

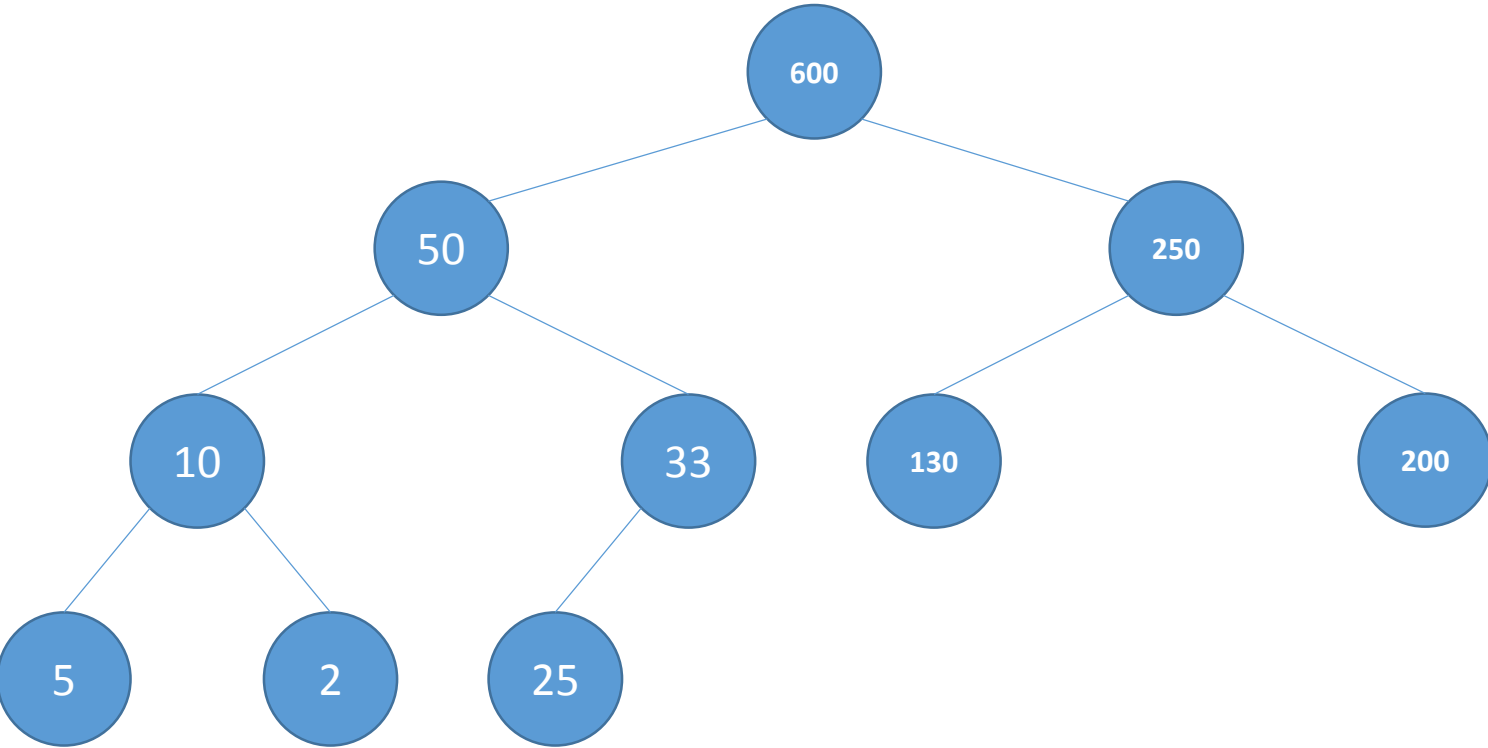
X	600	50	250	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 10



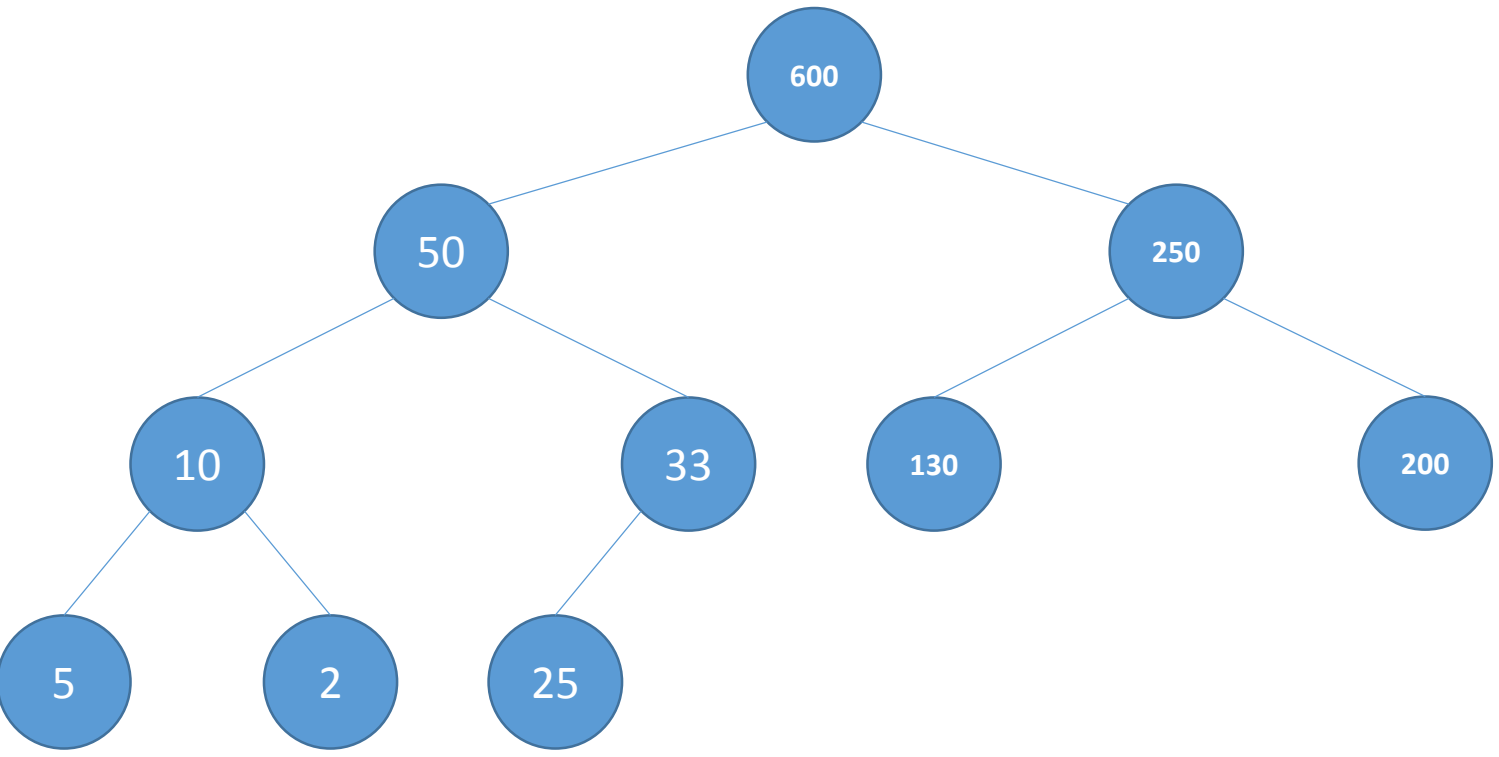
X	600	50	250	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 10
tempKey = 600



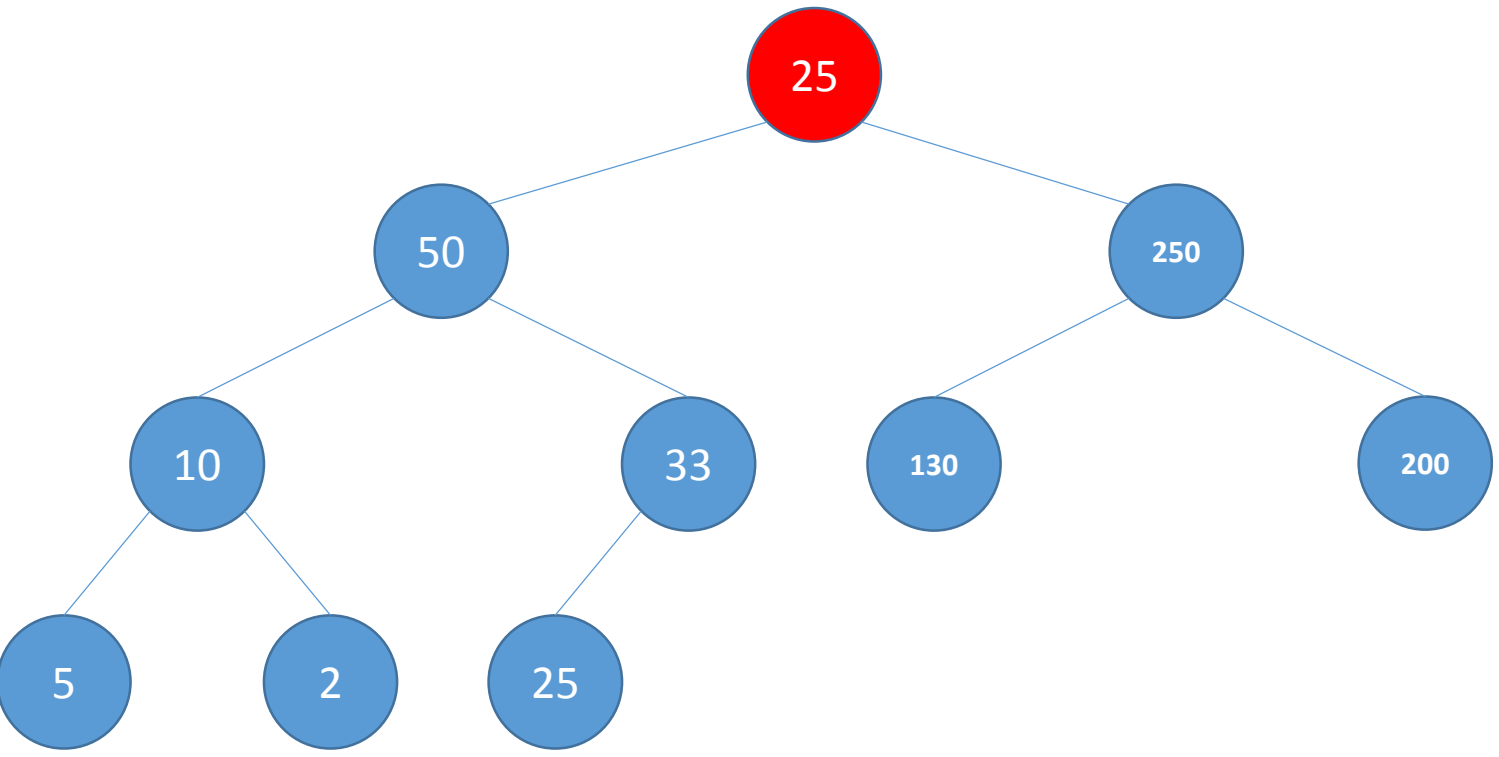
X	25	50	250	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 10
tempKey = 600



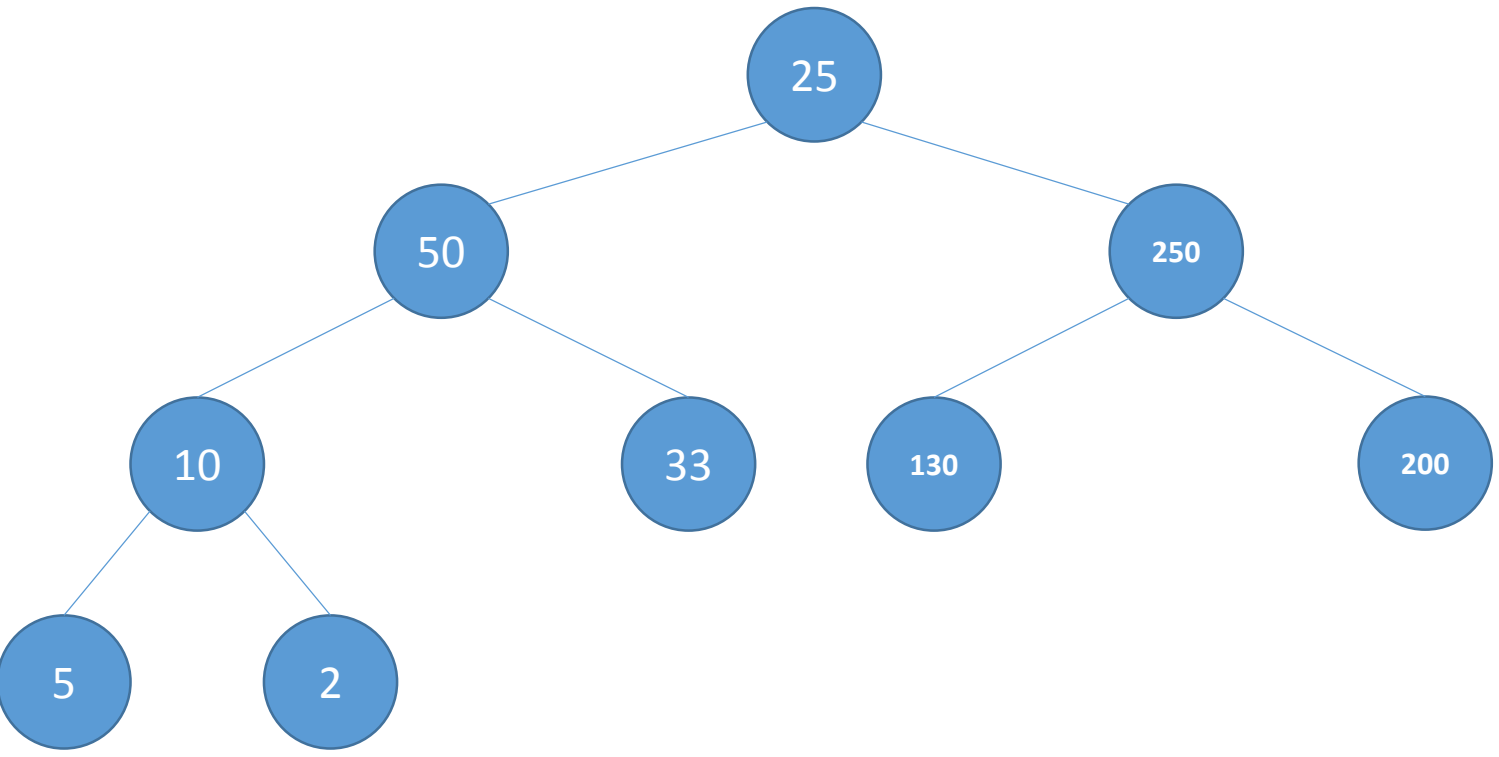
X	25	50	250	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

