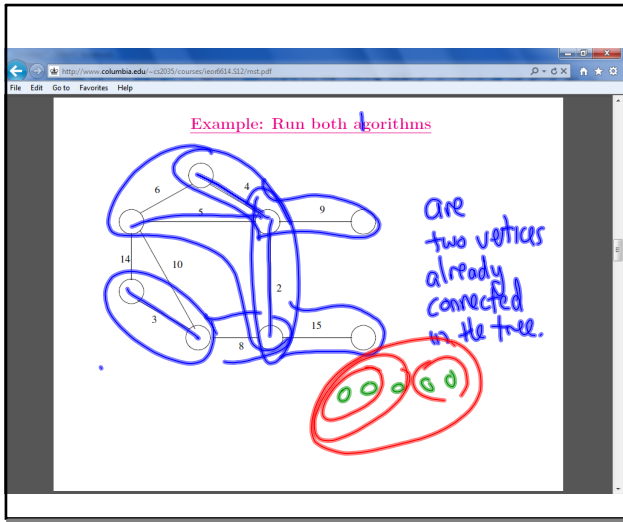
Jan 24-12:01 PM



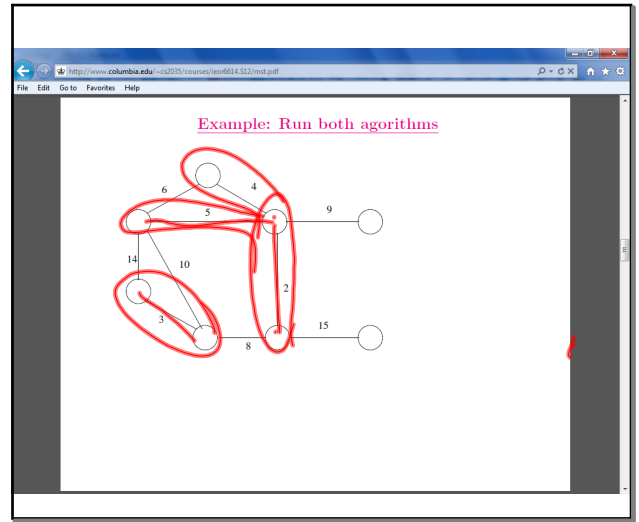
Jan 24-11:27 AM

		<u>Heap</u>	<u>Fib Heap</u>
n	INSERT	$\lg n$	$\lg n$
n	E-M	$\lg n$	$\lg n$
m	DK	$\lg n$	$\lg n$
		$O(m \lg n)$	$O(mn \lg n)$

Jan 24-11:27 AM



Jan 24-12:10 PM



Jan 24-12:14 PM