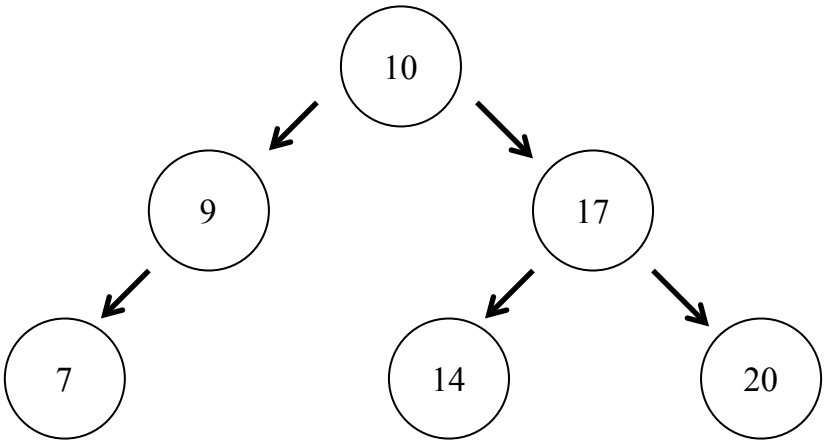
Binary Search Tree

1) In-order traversal

2) Actions:

insert(12);
insert(5);
delete(12);
delete(7);
delete(10);



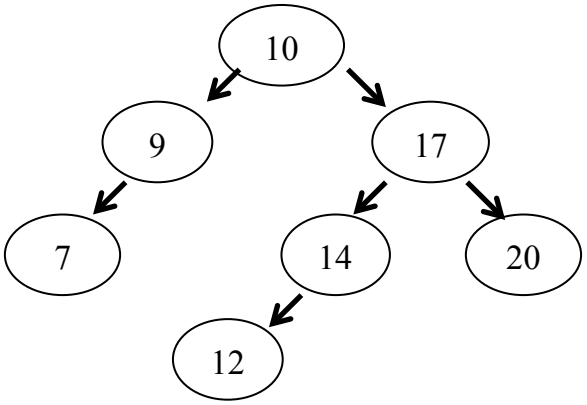
Binary Search Tree (Answer)

1) In-order traversal

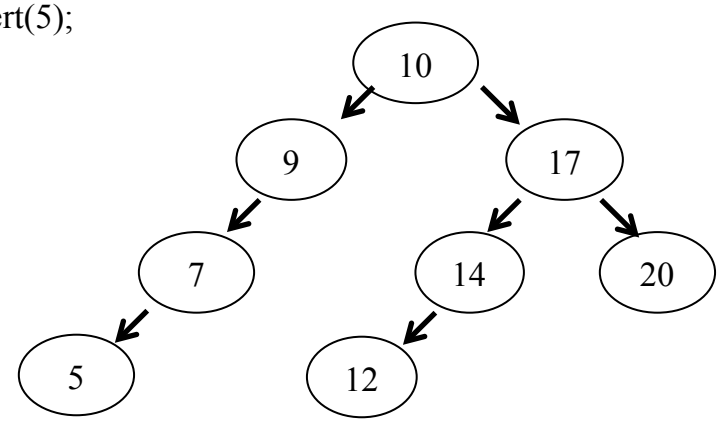
7 – 9 – 10 – 14 – 17 – 20

2) Actions:

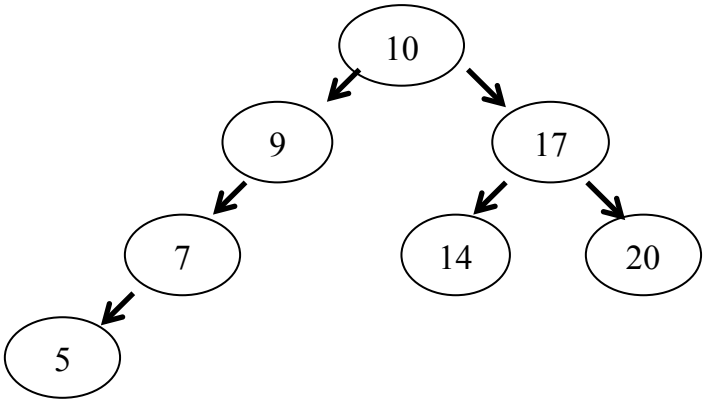
insert(12);