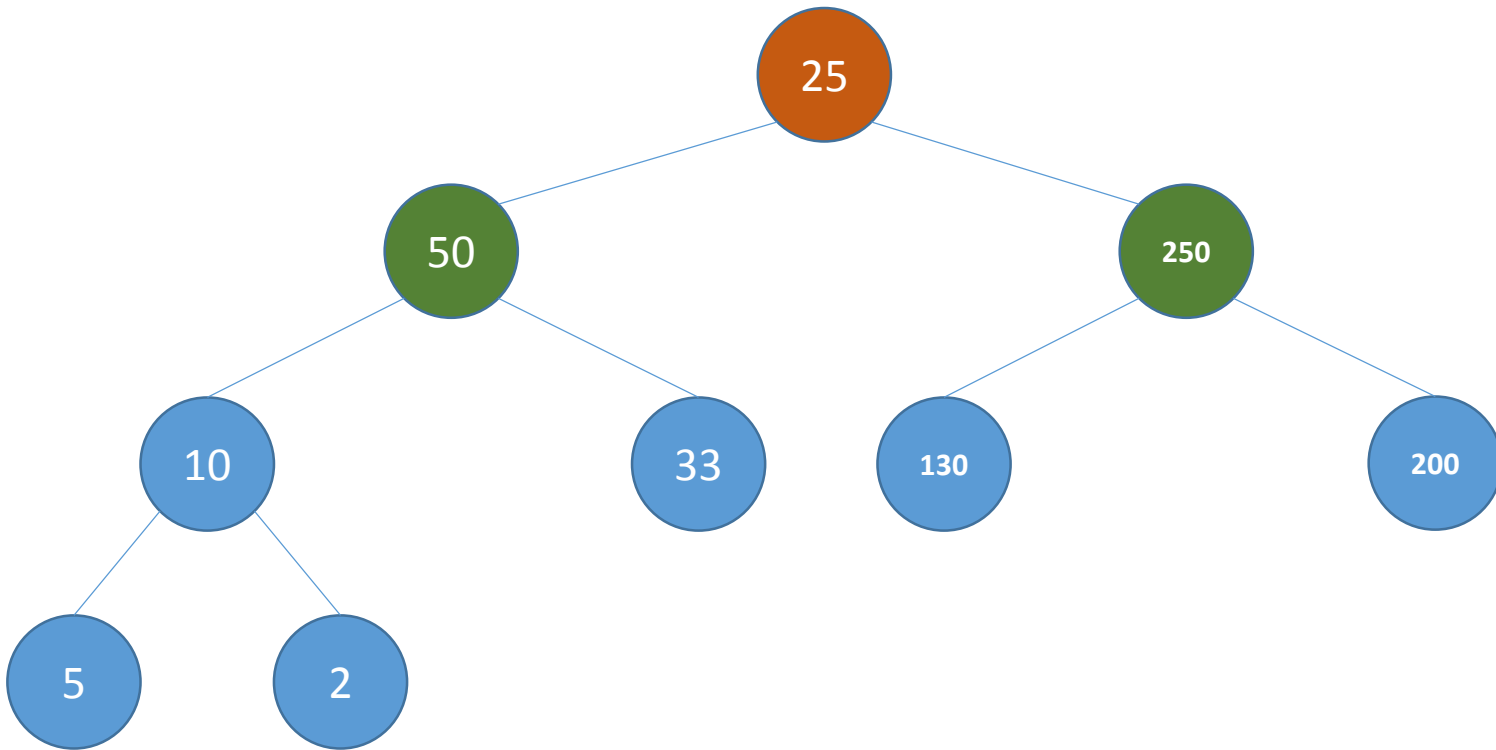
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 9
tempKey = 600



X	25	50	250	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 9
tempKey = 600

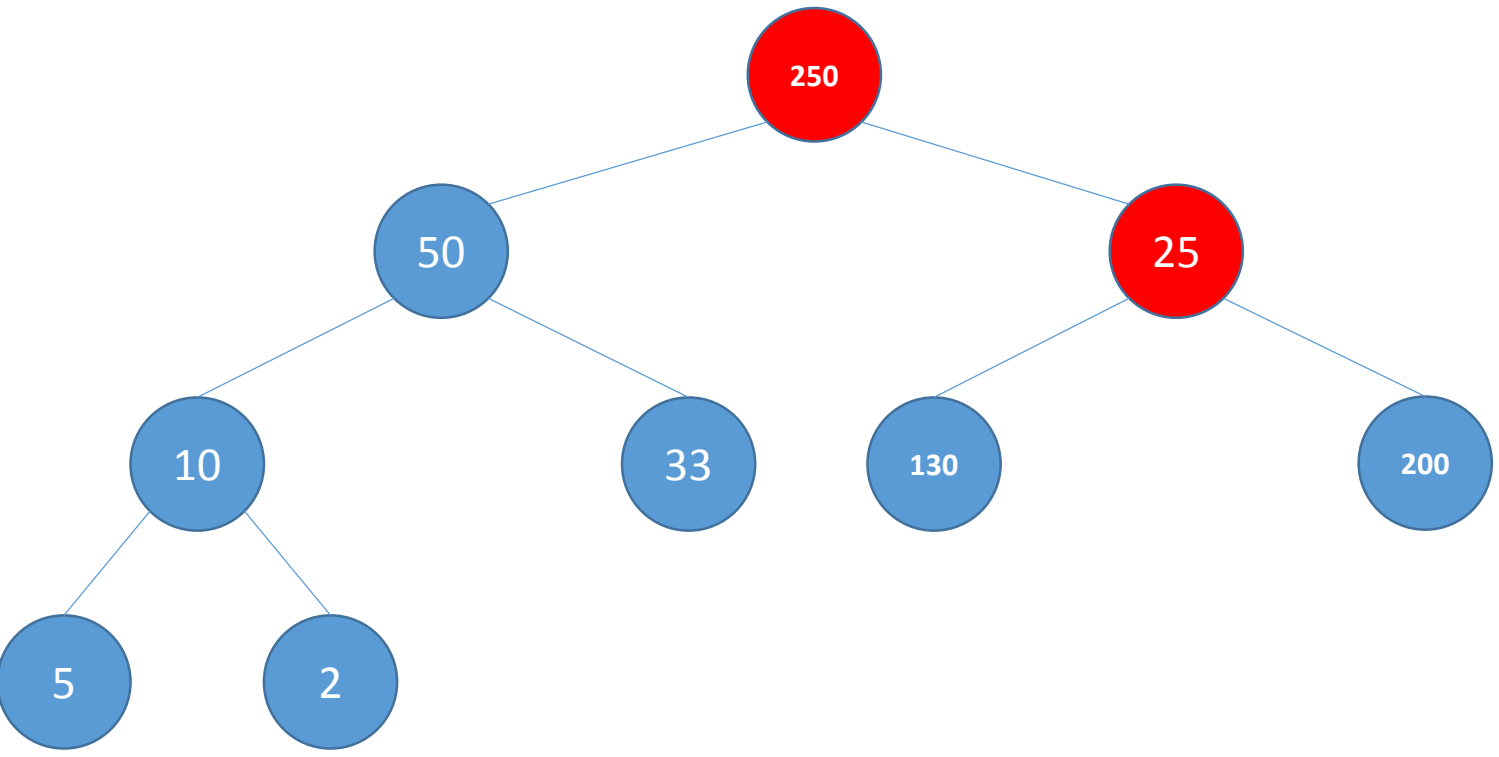
X	250	50	25	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 9
tempKey = 600



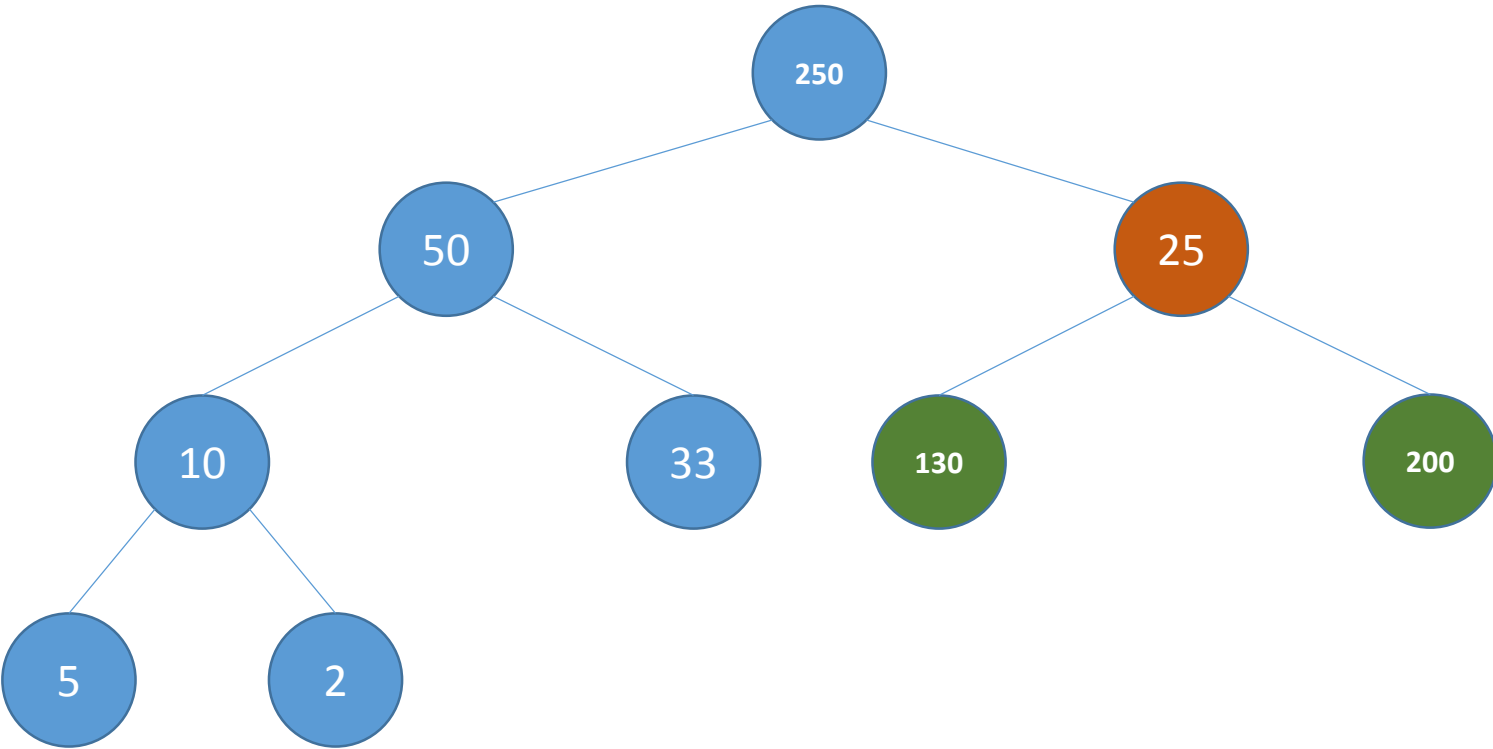
X	250	50	25	10	33	130	200	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 9
tempKey = 600



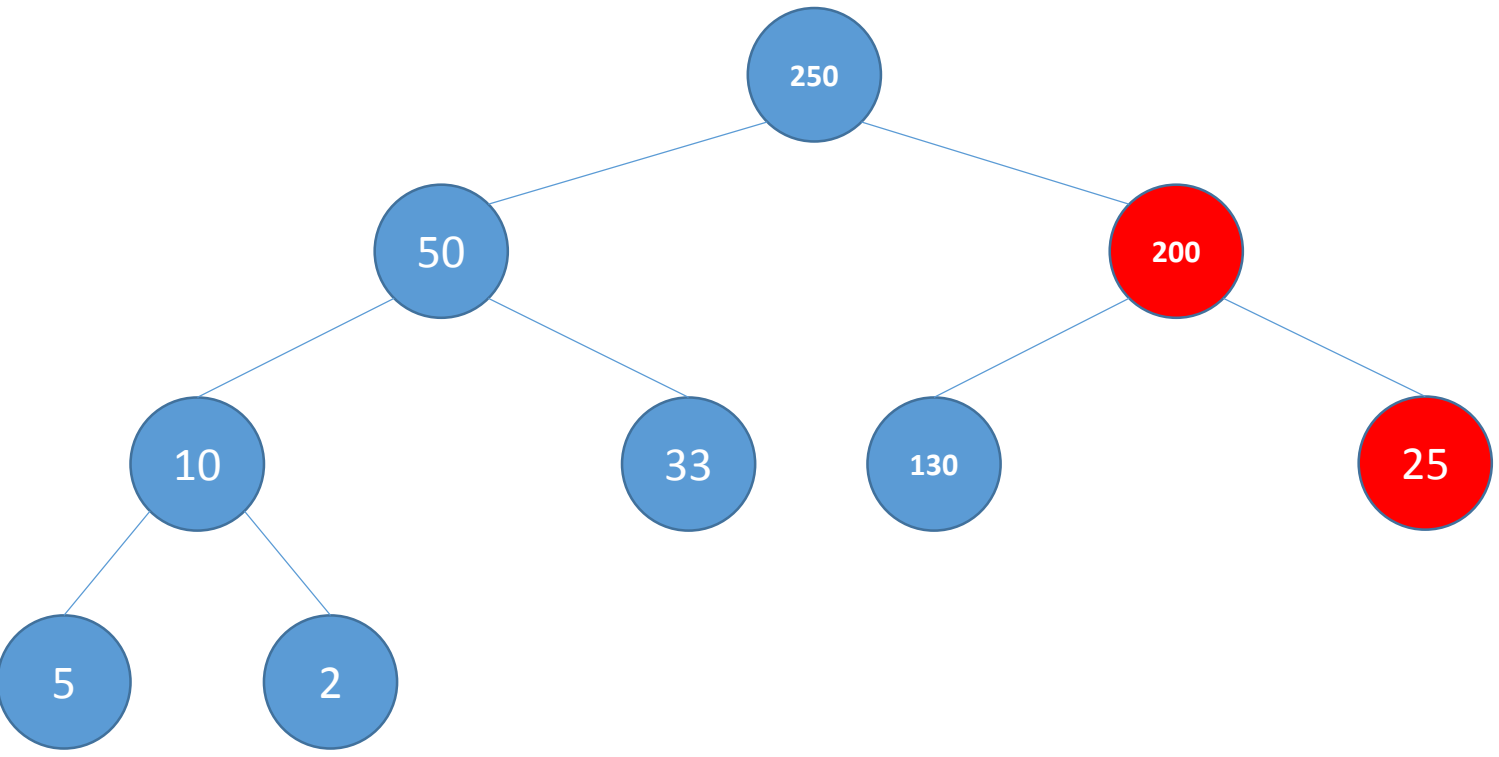
X	250	50	200	10	33	130	25	5	2	25		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 9
tempKey = 600



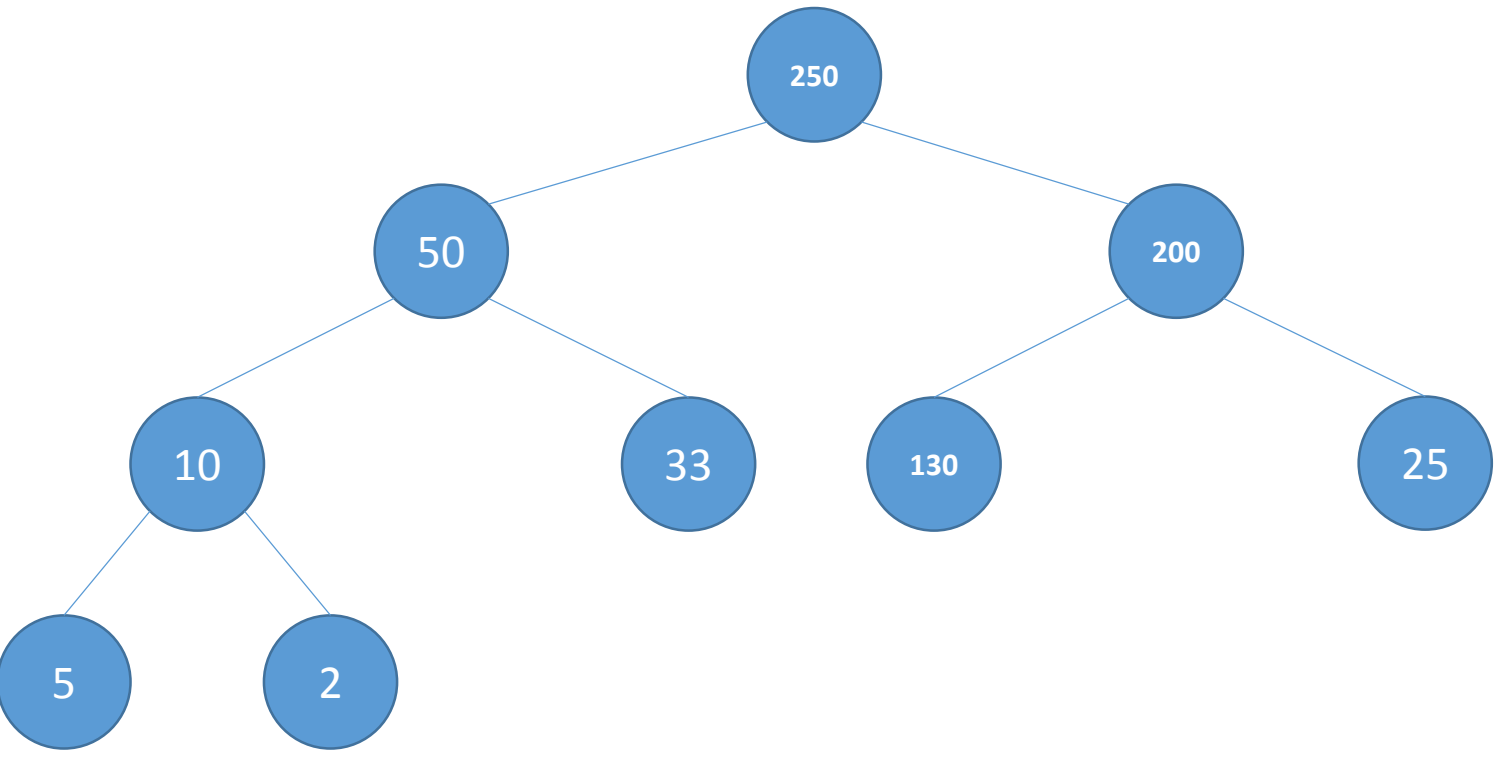
X	250	50	200	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

