

Name : Lucky Laurens

Discord : Luckymai

Group : OUTPLAY

Obscure Binary Search Trees:

Knight's Travails:

Text Editor:

Tree Simulations:

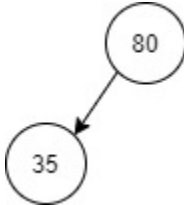
AVL

a. Insert 80:

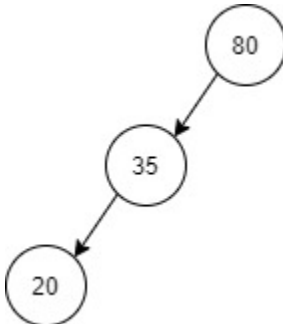


Insert 35:

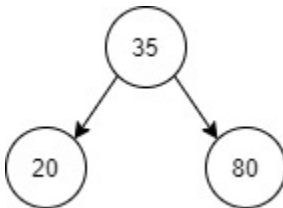
35 is less than 80, so go left.



Insert 20:

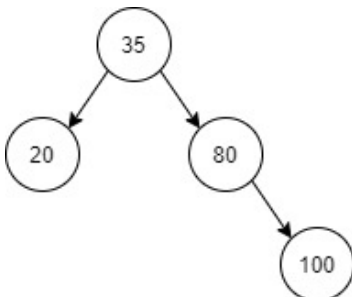


20 is less than 35, go left.



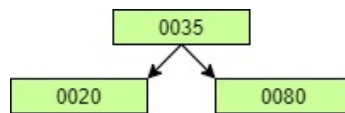
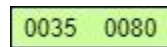
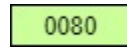
Do right rotation

Insert 100:

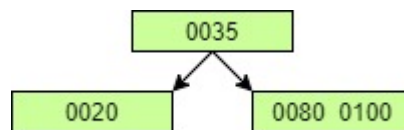


100 is more than 80, go right.

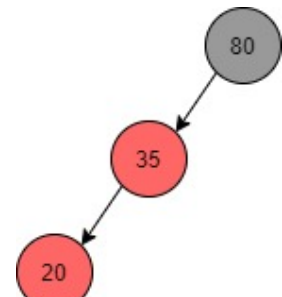
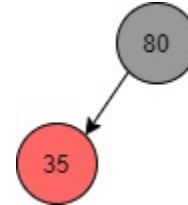
2-3 Tree



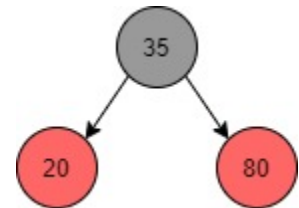
Root changed into only one key and has 2 children



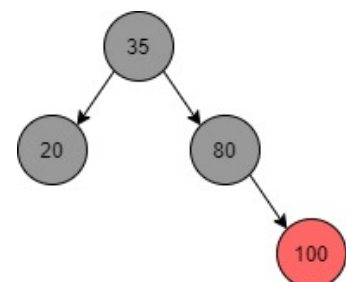
RB Tree



go left.

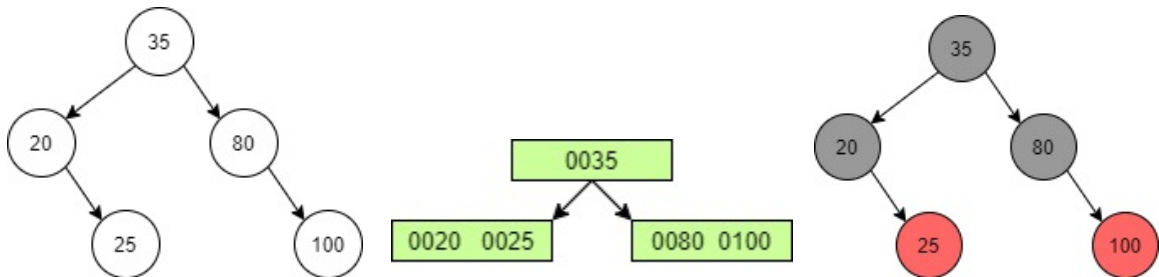


recolor and rotation



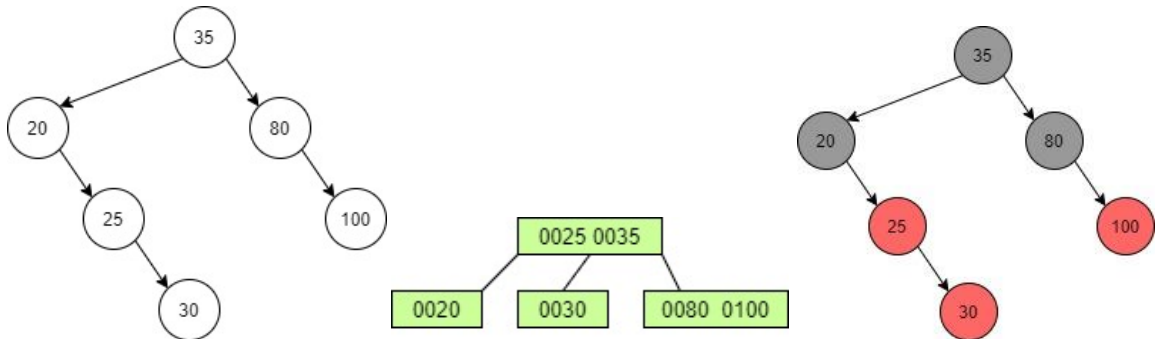
recolor the parent&uncle.