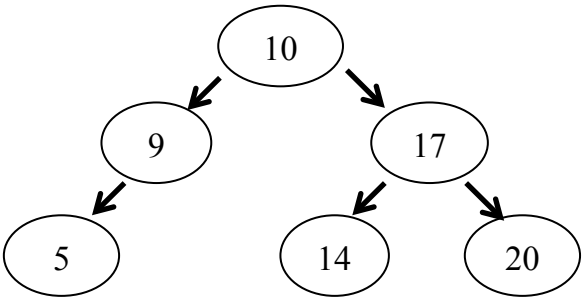
insert(5);



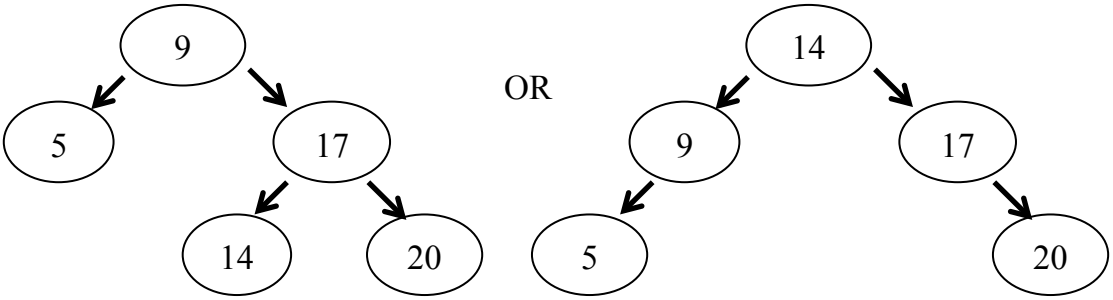
delete(12);



delete(7);



delete(10);

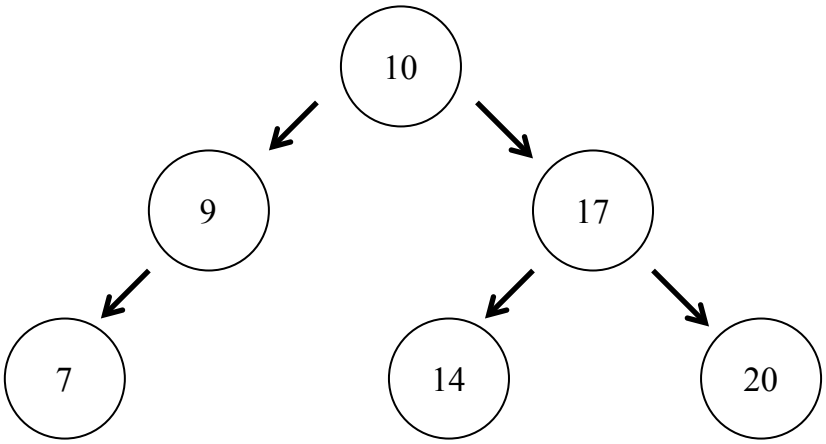


AVL Tree

1) Assign heights to each node. Is it balanced?

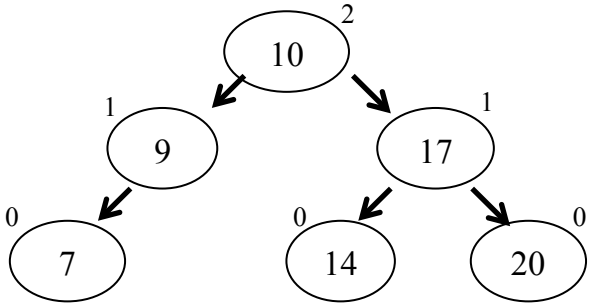
2) Actions:

insert(12);
insert(5);
insert(15);
insert(16);