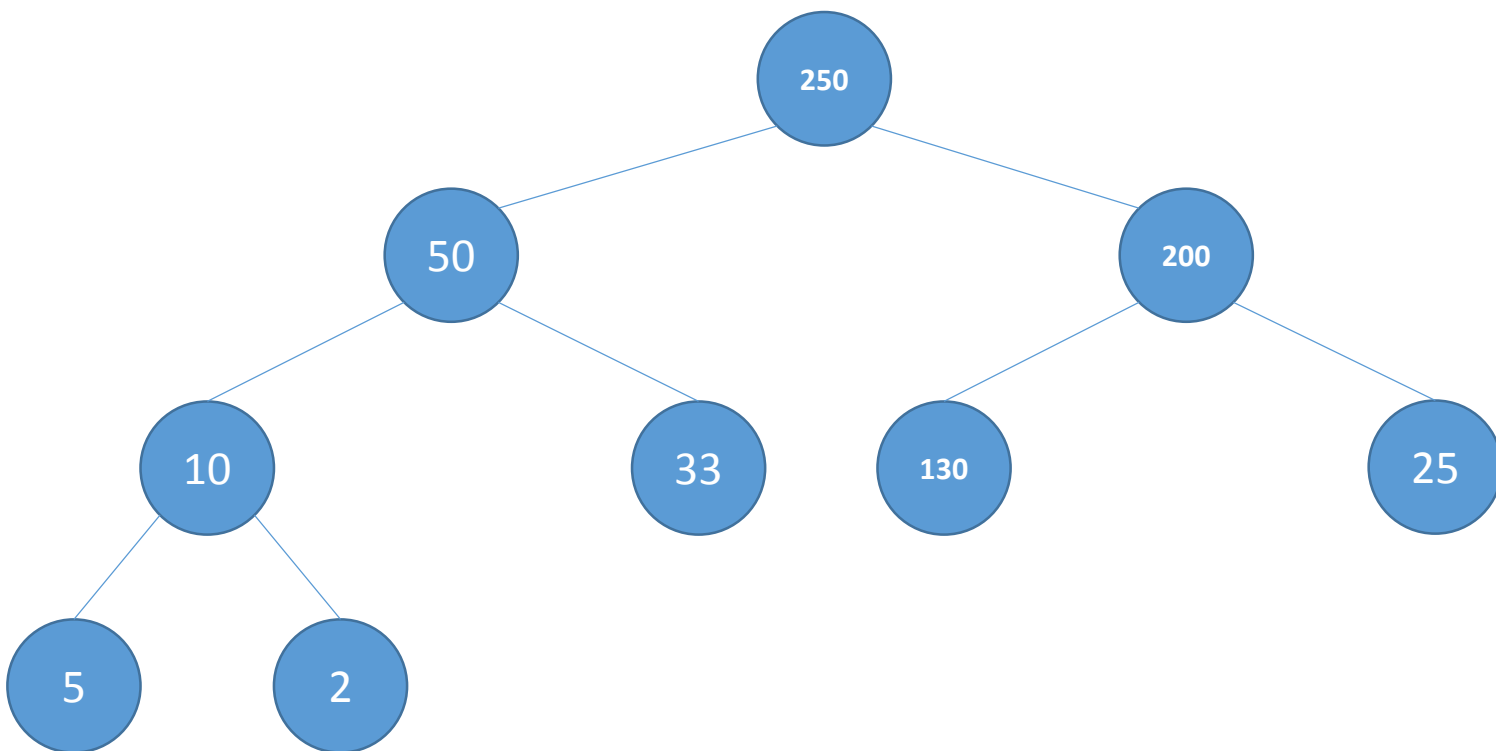
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 9
tempKey = 600



X	250	50	200	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 9

tempKey = 250

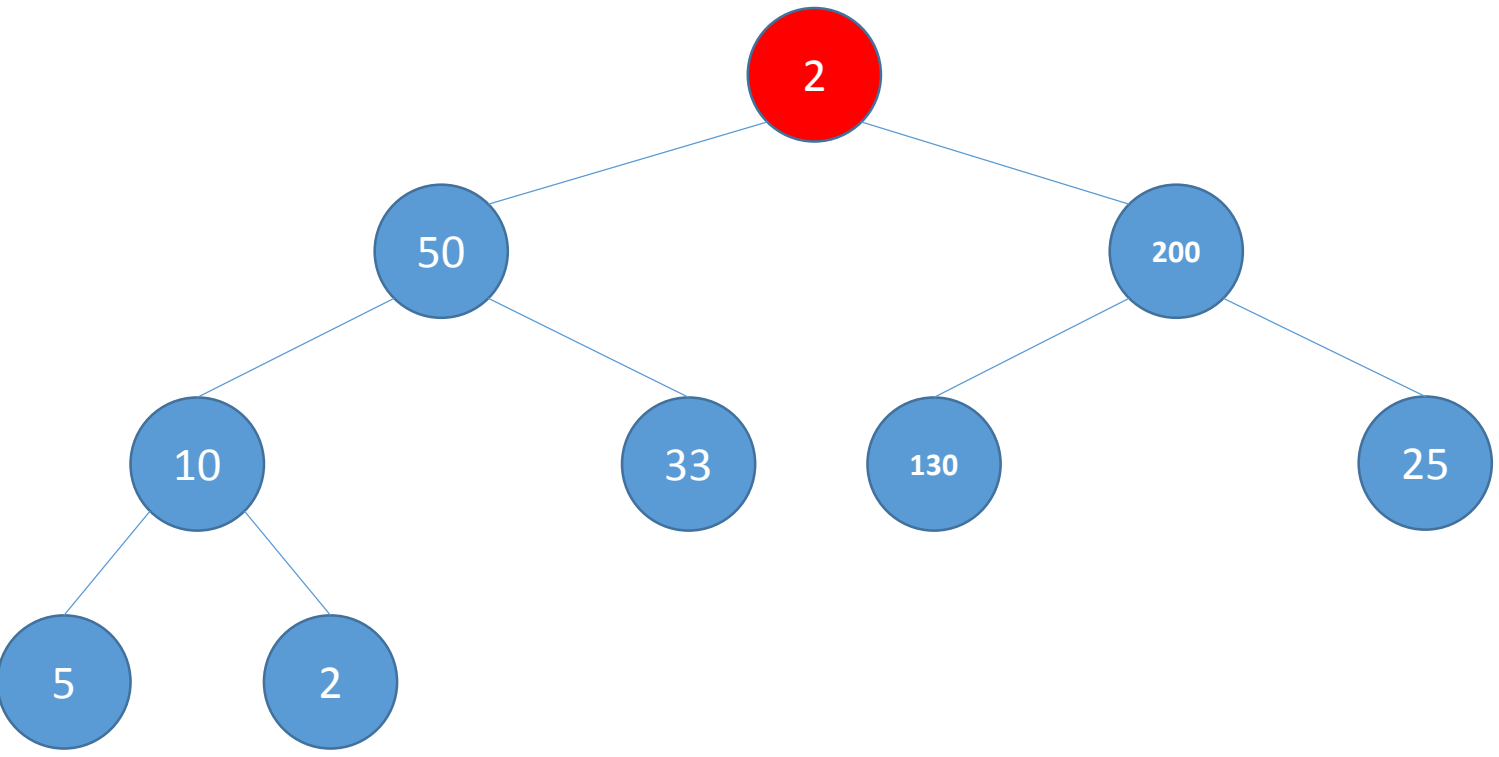
X	2	50	200	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 9
tempKey = 250



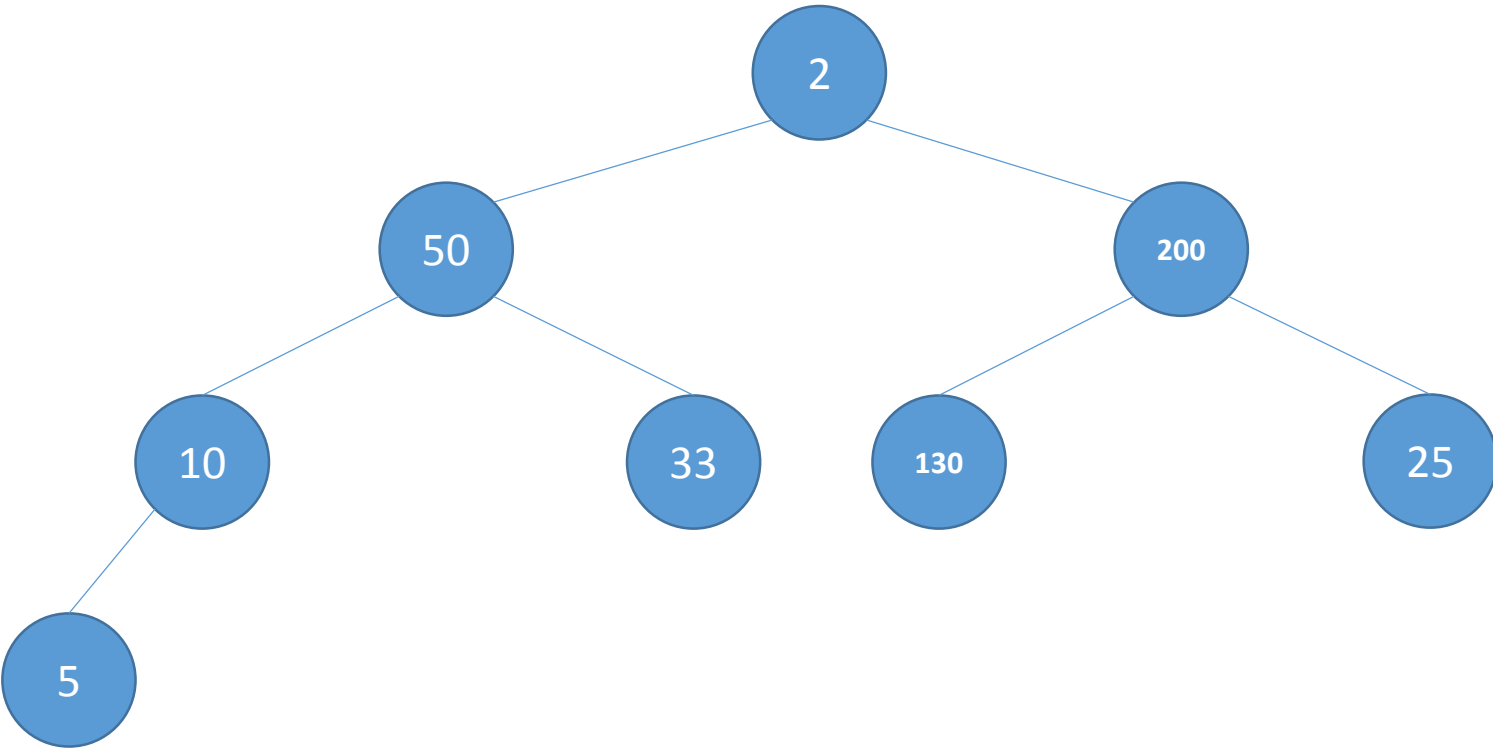
X	2	50	200	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

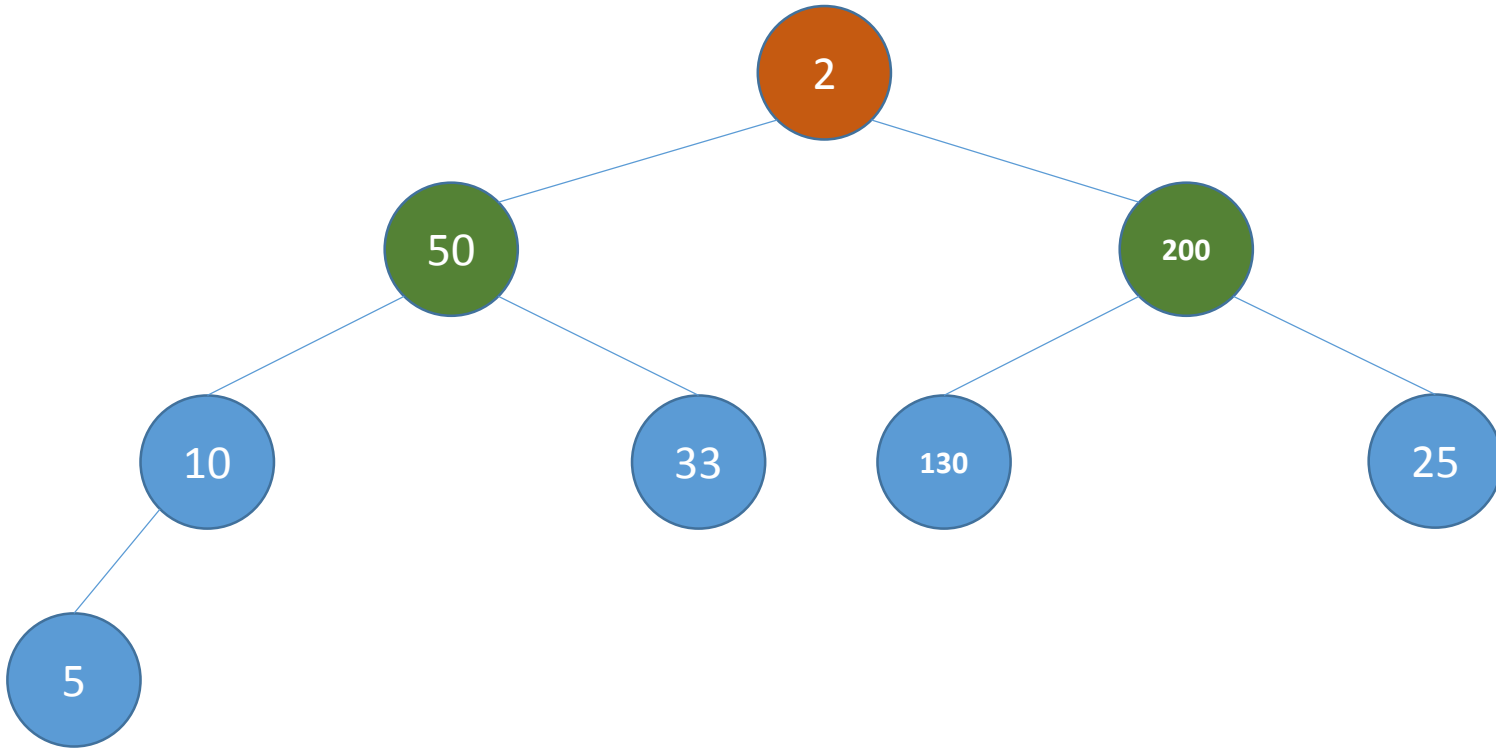
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 250



