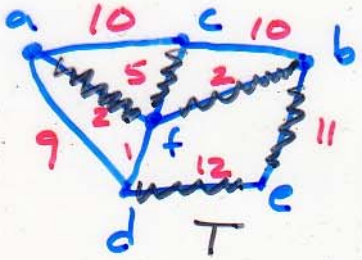
CS 223 Lec 19

MST cont.

$G = (V, E)$

weighted

$w : E \to \mathbb{R}$



edges in a spanning tree

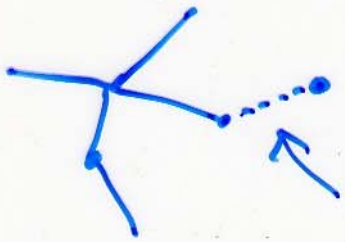
= # vertices in $G - 1$

$$w(T) = \sum_{(u,v) \in T} w(u,v)$$

$w(T) = 2 + 5 + 2 + 11 + 12$

$= 32$

"safe edge" : an edge that can be safely added to a partial MST



a partial MST

safe edge (when added can still find a MST)

GENERIC - MST algorithm

$A = \emptyset$

while $A \neq$ spanning tree {

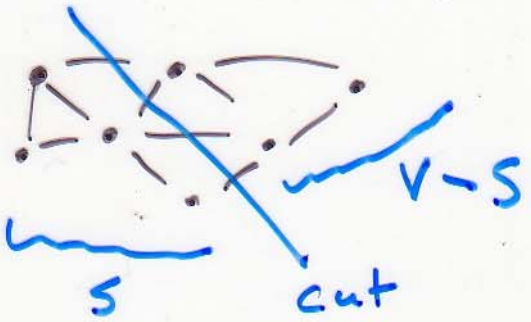
- find a new safe edge (u, v)
- add (u, v) to A

}

return A // A is a MST

Prim's Algorithm

"cut" in a graph



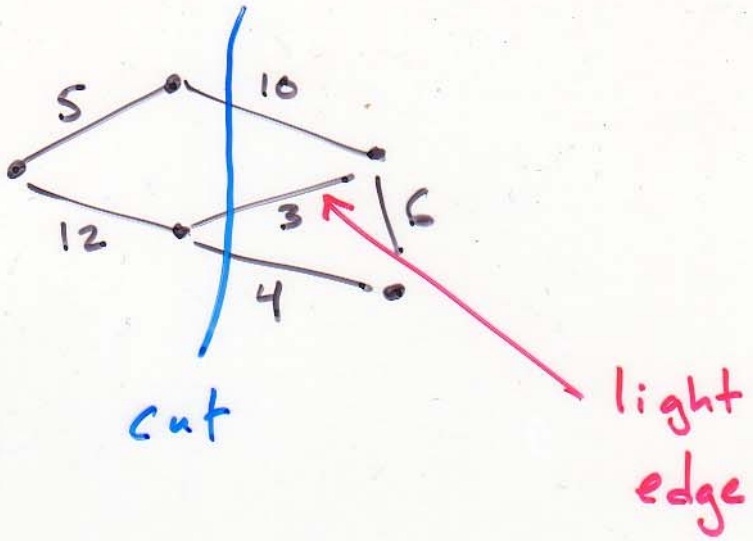
$(S, V-S)$ describes a cut in a graph

- edges (u,v) may cross a cut:

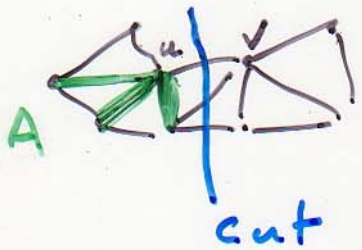
$\Leftrightarrow$ $u \in S$

$v \in V-S$

- We say an edge is "light" for a cut if it crosses the cut and has the minimum weight of all crossing edges.