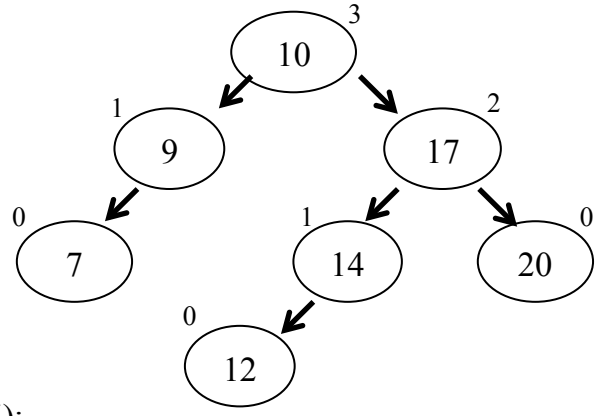


AVL Tree (Solution)

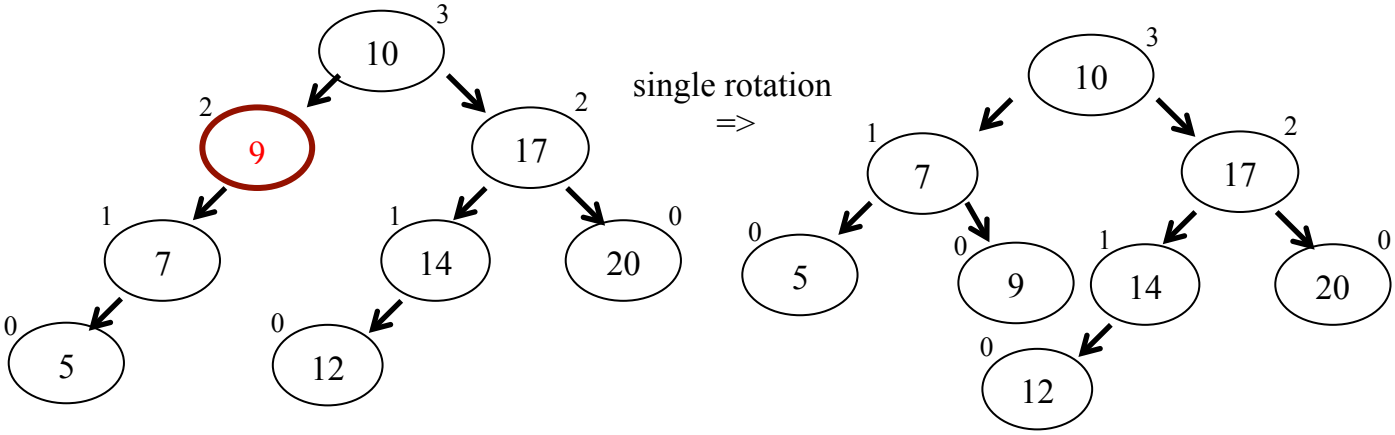
1) Assign heights to each node. Is it balanced?
Yes.



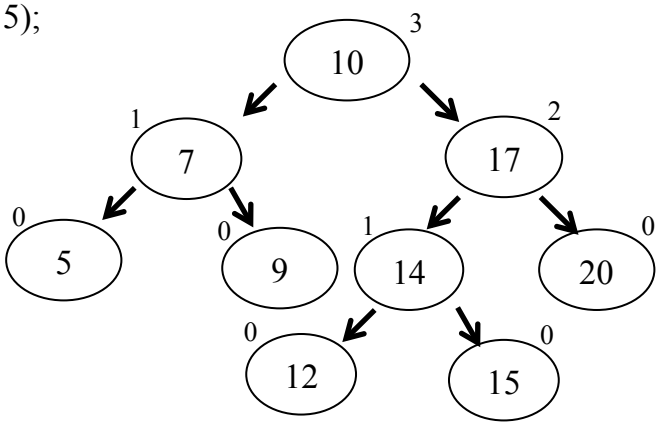
2) Actions:
insert(12);