X	2	50	200	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 8
tempKey = 250

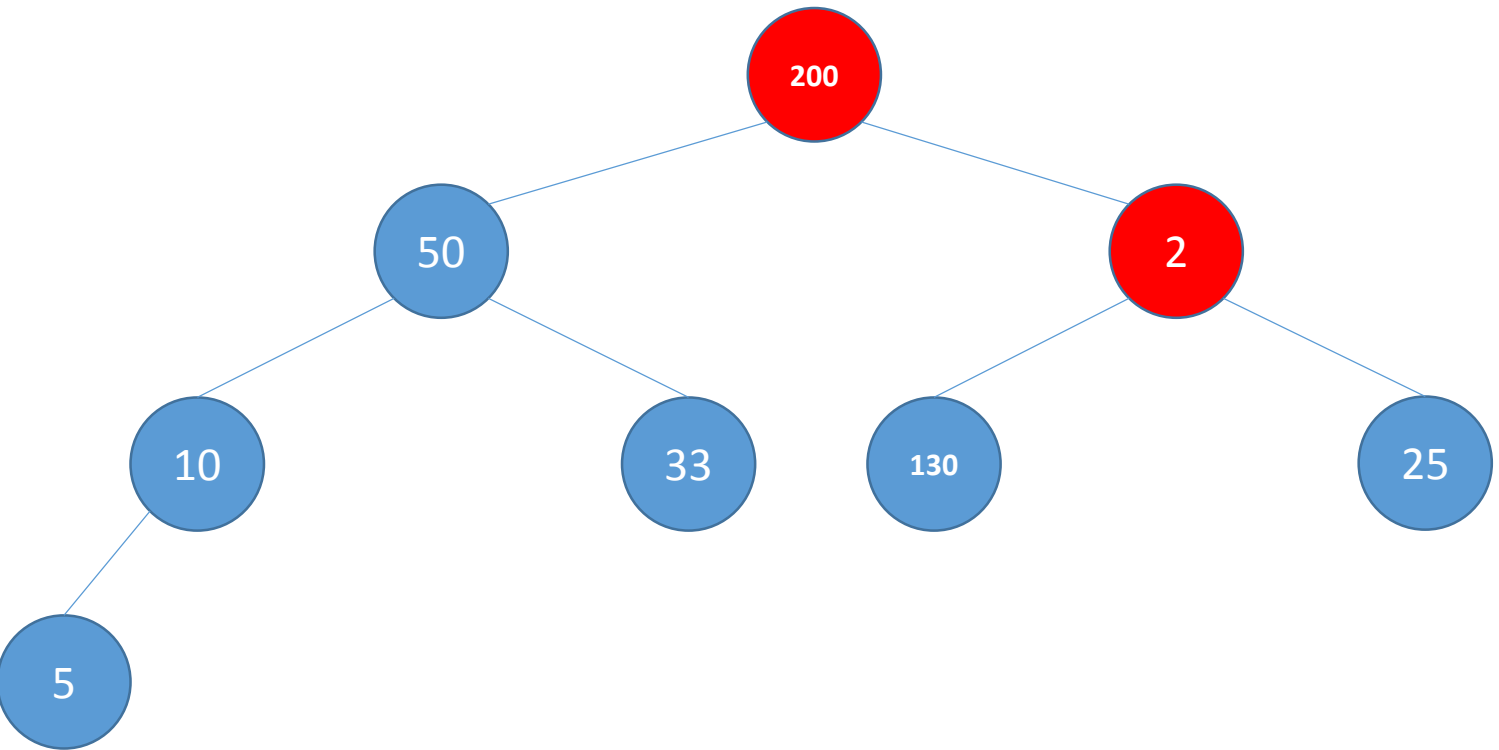
X	200	50	2	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 250



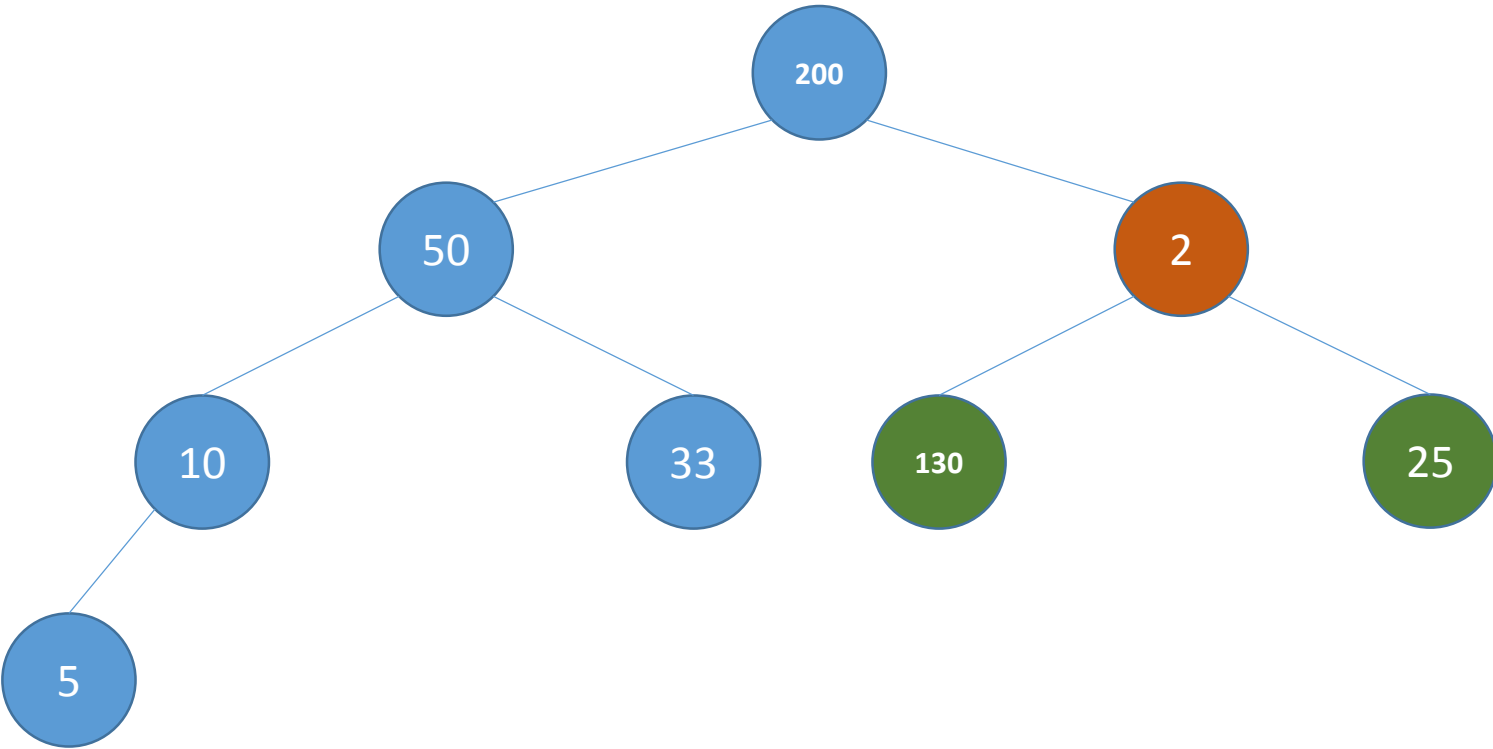
X	200	50	2	10	33	130	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 250



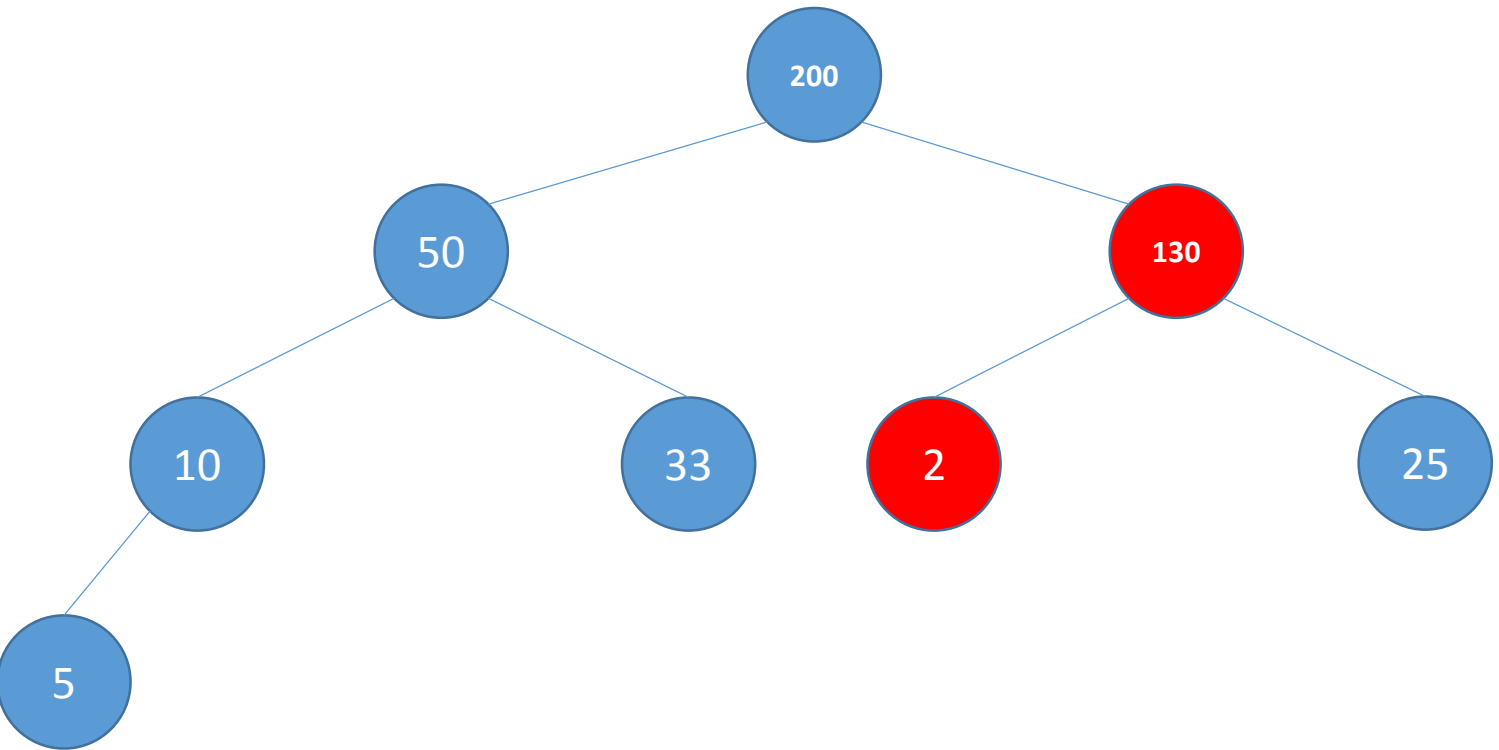
X	200	50	130	10	33	2	25	5	2	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 250



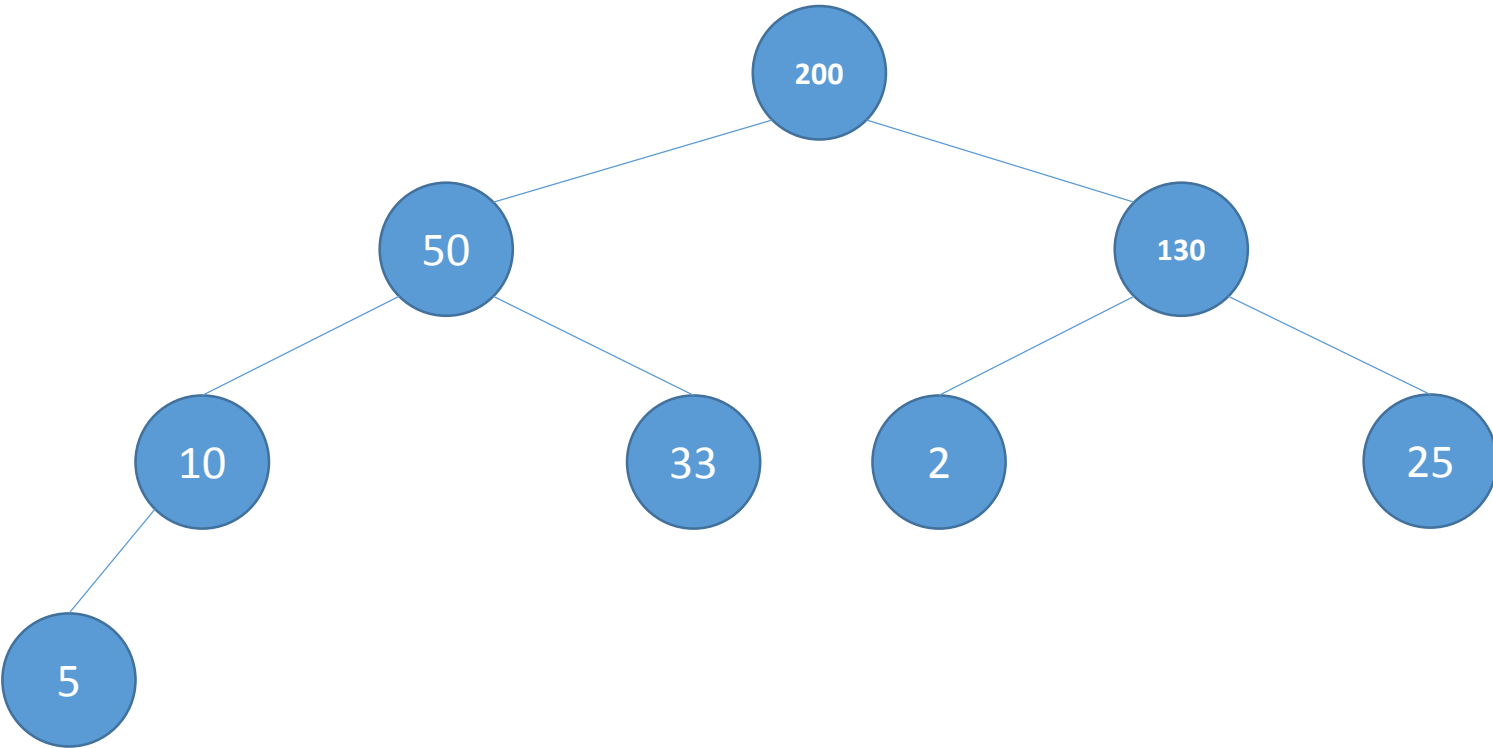
X	200	50	130	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 250