Thm Given $G = (V, E)$
Let A be a partial MST
Let $(S, V-S)$ be a cut that respects A (no edges in A cross the cut)



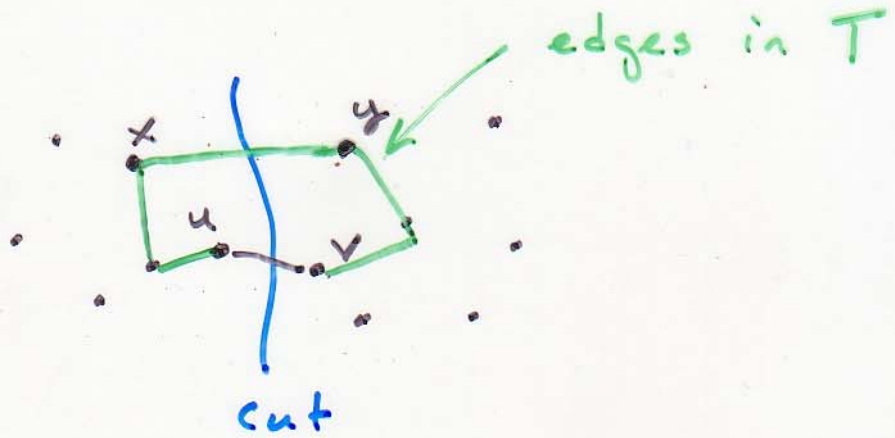
Let (u,v) be a light edge
Then (u,v) is a <u>safe edge</u>.

Pf

Let T be a MST that contains A.

If $(u,v) \in T$ then (u,v) is (by definition) safe.

If $(u,v) \notin T$:



Let $T' = T + (u,v) - (x,y)$

T' is still a spanning tree

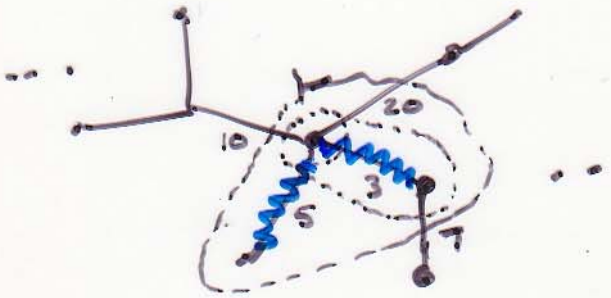
$w(u,v) \leq w(x,y)$ (since (u,v) light)

$\Rightarrow w(T') \leq w(T)$

$\Rightarrow T'$ is a MST

$\Rightarrow (u,v)$ is a safe edge to add to A.

Prim's algorithm: "grow-up" a MST



key(v) = distance (through) to the "shell" to the current partial MST.

⇒ use a priority queue to find the next vertex to add ...

Depth First Search in Graphs

- like DFS in binary trees...

Color Scheme

white → v is undiscovered

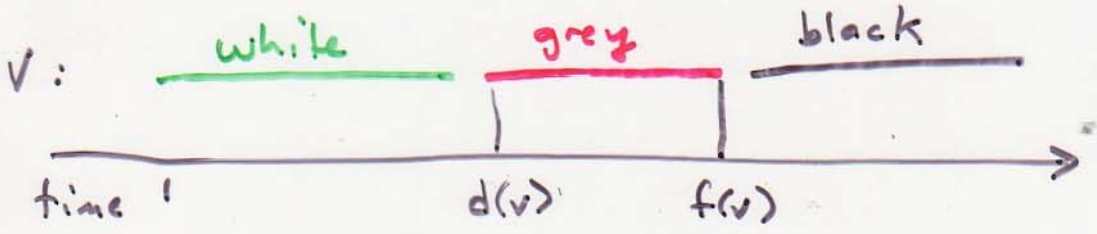
grey → v is discovered

black → v is finished

time stamps

d[v] = discovery time of v

f[v] = finishing time of v (just before v turns black)



DFS(G) ← G is a graph

DFS-Visit(u) ← visits and searchs from a vertex u