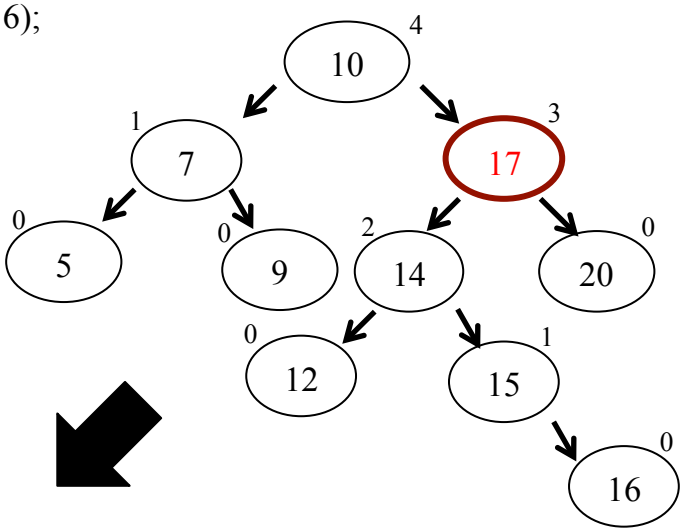
insert(5);

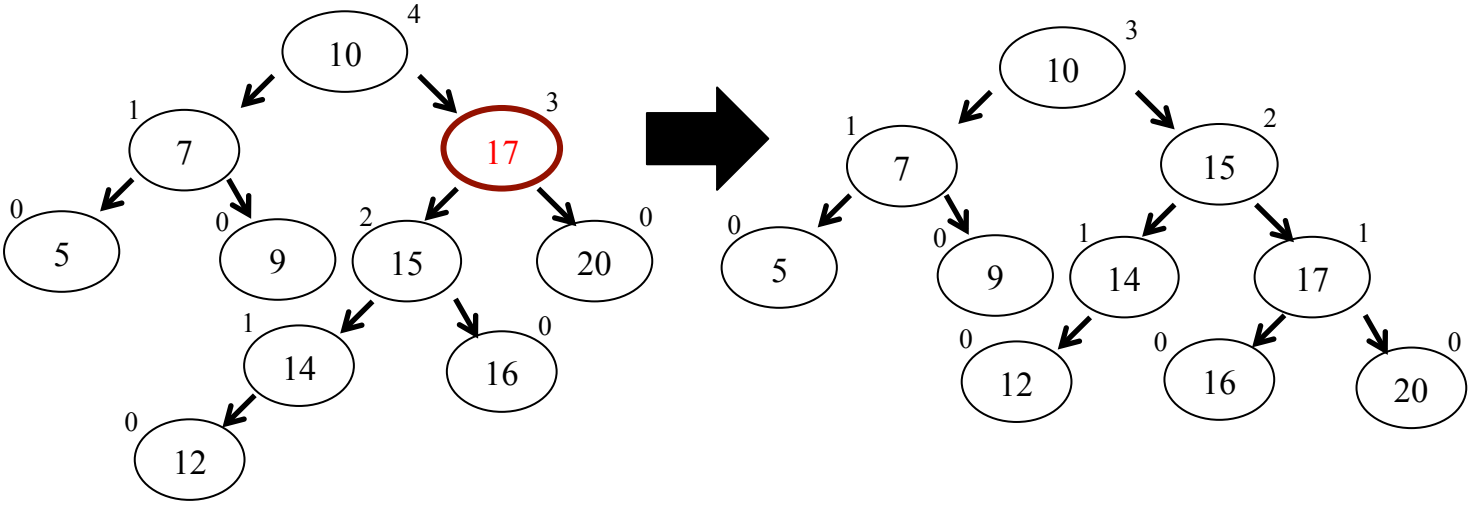


insert(15);



insert(16);