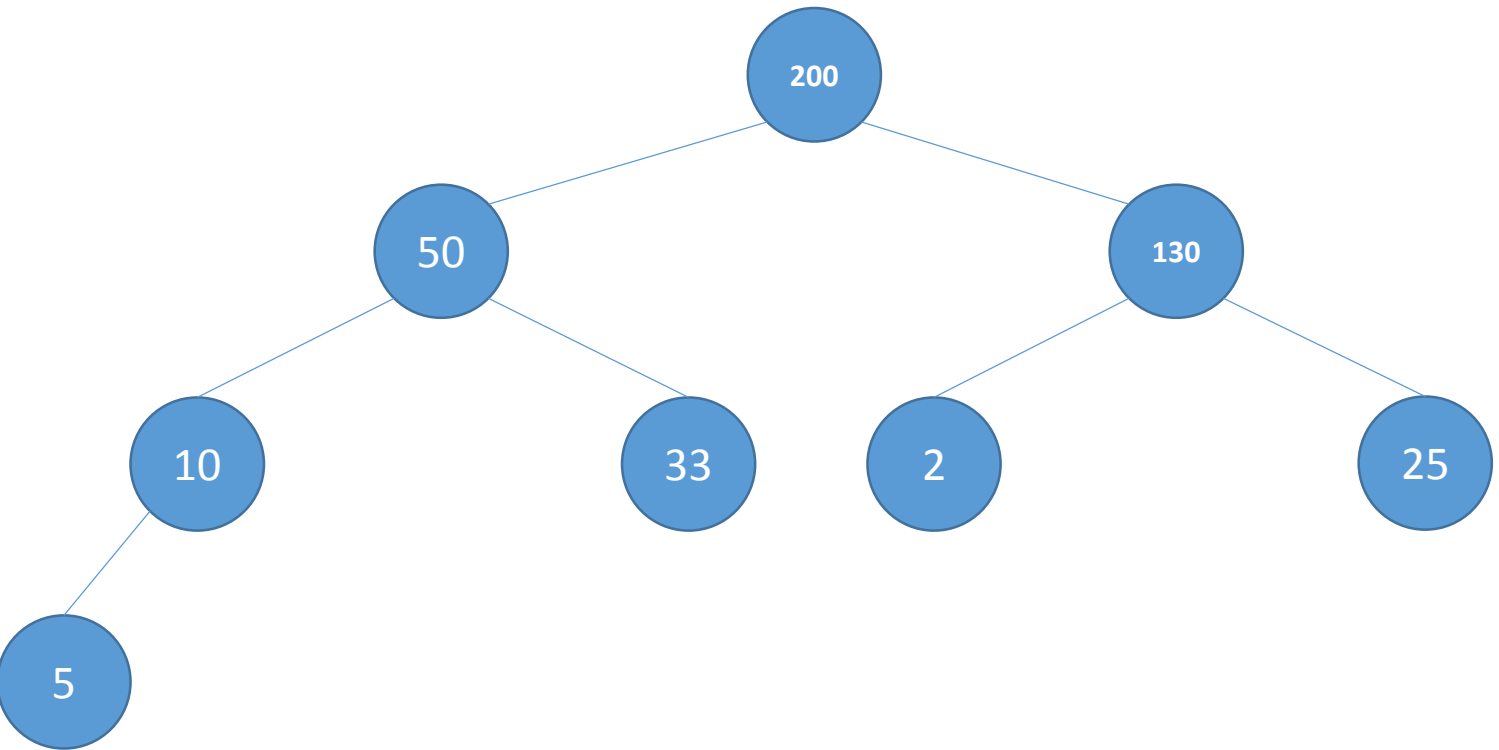
X	200	50	130	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 200