Insert 25 :

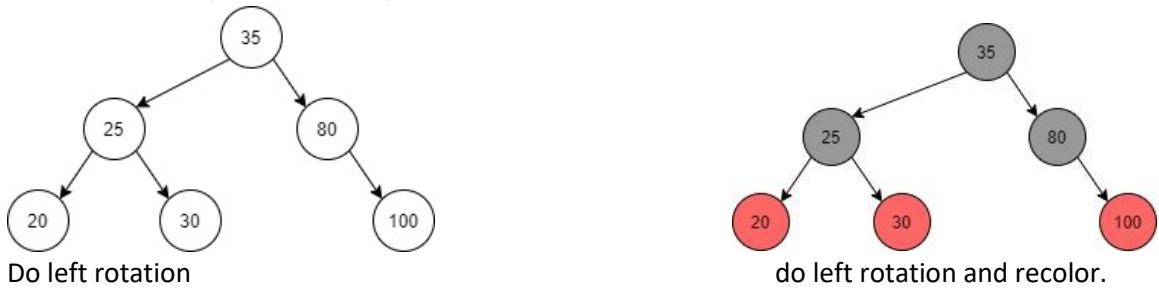


25 is less than 35, go left
25 is more than 20, go right.

Insert 30:



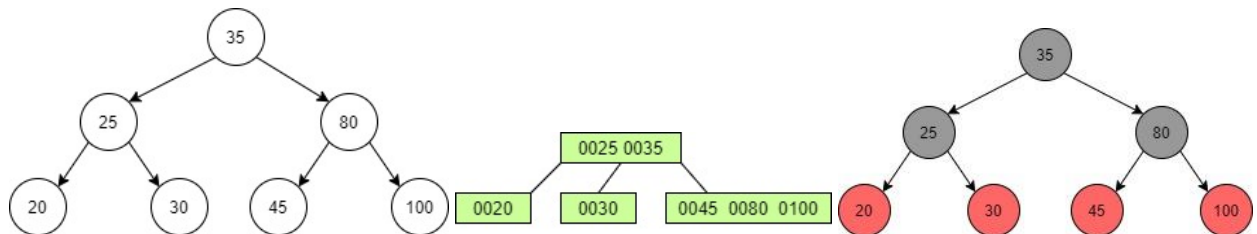
30 is less than 35, more than 20, more than 25.



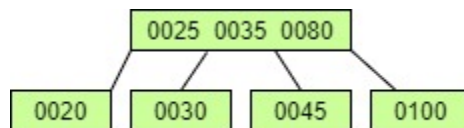
Do left rotation

do left rotation and recolor.

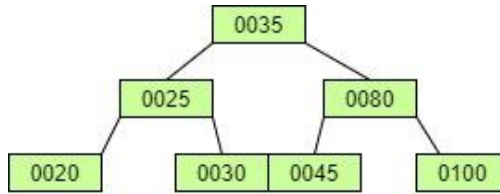
Insert 45:



45 is more than 35 and less than 80.

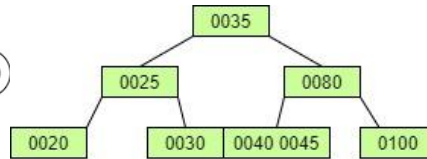
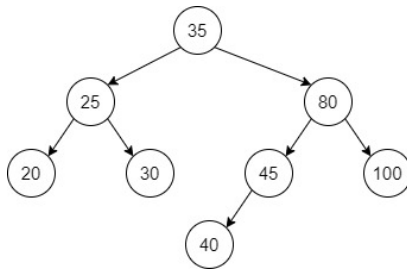


3 children is not allowed, so we pushing up a key

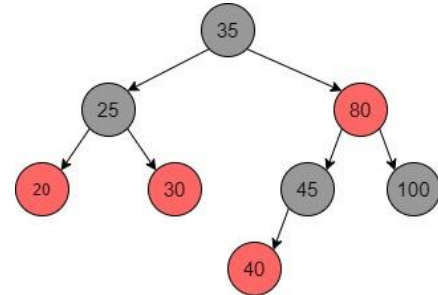
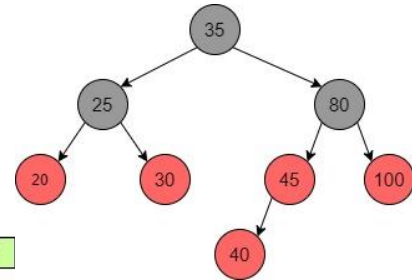


3 children is not allowed, so we pushing up a key again.

Insert 40:

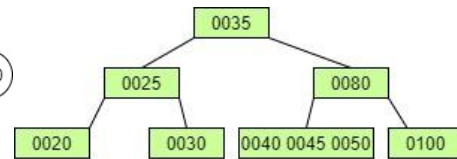
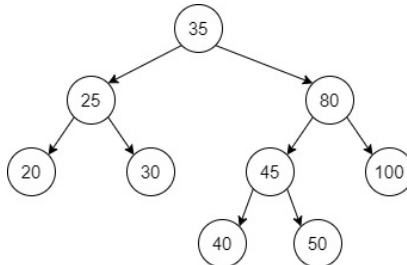


40 is more than 35, less than 80, less than 45.

