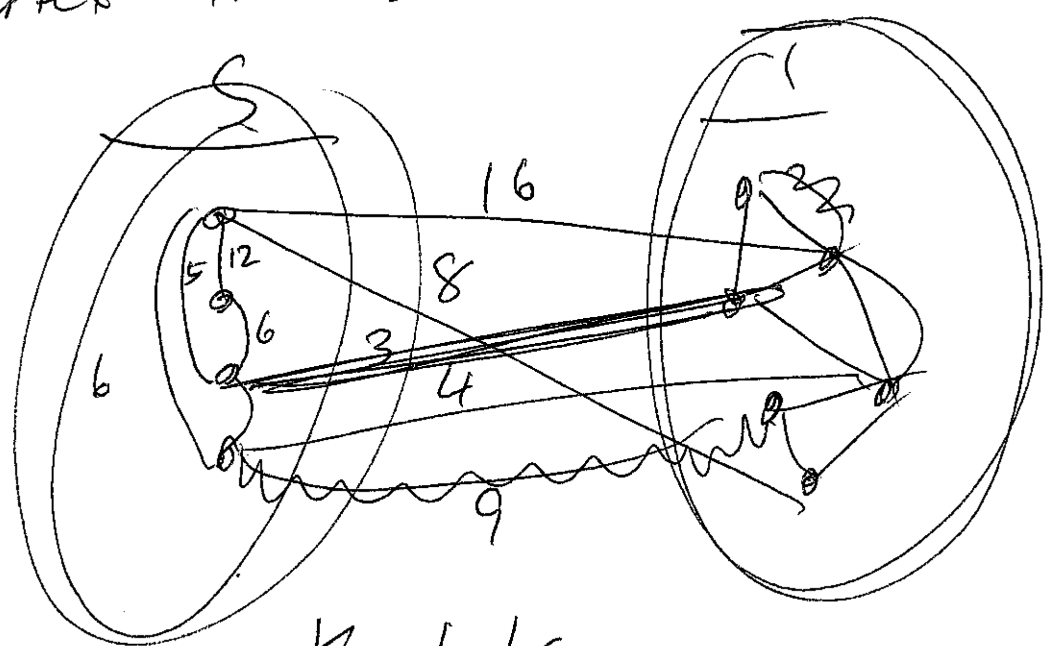
Informal Pf of Correctness: Kruskals

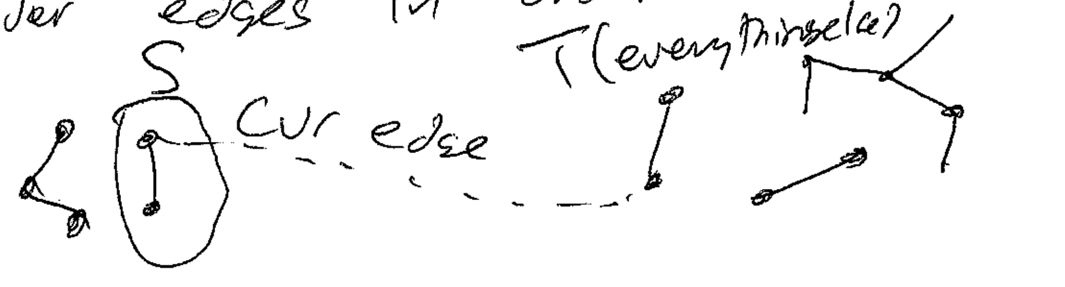
Key fact about MSTs:

If you split the vertices (V) in the graph into any 2 sets S, T where $S \cap T = \emptyset \wedge S \cup T = V$, then there exist a minimum spanning tree containing the minimum edge weight connecting a vertex from S to a vertex in T.