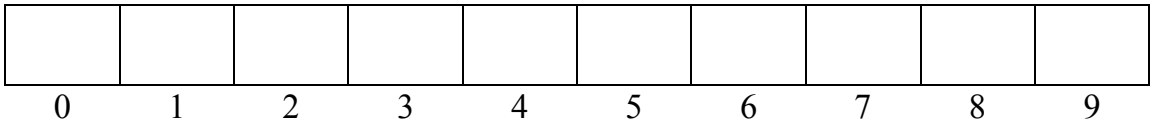


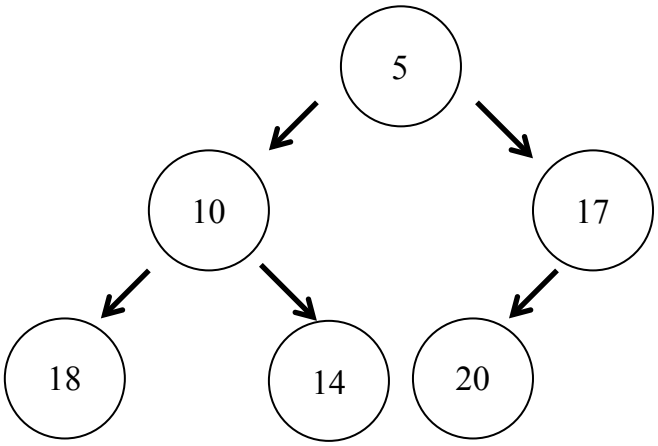
Priority Queue & Heap

1) What the array representation of the binary heap should look like?

2) Actions:

insert(28);
insert(3);
deletemin();





Priority Queue & Heap (Solution)

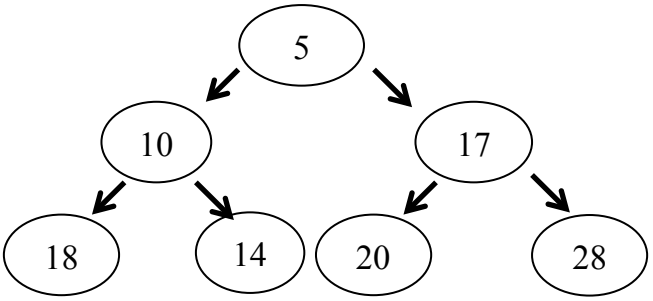
1) What the array representation of the binary heap should look like?

	5	10	17	18	14	20			
0	1	2	3	4	5	6	7	8	9

2) Actions:

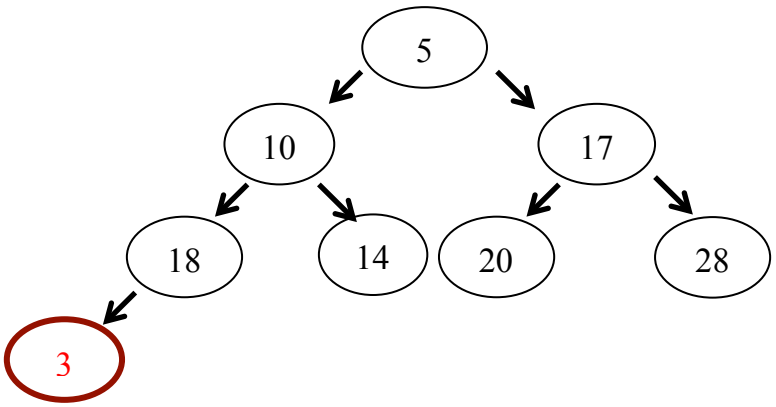
insert(28);

	5	10	17	18	14	20	28		
0	1	2	3	4	5	6	7	8	9

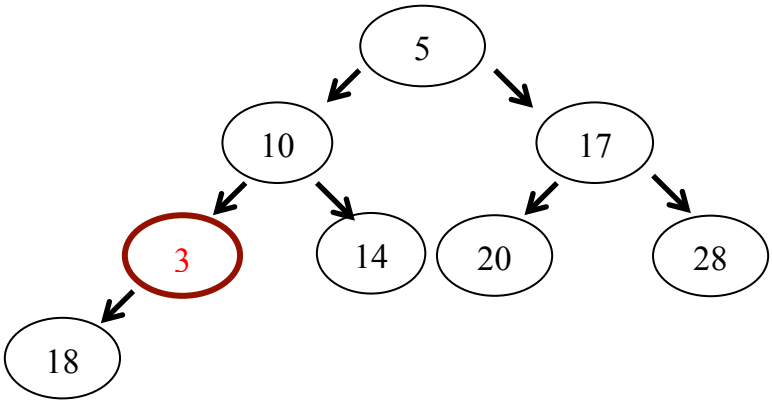


insert(3);