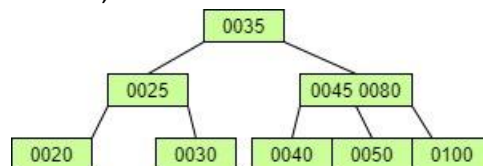


Recoloring

Insert 50:

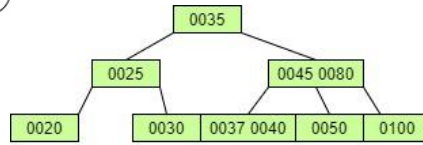
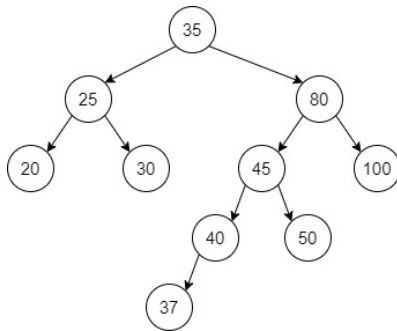


50 is more than 35, less than 80, and more than 45.

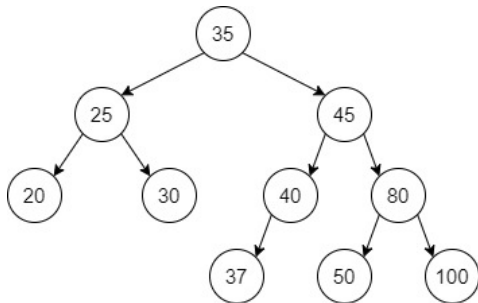


3 children is not allowed, so we pushing up a key into parent.

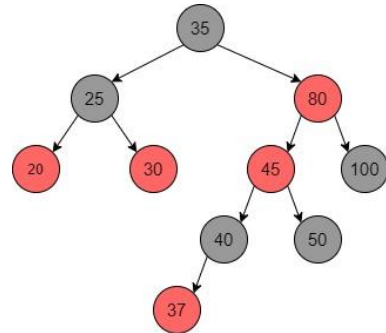
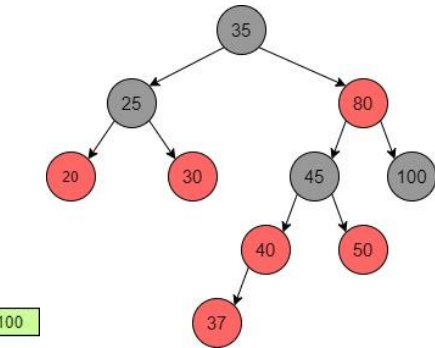
Insert 37:



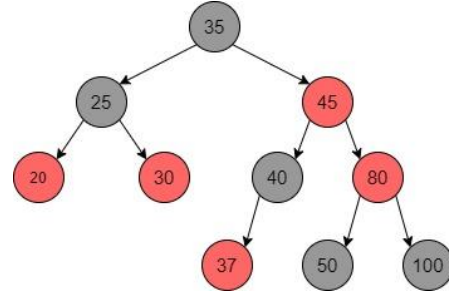
37 is more than 35, less than 80, less than 45, less than 40.



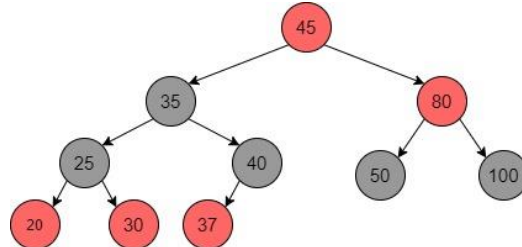
Do single right rotation on 40 45 and 80.



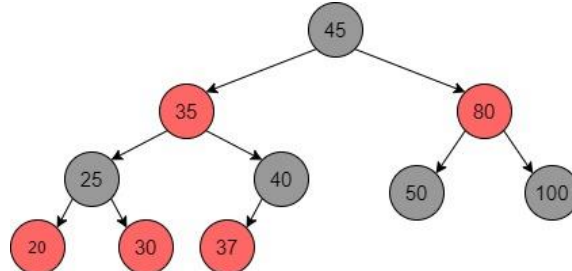
Recolor the parent, uncle, and grandparent



because there are 2 reds in a row, do right rotation on 45



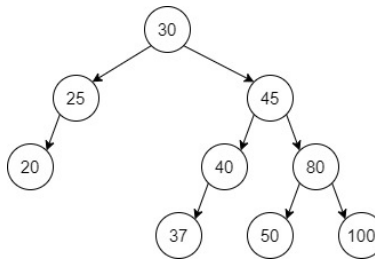
there are still 2 reds in a row, do right rotation on 45



Recoloring

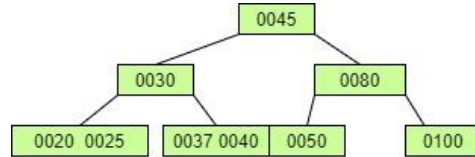
AVL

Remove 35



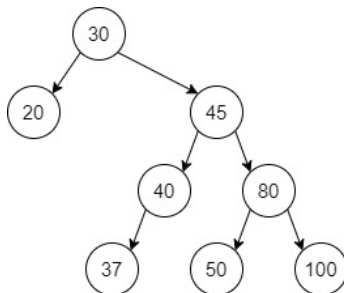
2-3 Tree

Looking for the predecessor to replace 35.