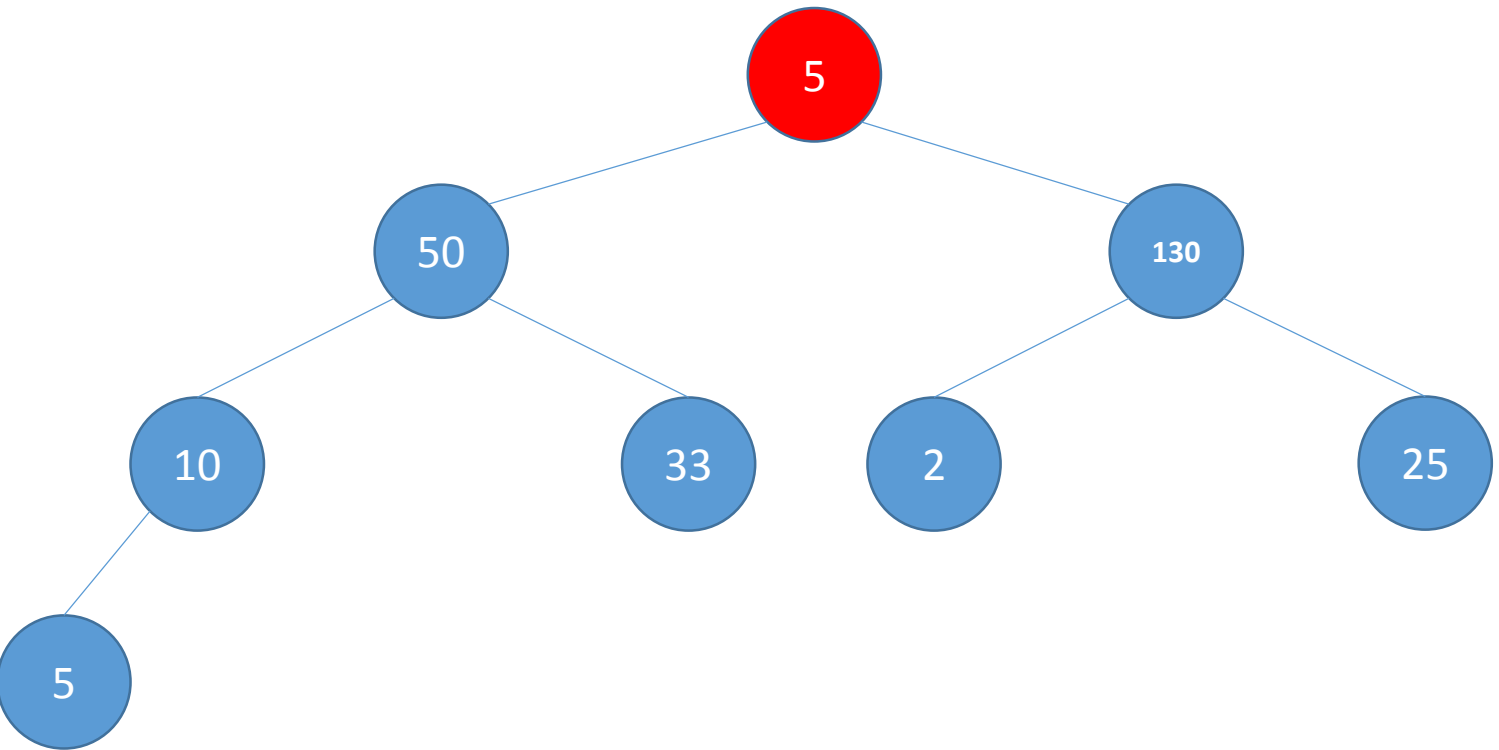
X	5	50	130	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

```

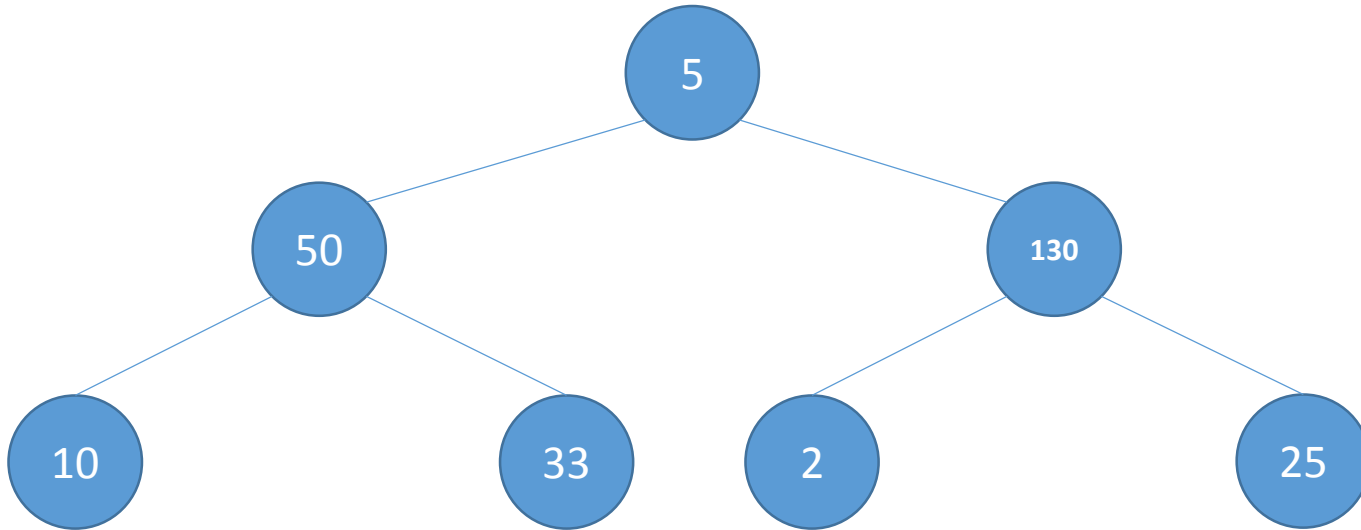
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 8
tempKey = 200



X	5	50	130	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



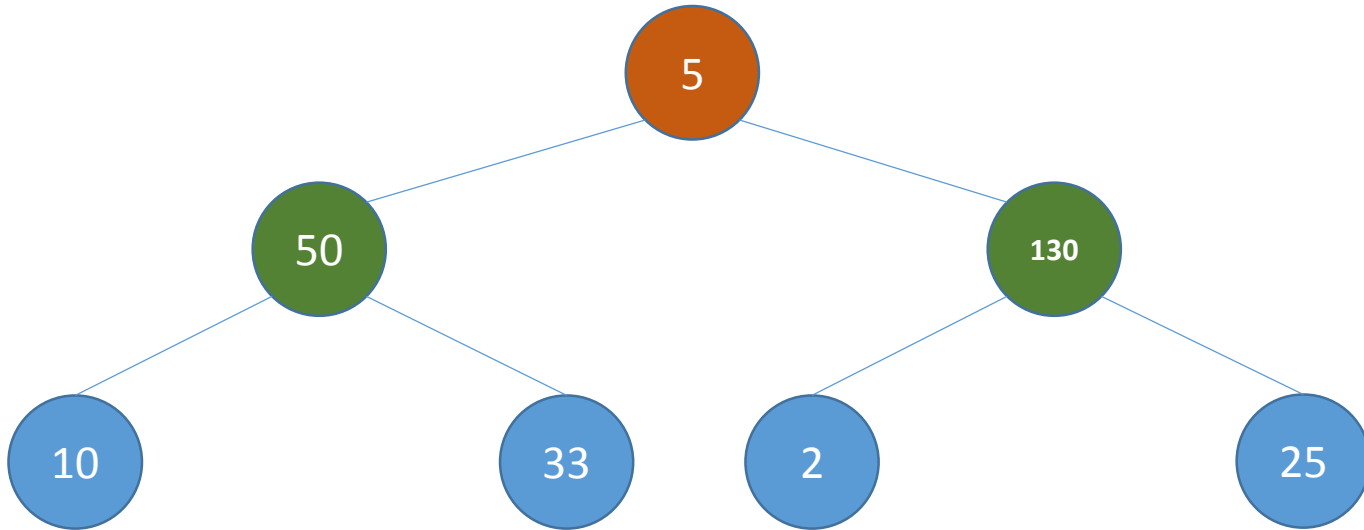
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 7
tempKey = 200

X	5	50	130	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

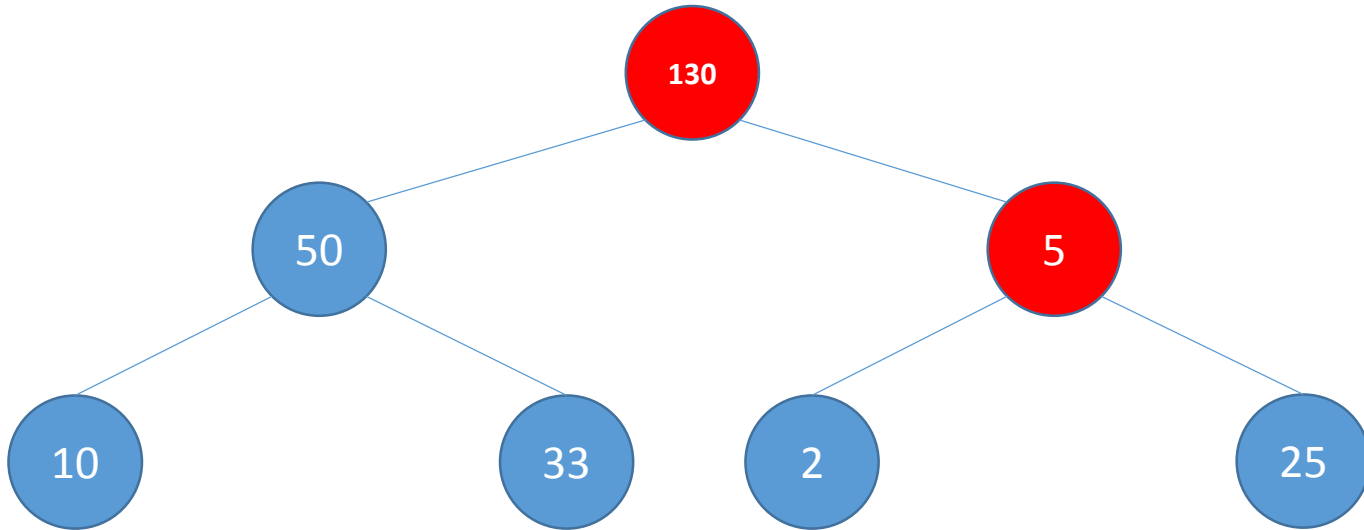


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 7
tempKey = 200

X	130	50	5	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



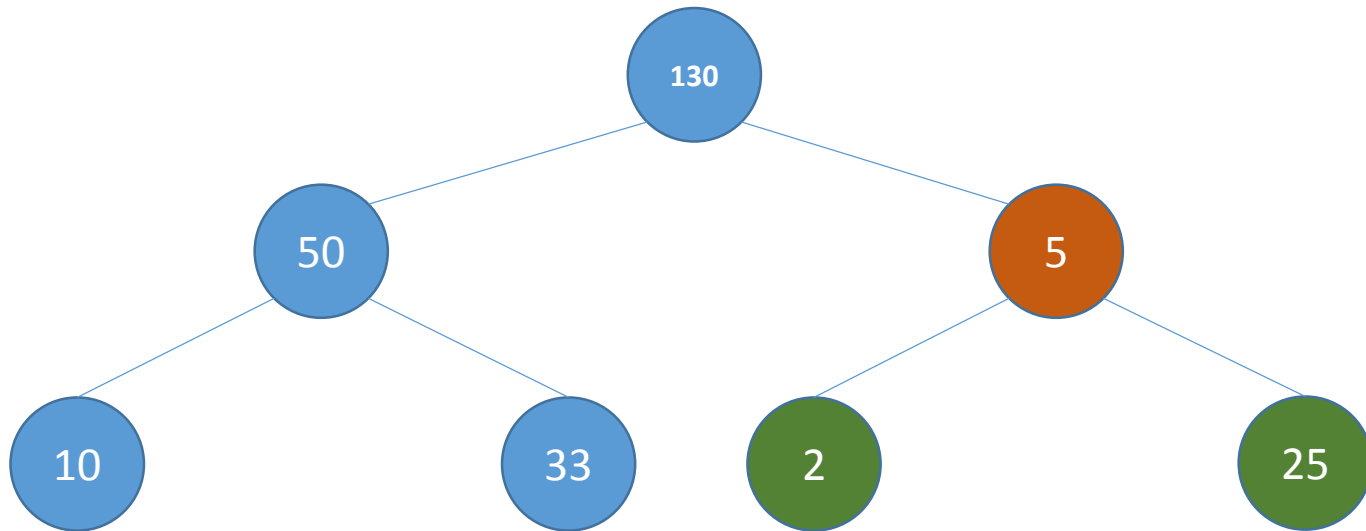
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 7
tempKey = 200

X	130	50	5	10	33	2	25	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



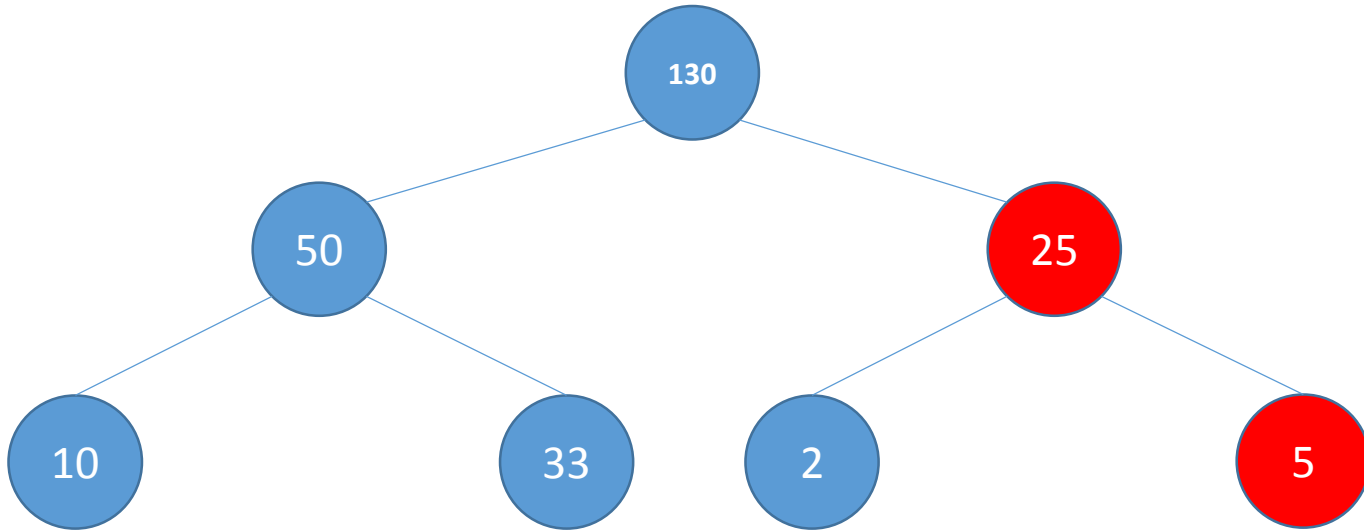
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 7
tempKey = 200

X	130	50	25	10	33	2	5	5	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



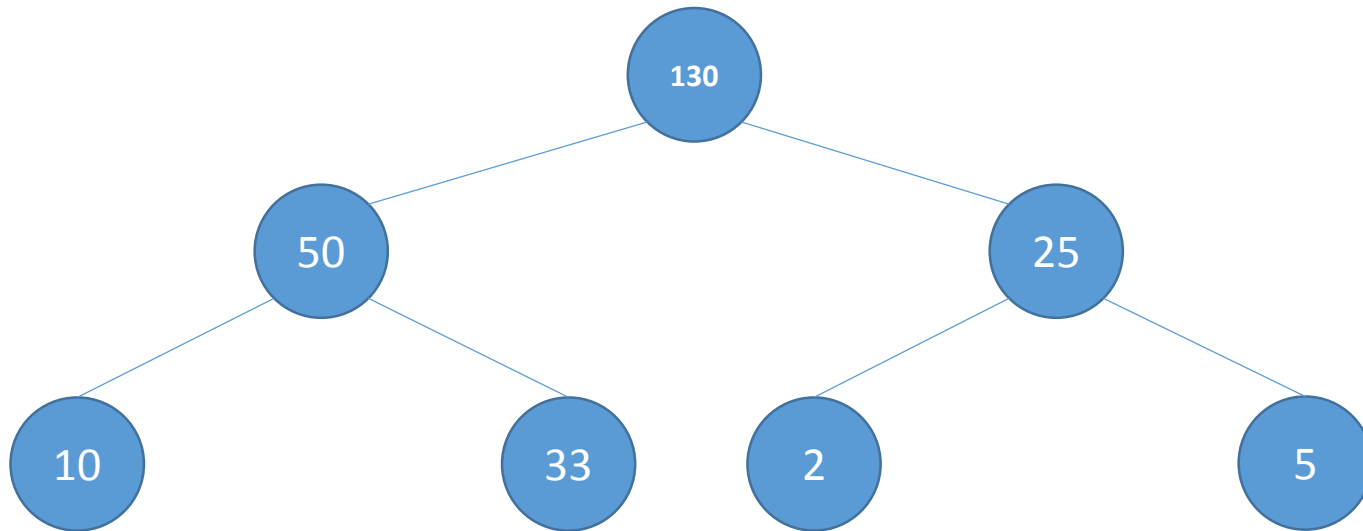
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 7
tempKey = 200

X	130	50	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



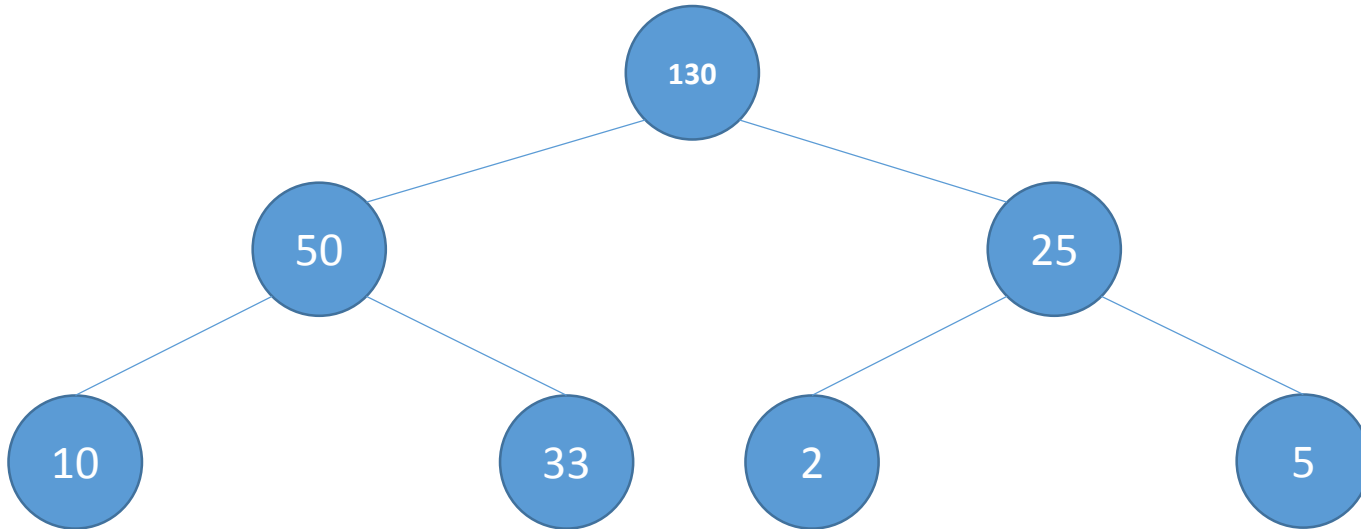
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 7
tempKey = 200

X	130	50	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



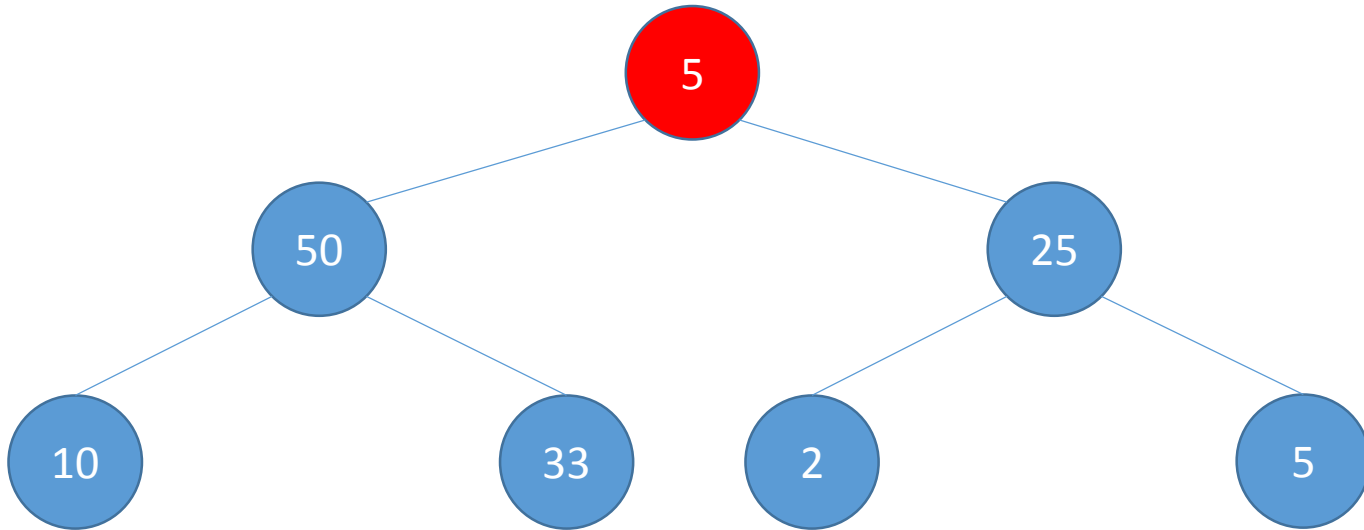
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 7

tempKey = 130

X	5	50	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

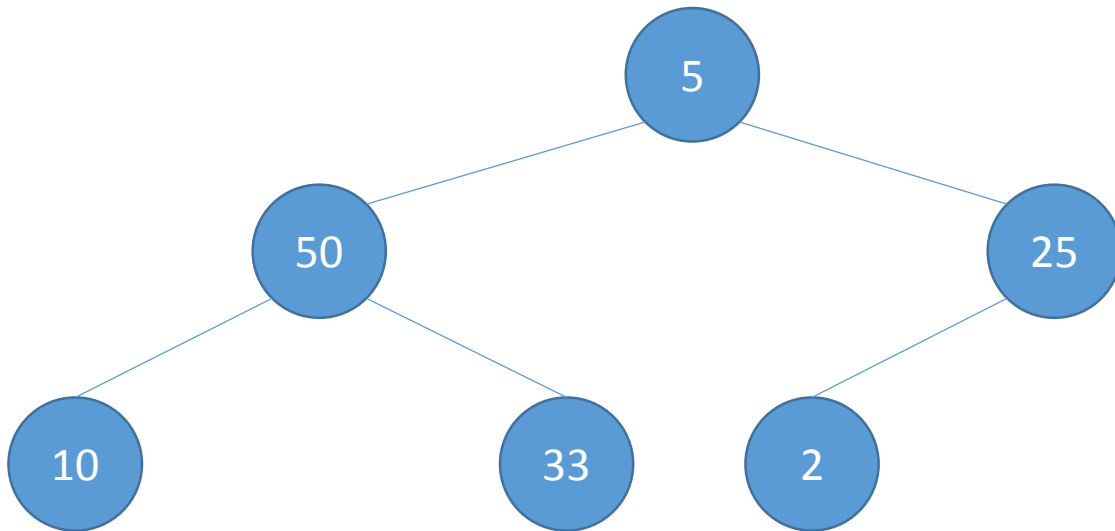


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 7
tempKey = 130

X	5	50	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



