

insert 24: [24]

insert 60: [24 | 60]

insert 100: [24 | 60 | 100]

insert 48: [60]
[24 | 48 | 100]

insert 50: [60]
[24 | 48 | 50 | 100]

insert 17: [60]
[17 | 24 | 48 | 50 | 100] → [48 | 60]
[17 | 24 | 50 | 100]

insert 32: [48 | 60]
[17 | 24 | 32 | 50 | 100]

insert 24: [48 | 60] → [24 | 48 | 60]
[17 | 24 | 32 | 50 | 100] [17 | 24 | 32 | 50 | 100]

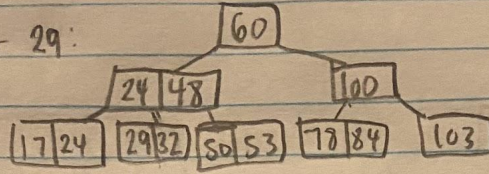
insert 53: [24 | 48 | 60]
[17 | 24 | 32 | 50 | 53 | 100]

insert 78: [24 | 48 | 60]
[17 | 24 | 32 | 50 | 53 | 78 | 100]

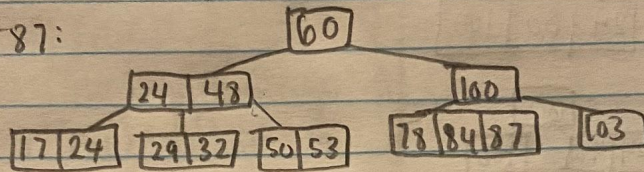
insert 103: [24 | 48 | 60]
[17 | 24 | 32 | 50 | 53 | 78 | 100 | 103]

insert 84: [24 | 48 | 60 | 100] → [60]
[17 | 24 | 32 | 50 | 53 | 78 | 84 | 103] [24 | 48]
[17 | 24 | 32 | 50 | 53 | 78 | 84 | 103]

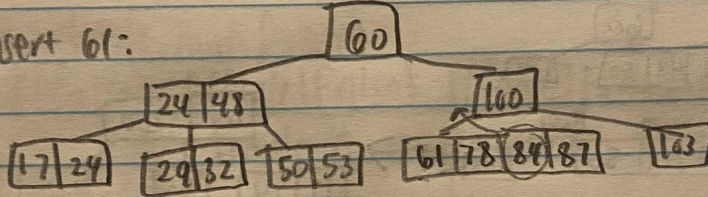
insert 29:



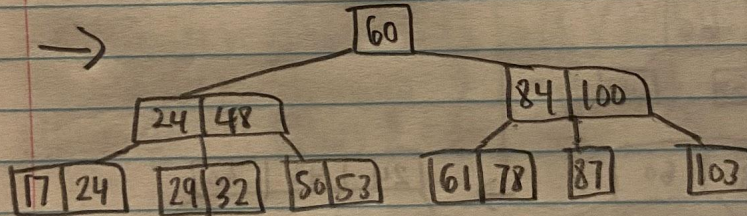
insert 87:

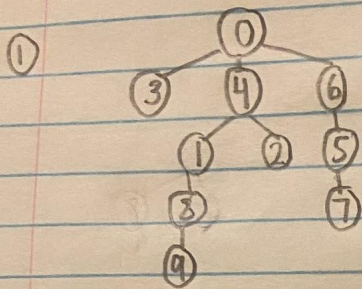


insert 61:



→

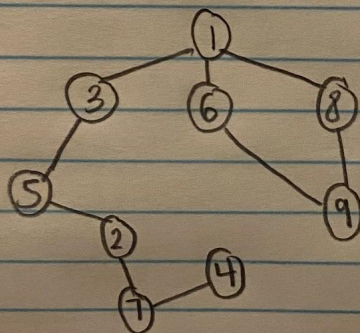




BFS uses a queue $\rightarrow [0, 3, 4, 6, 1, 2, 5, 8, 7, 9]$

DFS uses a stack $\rightarrow [0, 3, 4, 1, 8, 9, 2, 6, 5, 7]$

②



DFS visualization

```

      6
     / \
    1   9
   / \
  3   8
 / \
5   2
 \ /
  7
 / \
4   9
  
```

BFS uses a queue $\rightarrow [8, 1, 9, 3, 6, 5, 2, 7, 4]$

DFS uses a stack $\rightarrow [6, 1, 3, 5, 2, 7, 4, 8, 9]$