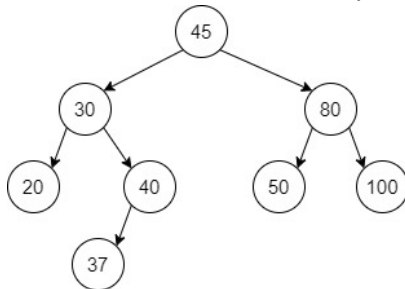


Do rotation.

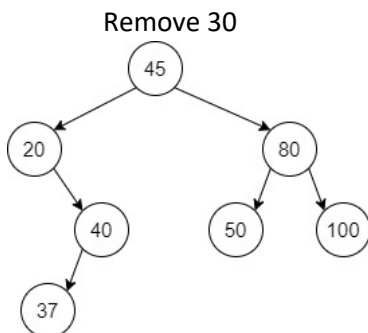
Remove 25



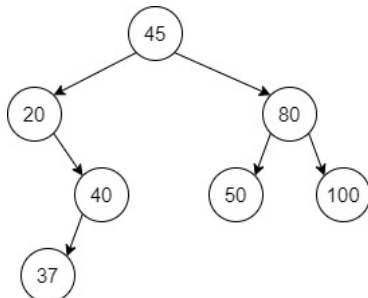
Search 25 and directly delete it.



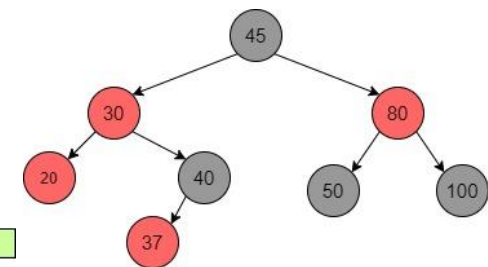
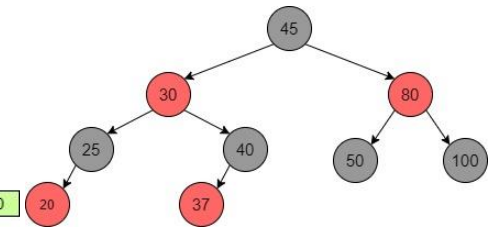
Do rotation on 45, because the different level is more than 1



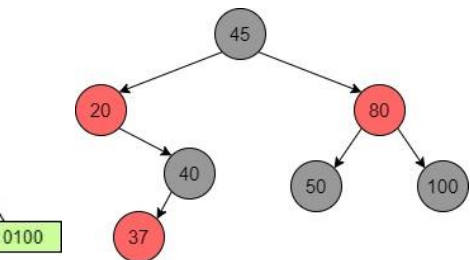
Remove 30



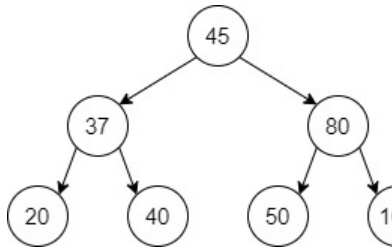
RB Tree



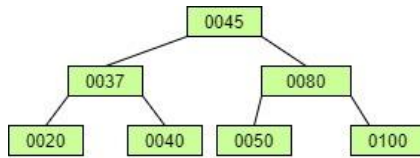
recolor node 20 into black.



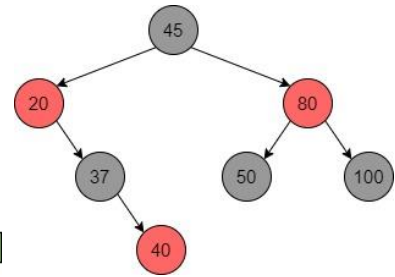
Replace 30 with it's predecessor.



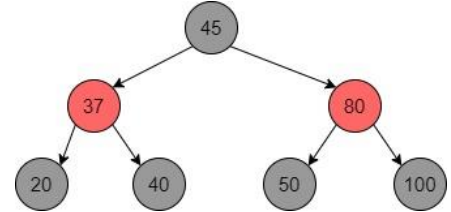
Do a right- left rotation.



Do rotation on 20,37 and 40.

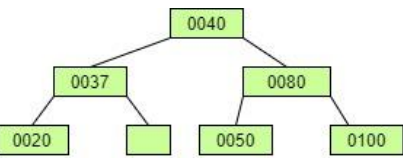
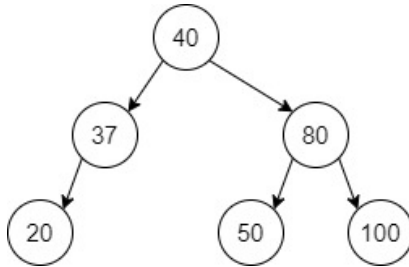


Do right rotation.