Kruskals

Consider edges in order of weight

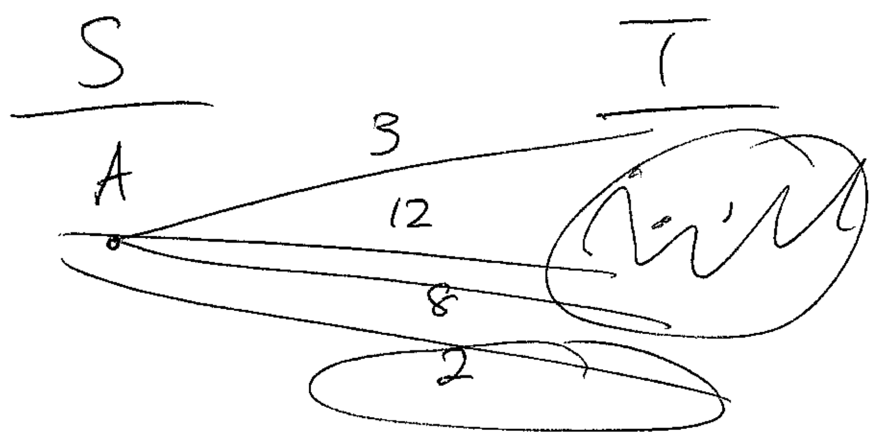


1/25/18 ②

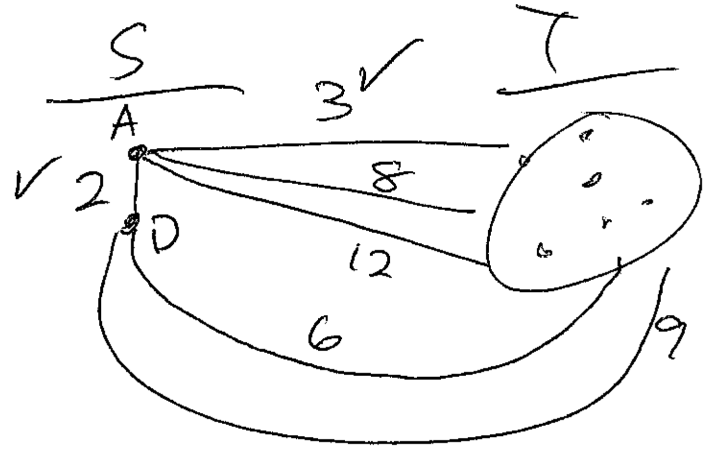
Kruskal's - we have many separate trees that eventually get connected into one. (each time my set S may be different)

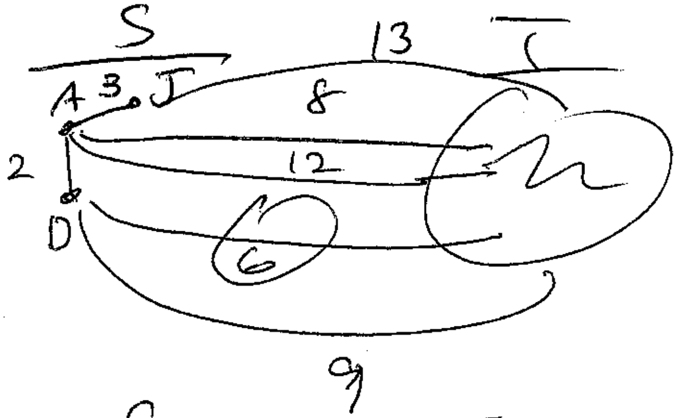
Alternate Strategy - arbitrarily put 1 vertex in S, then "grow" S by adding in the best "cross edge" over and over again.

Step 1

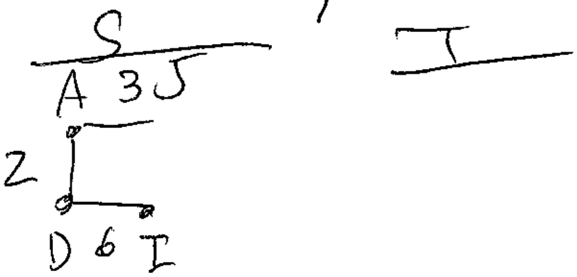


Step 2



Step 3 

1/25/18 (3)

Step 4 

This is Prim's!

(1)

$S = \{V\}$

Priority Queue (edge) = { each edge connected to V }

(2) while (S != V) {

(a) edge e = pq.poll(); // Get the next smallest edge in P.Q.

(b) if e connects two vertices in S continue; // Skip these edges.

(c) S = S ∪ {~~e~~} // Add the edge to mst → V1 or V2 new vertex

used[V1] = true

used[V2] = true

(d) Add all edges incident to V1 or V2 (new edge) to the priority queue.

1/25/18 4

13

15 ~~16~~

-3

array $\boxed{3}$, 6, −3, 4, −2, 1, −7, 3, −8, 4, 2, 8, 1, −2, 6

sum 0, 3, 9, 6, 10, 8, 9, 2, 5, $\boxed{-3}$, 1, 3, 11, $\boxed{12}$

4, 6, 14, 15

+ − +

best answer either starts at 3 OR here (OR later)