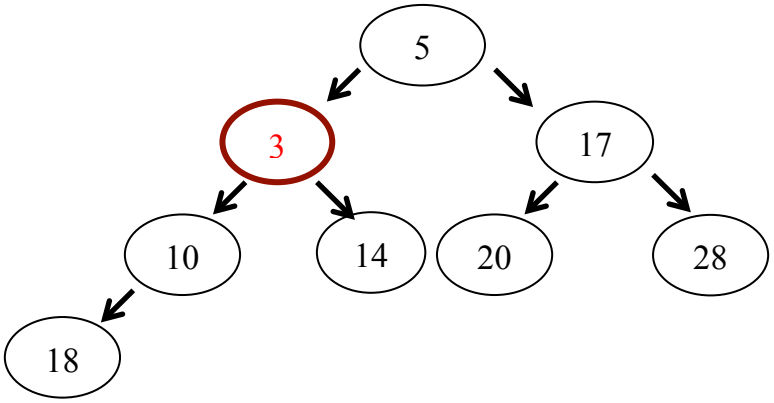
	5	10	17	18	14	20	28	3	
0	1	2	3	4	5	6	7	8	9



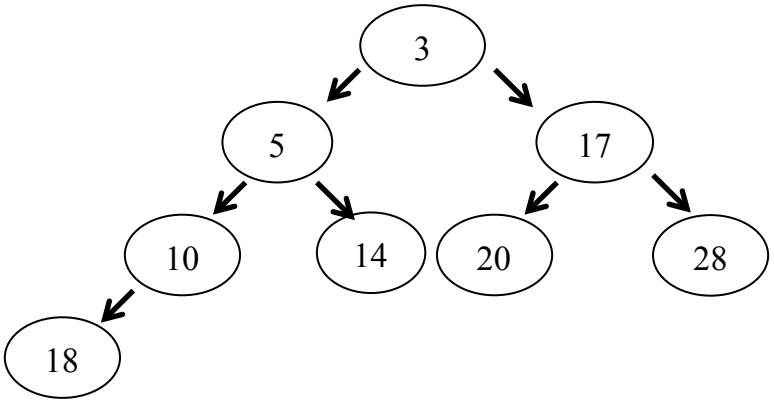
	5	10	17	3	14	20	28	18	
0	1	2	3	4	5	6	7	8	9



	5	3	17	10	14	20	28	18	
0	1	2	3	4	5	6	7	8	9

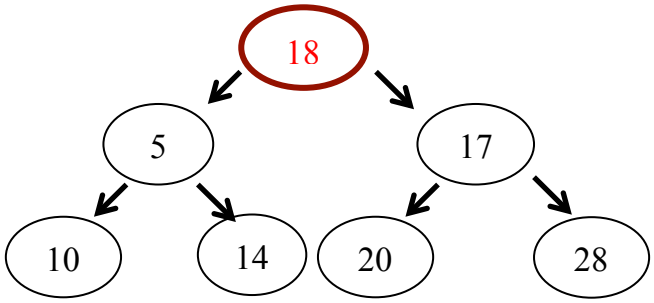


	3	5	17	10	14	20	28	18	
0	1	2	3	4	5	6	7	8	9

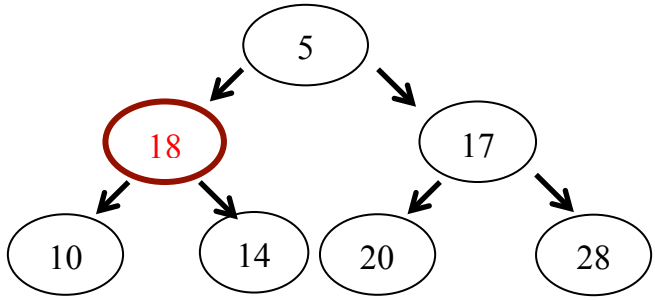


deletemin();

	18	5	17	10	14	20	28		
0	1	2	3	4	5	6	7	8	9



	5	18	17	10	14	20	28		
0	1	2	3	4	5	6	7	8	9



	5	10	17	18	14	20	28		
0	1	2	3	4	5	6	7	8	9