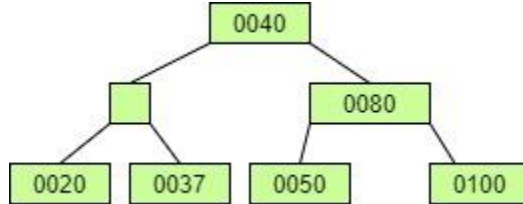


Do left rotation and recolor.

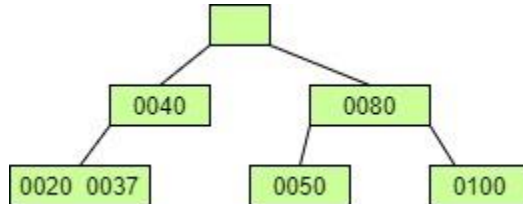
Remove 45



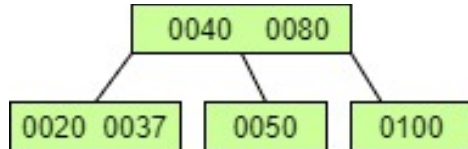
Search for the predecessor.



Move 37.

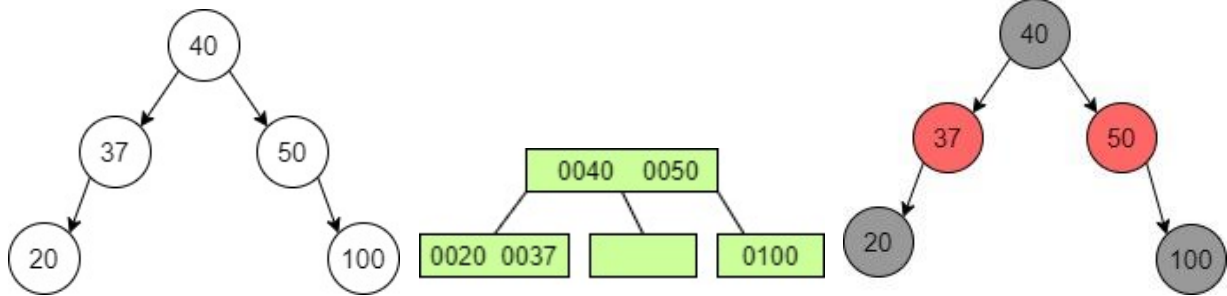


Move 40, and merge 20 and 37.

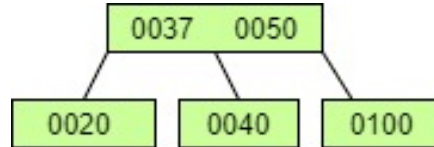


Merge 40 and 80.

Remove 80



Find the predecessor of 80

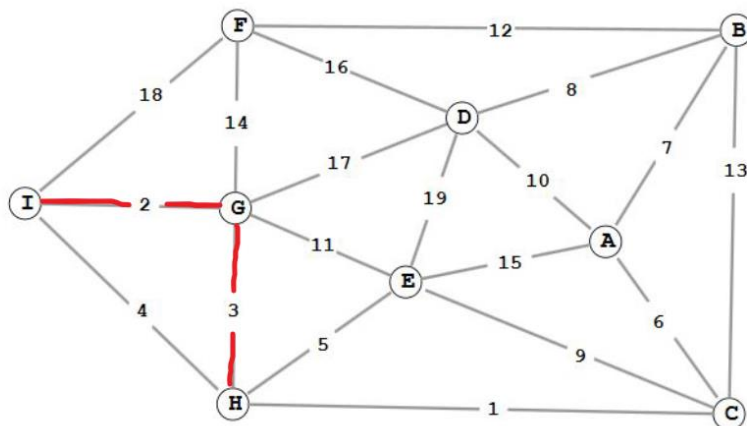
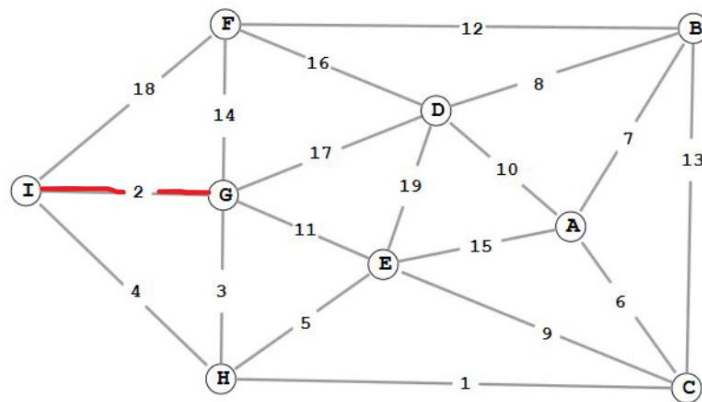