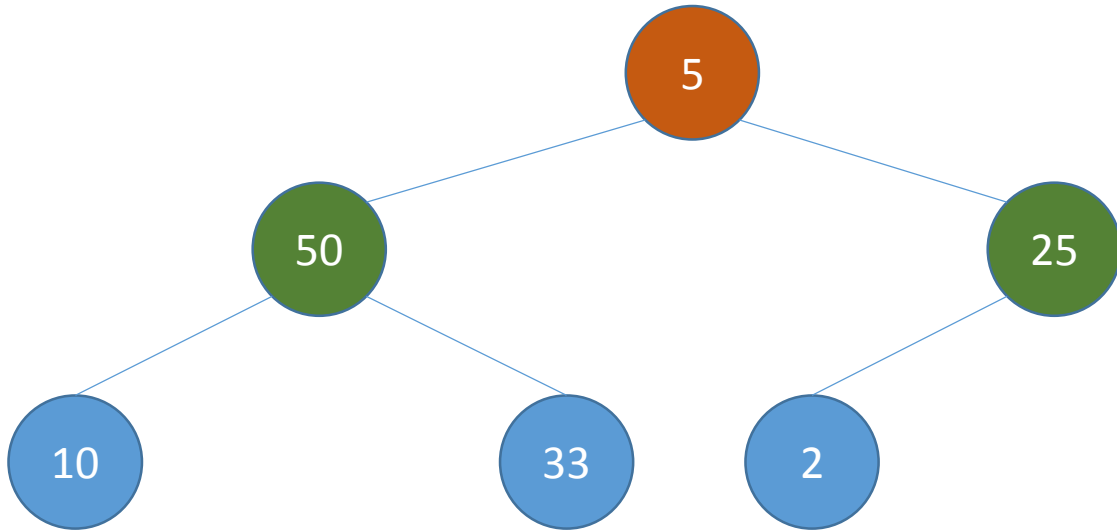
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6
tempKey = 130

X	5	50	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

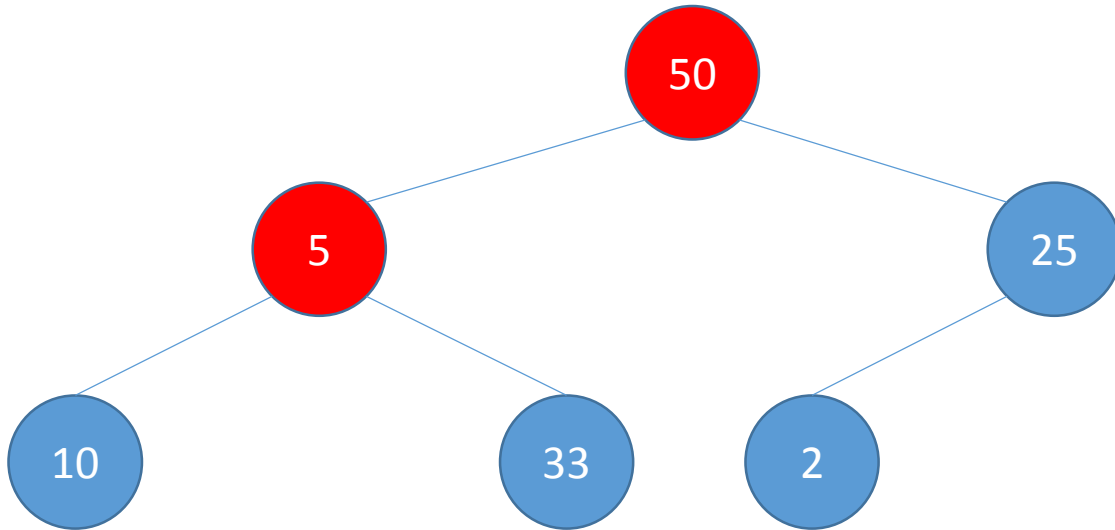


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 6
tempKey = 130

X	50	5	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



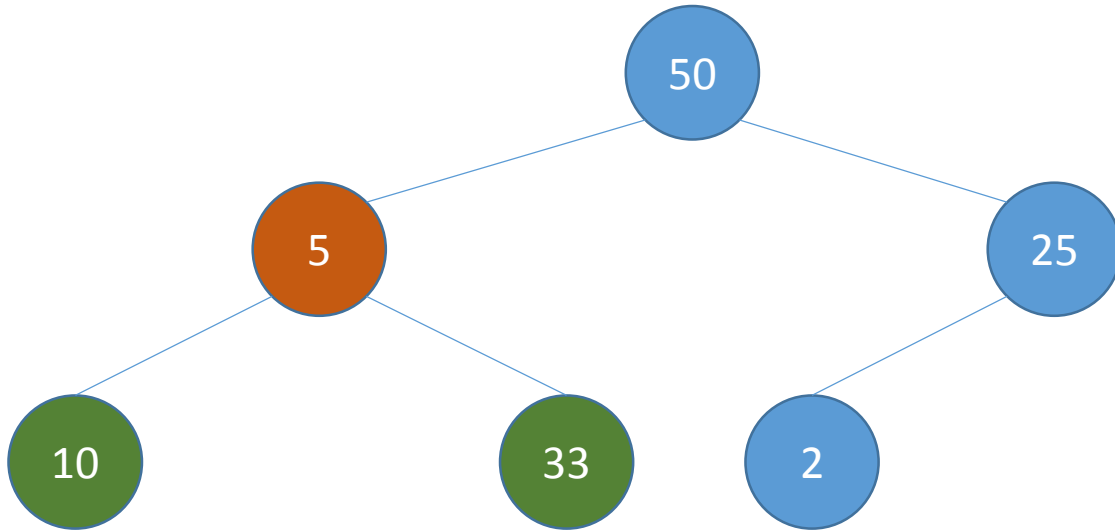
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6
tempKey = 130

X	50	5	25	10	33	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



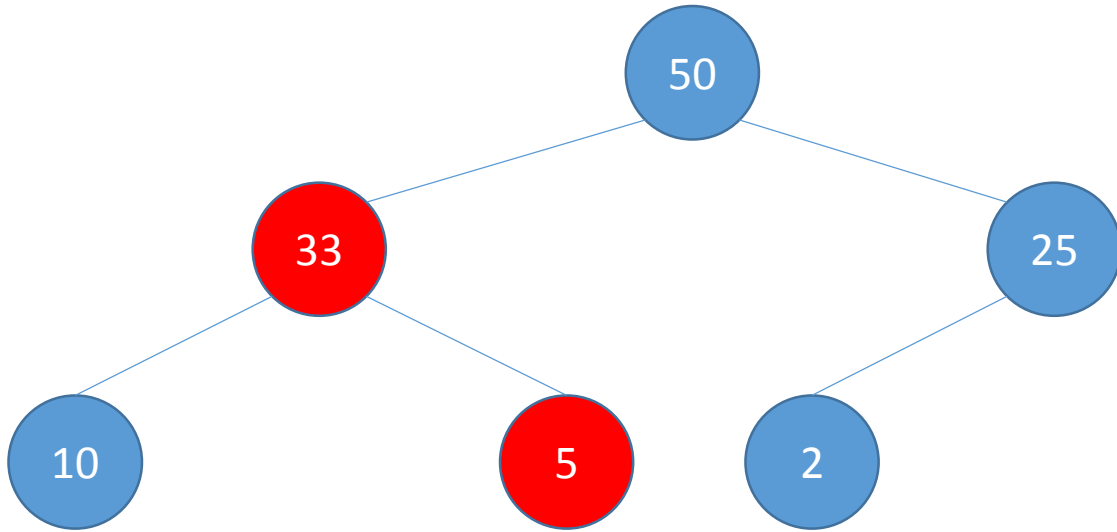
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6
tempKey = 130

X	50	33	25	10	5	2	5	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



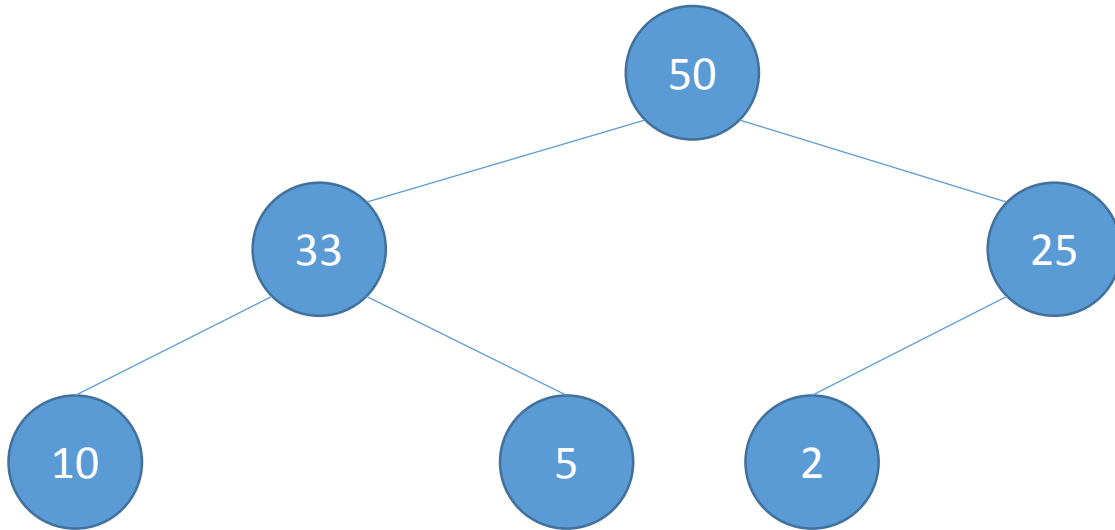
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6
tempKey = 130

X	50	33	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



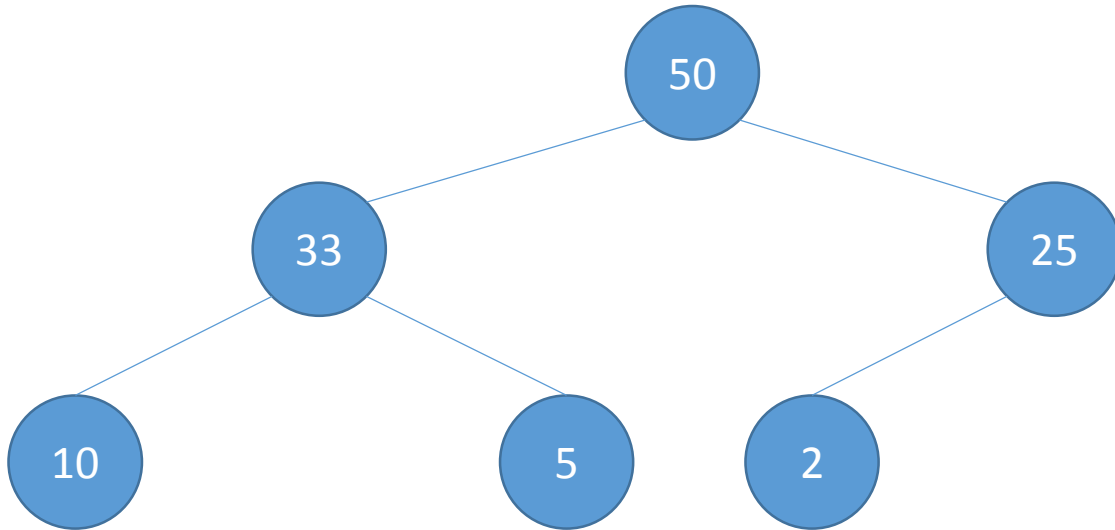
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6
tempKey = 130

X	50	33	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

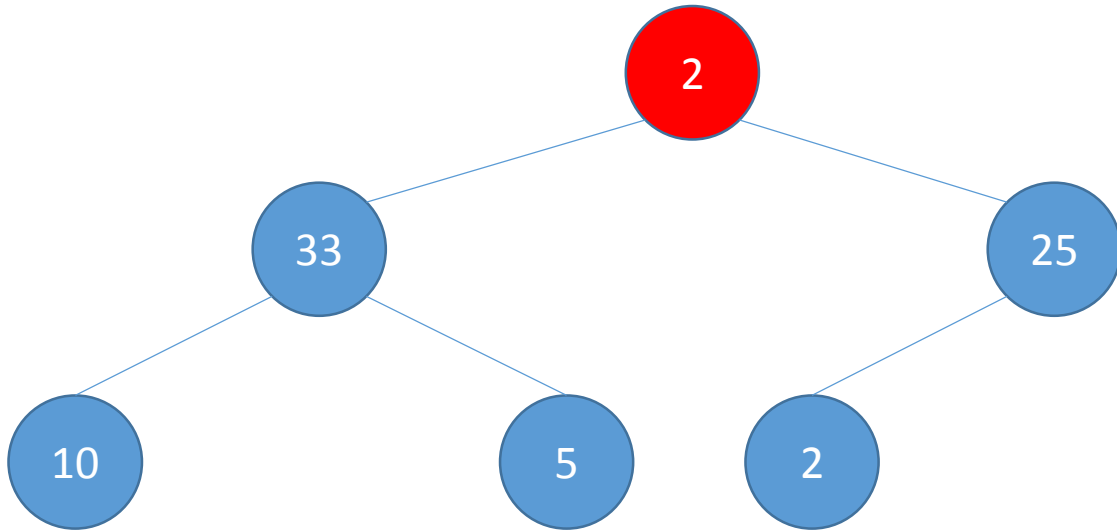
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6

tempKey = 50

X	2	33	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



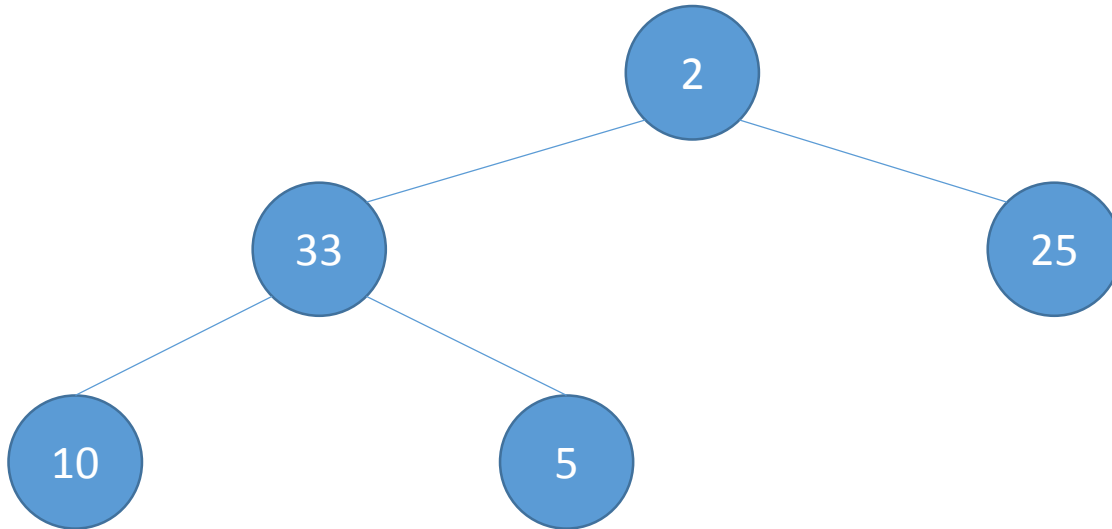
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 6
tempKey = 50

X	2	33	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



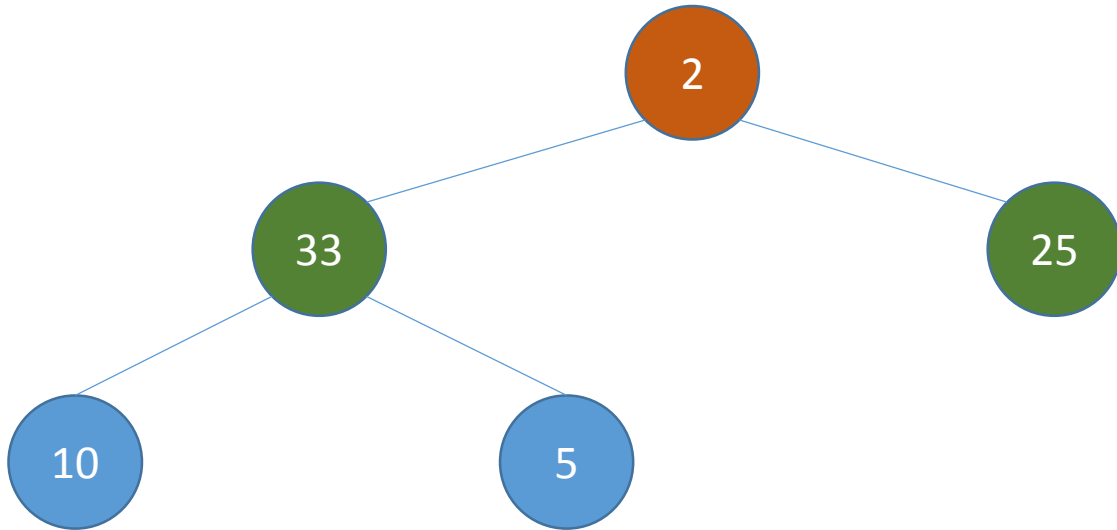
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 5
tempKey = 50

X	2	33	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

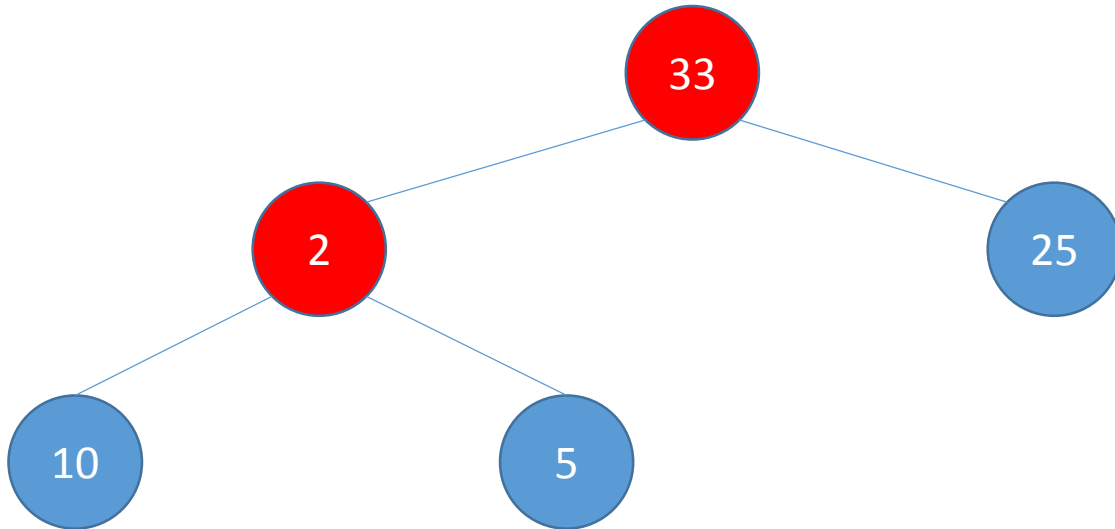


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 5
