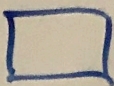


Union Find maintains a set for each connected component in T

— structure initially is empty: 

→ start at top of list with $e(0,1)$

→ Vertices 0 and 1 both not in structure,

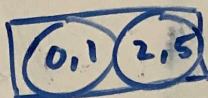
O.K. to add to MST.

try to add:

→ $e(2,5)$

Both V's not in set, add as new component

→



O.K. to add $e(2,5)$ to MST.

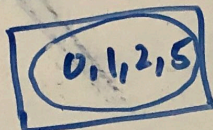
MST: $e(0,1); e(2,5)$

→ try to add: $e(0,2)$

V in one component, W in another so → merge them

(O.K. to add to MST.)

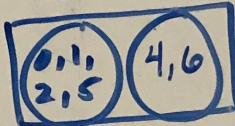
MST: $e(0,1); e(2,5); e(0,2)$



→ try to add: $e(4,6)$

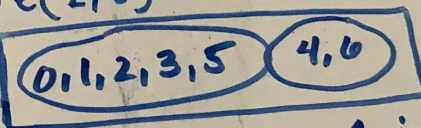
Both NOT IN = New component (O.K.)

MST: $e(0,1); e(2,5); e(0,2); e(4,6)$

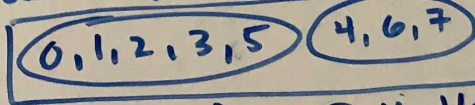


→ try to add: $e(2,3)$ → 1 in 1 component → O.K. to add to MST.

MST: $e(0,1); e(2,5); e(0,2); e(4,6); e(2,3)$



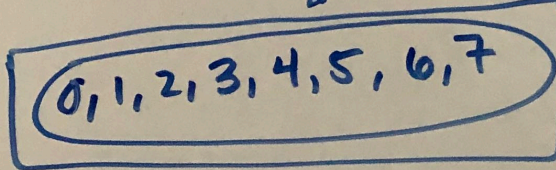
→ try to add: $e(6,7)$ → only 1 in 1 component → O.K. to add to MST.



MST: $e(0,1); e(2,5); e(0,2); e(4,6); e(2,3); e(6,7)$

→ try to add: $e(3,5)$ → Both Vertices in one component so can't add → MST stays the same

→ try to add: $e(5,6)$ → In different components so merge → O.K. to add to MST.



MST: $e(0,1); e(2,5); e(0,2); e(4,6); e(2,3); e(6,7); e(5,6)$

all vertices are in one component and

→ MST length = # of vertices - 1 so we are done.