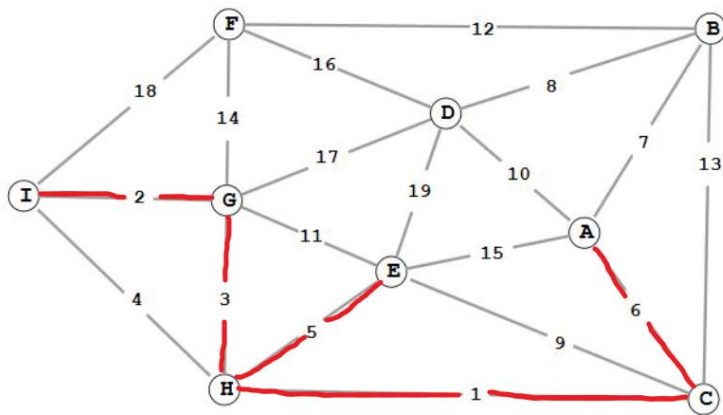
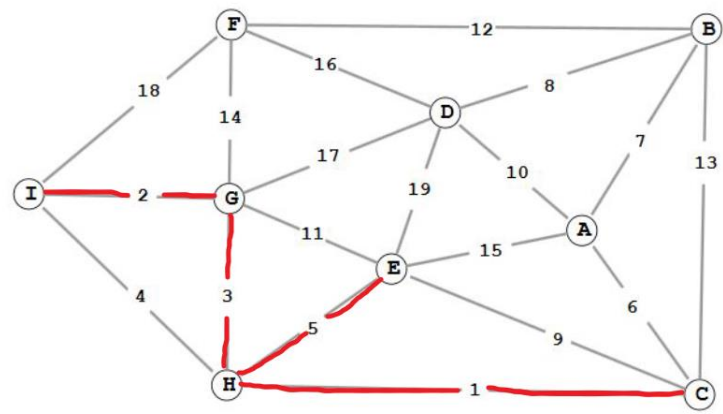
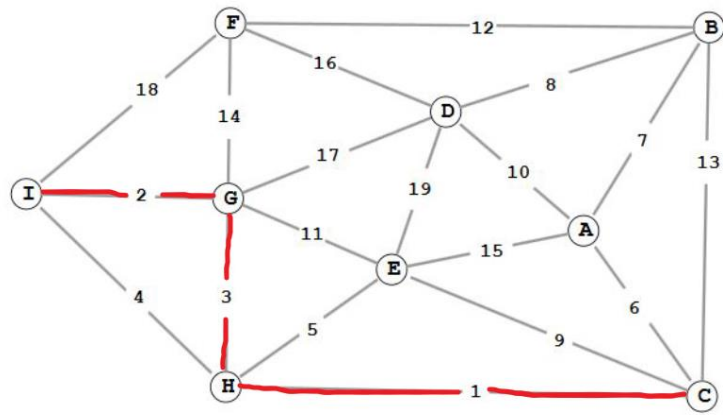
Move 40 into child, and up 37 into parent.

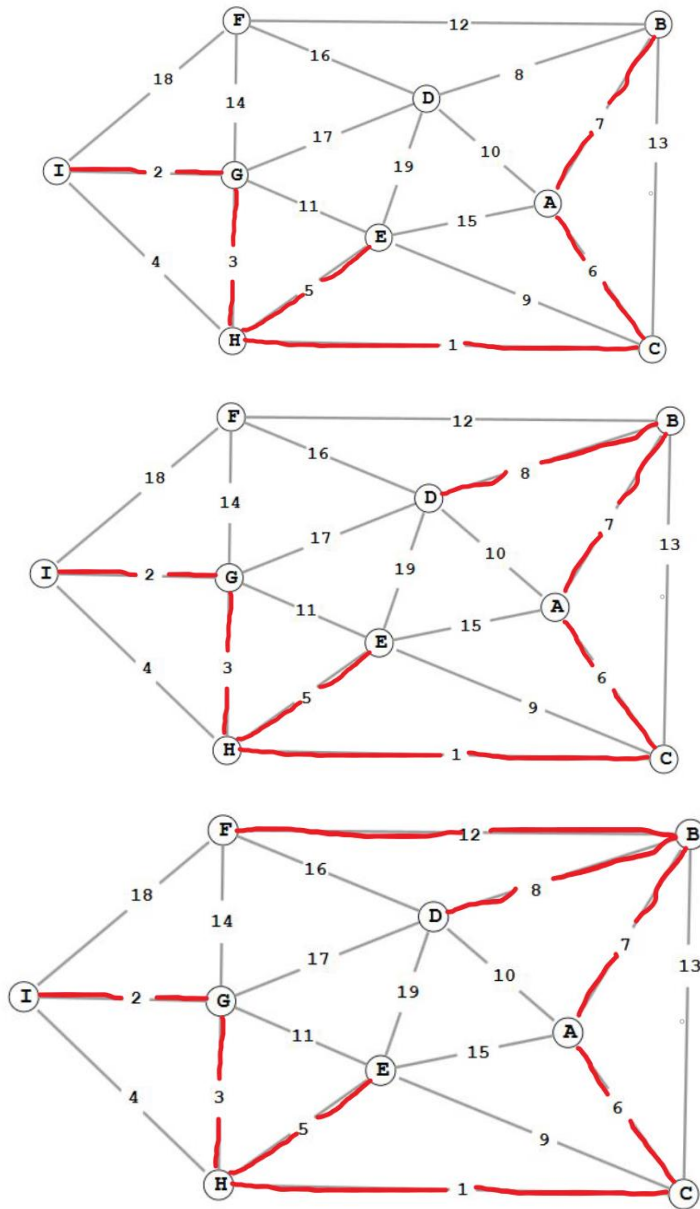
b. Insert

Disjoint Sets and Graphs:

- Prim's Algorithm

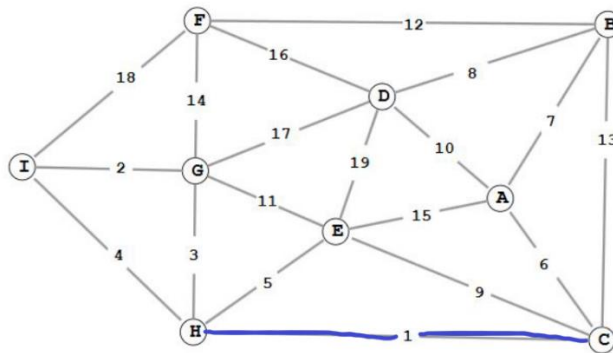


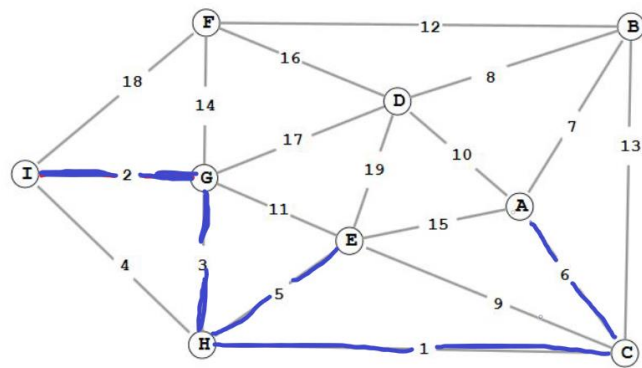
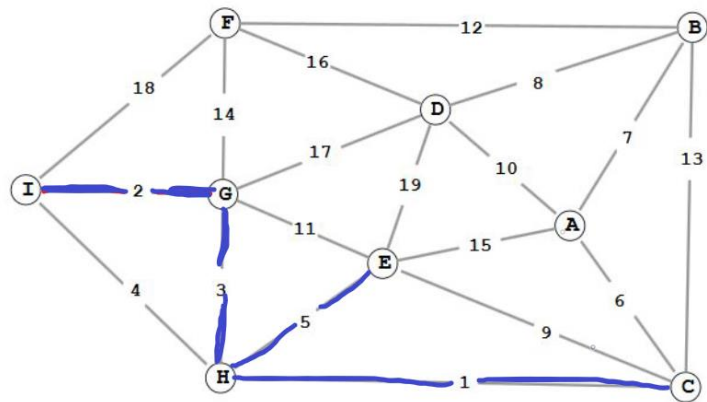
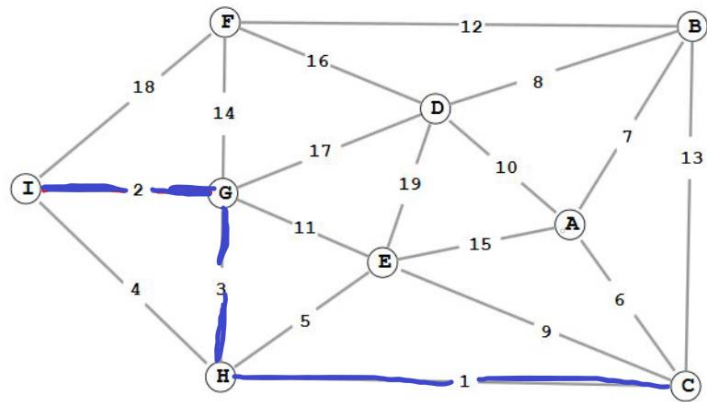
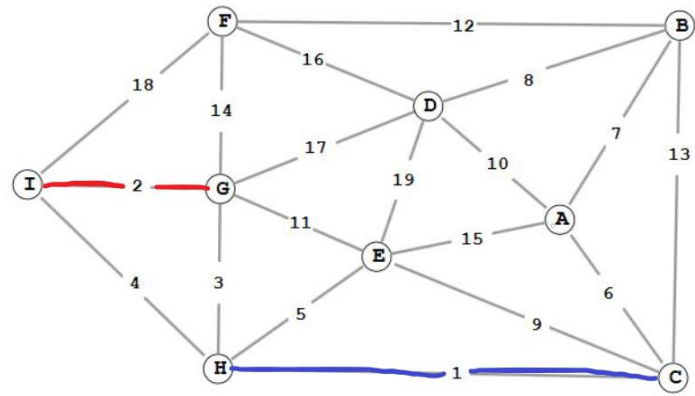


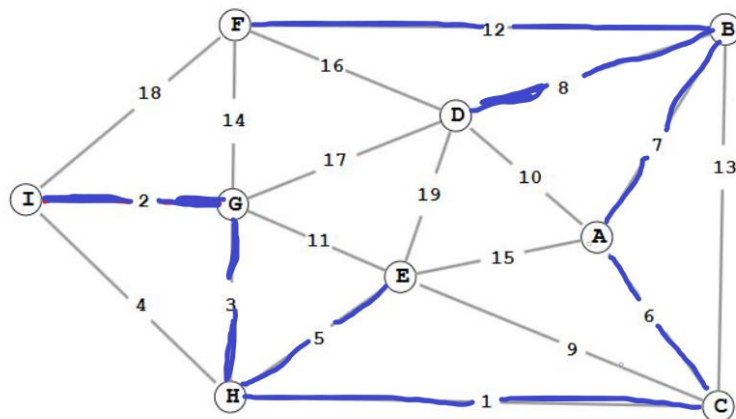
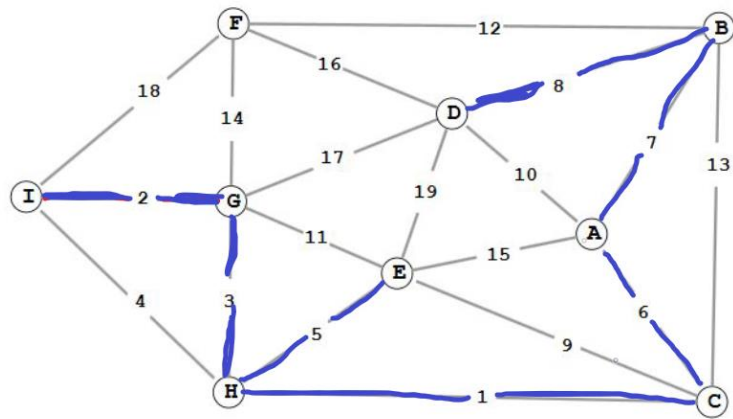
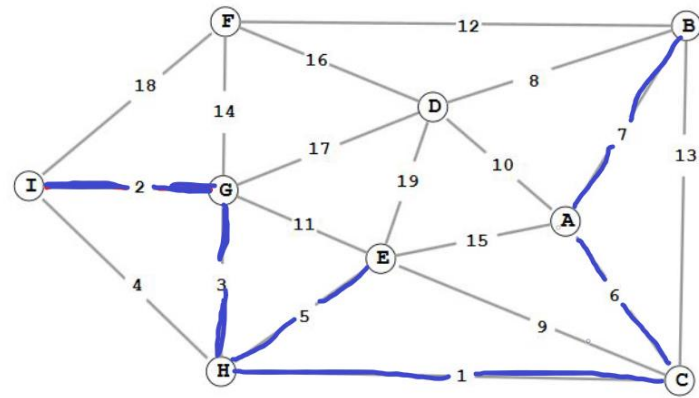


Minimum spanning tree = 2 + 3 + 5 + 1 + 6 + 7 + 8 + 12 = 44

- Kruskal Algorithm

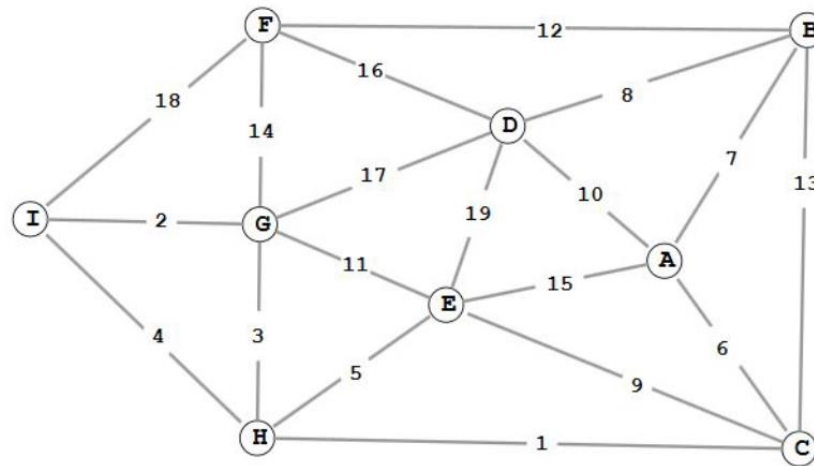






Minimum spanning tree = $2 + 3 + 5 + 1 + 6 + 7 + 8 + 12 = 44$

- Shortest Path from I to A



	A	B	C	D	E	F	G	H	I
I	INF	INF	INF	INF	INF	18I	2I	4I	0I
G	INF	INF	INF	19G	13G	16G	2I	4I	X
H	INF	INF	5H	19G	9H	16G	X	4I	X
C	11C	18C	5H	19G	9H	16G	X	X	X
E	11C	18C	X	19G	9H	16G	X	X	X
A	11C	18C	X	19G	X	16G	X	X	X

I -> G = 2

I -> H = 4

I -> H -> C = 5

I -> H -> E = 9

I -> H -> C -> A = 11

So, the sortest path from I to A is 11.

- Shortest Path from F to C

	A	B	C	D	E	F	G	H	I
F	INF	12F	INF	16F	INF	0F	14F	INF	18F
B	19B	12F	25B	16F	INF	X	14F	INF	18F
G	19B	X	25B	16F	25G	X	14F	17G	16G
D	19B	X	25B	16F	25G	X	X	17G	16G
I	19B	X	25B	X	25G	X	X	17G	16G
H	19B	X	18H	X	22H	X	X	17G	X
C	19B	X	18H	X	22H	X	X	X	X

F->B = 12

F->G = 14

F->D = 16

F->G->I = 16

F->G->H = 17

F->G->H->C = 18

So, the shortest path from F to C is 18.