tempKey = 50

X	33	2	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



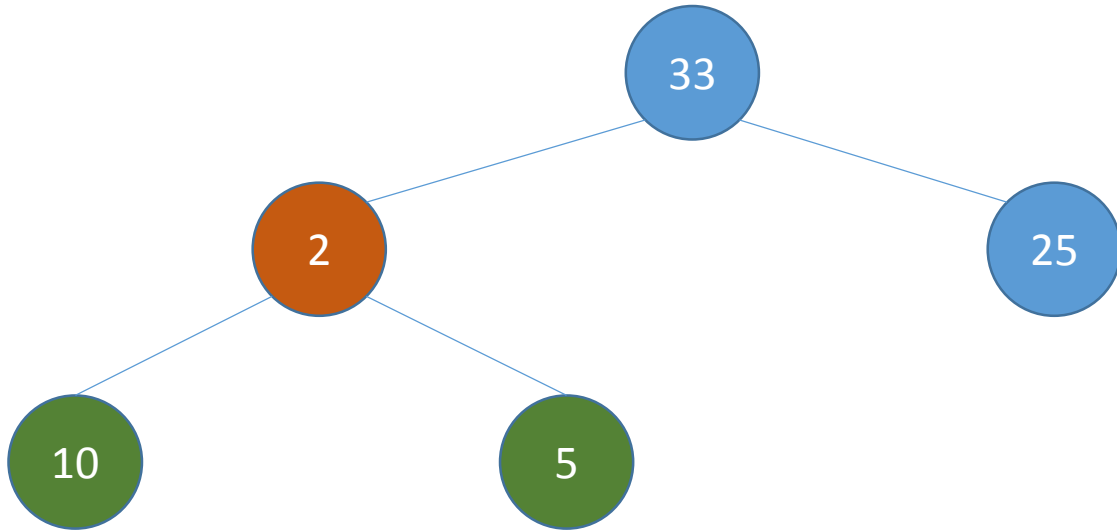
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 5
tempKey = 50

X	33	2	25	10	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

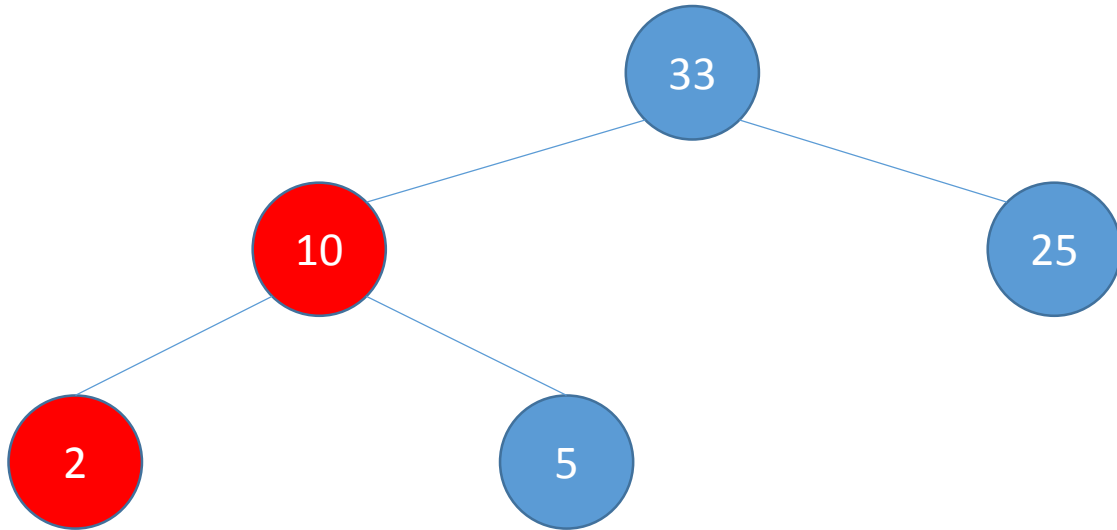


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 5
tempKey = 50

X	33	10	25	2	5	2	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



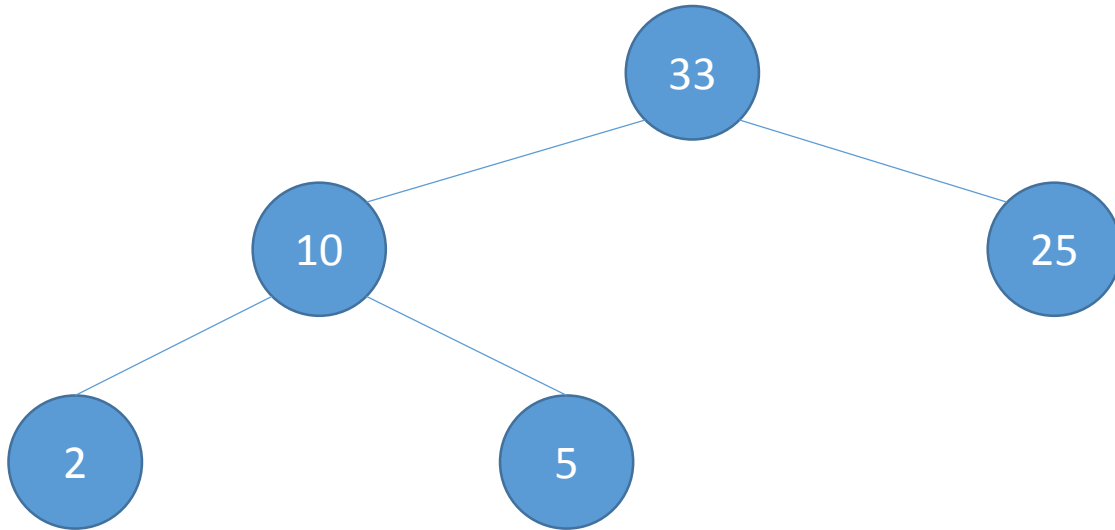
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 5
tempKey = 50

X	33	10	25	2	5	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



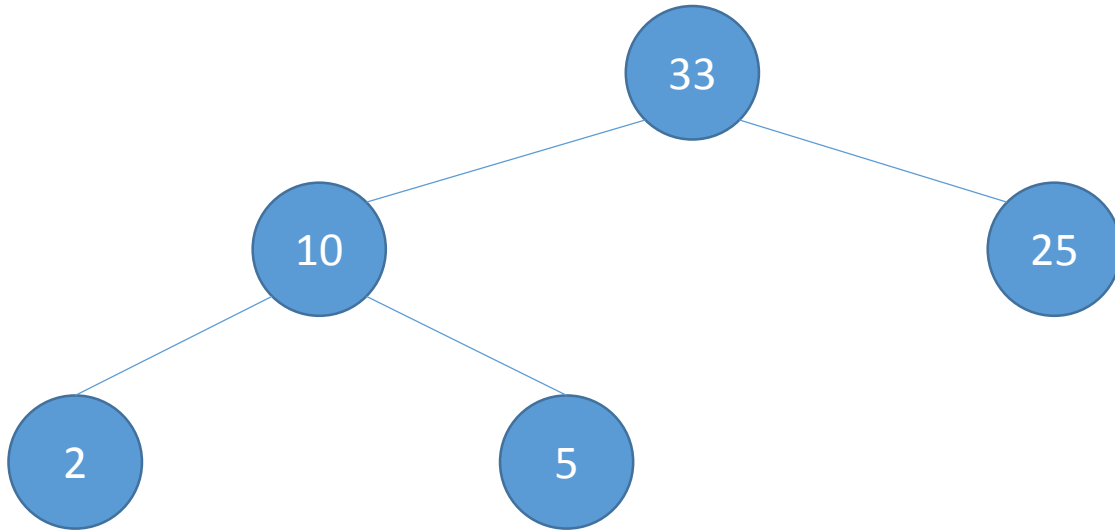
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 5
tempKey = 50

X	33	10	25	2	5	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

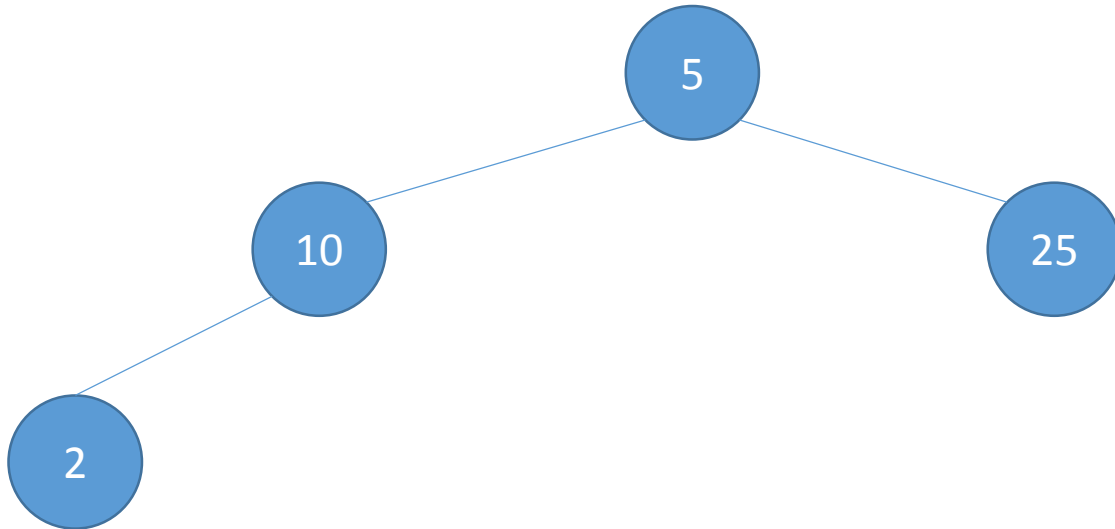
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 5

tempKey = 33

X	5	10	25	2	5	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

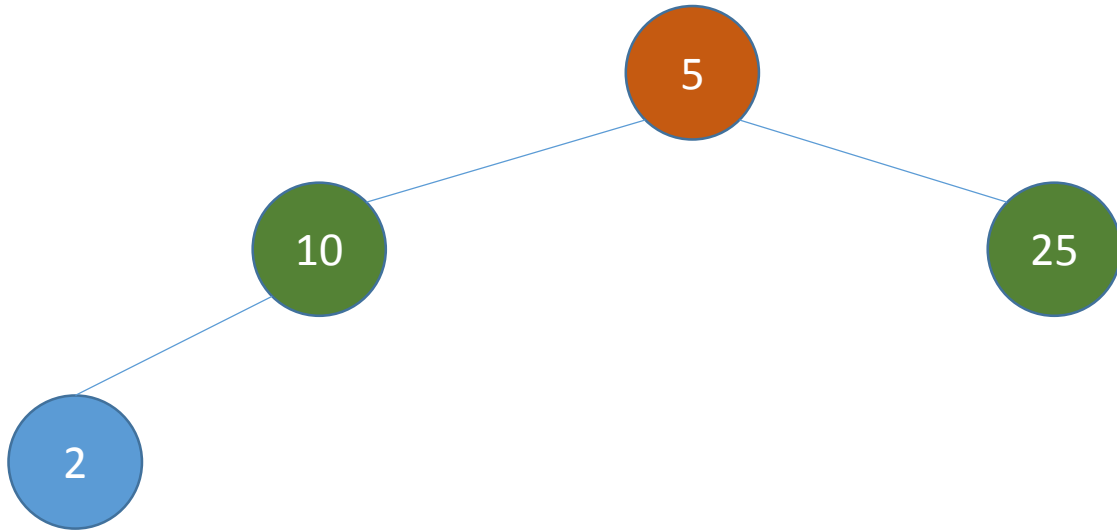


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 4
tempKey = 33

X	5	10	25	2	5	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

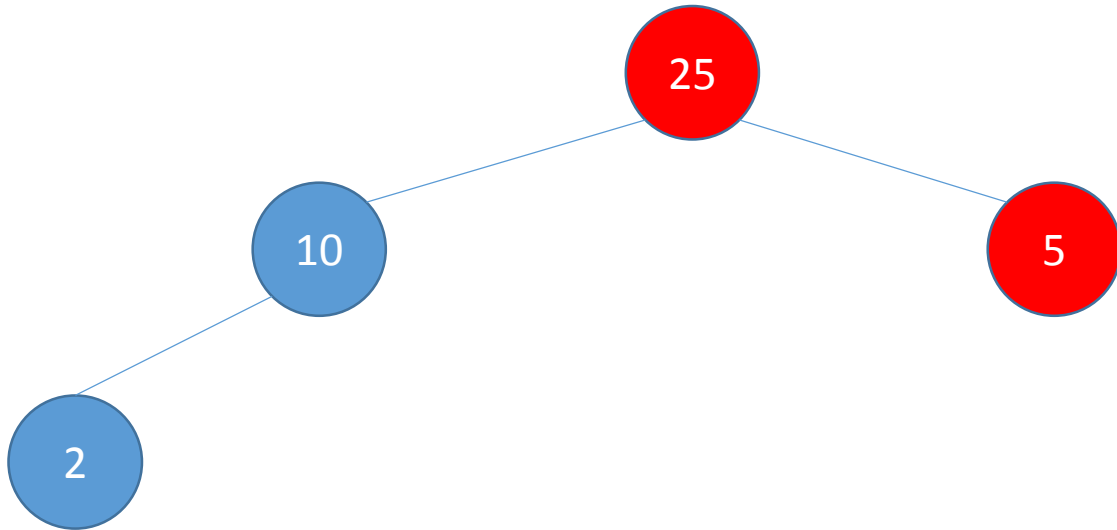


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 4
tempKey = 33

X	25	10	5	2	5	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

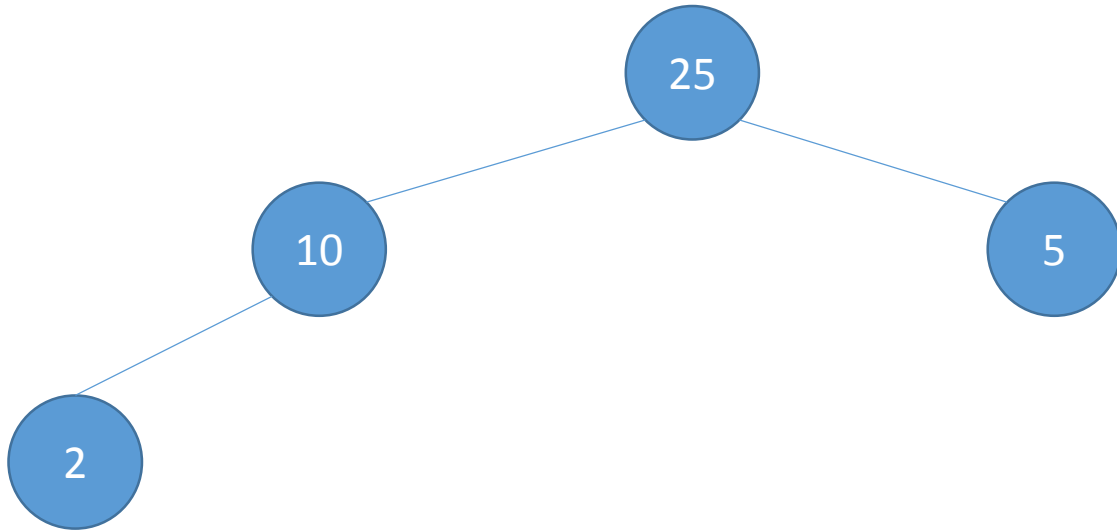


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 4
tempKey = 33

X	25	10	5	2	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

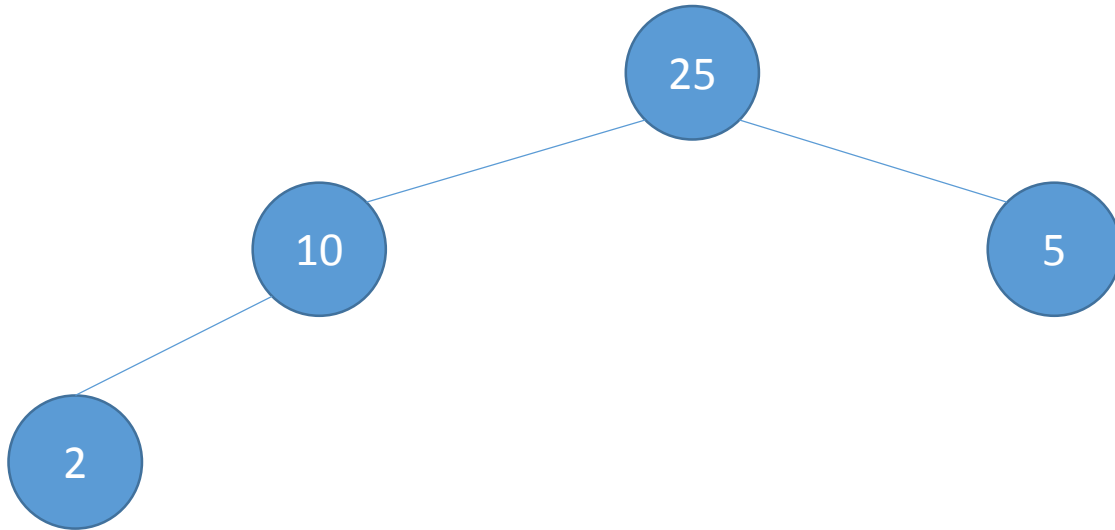


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 4
tempKey = 33

X	25	10	5	2	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



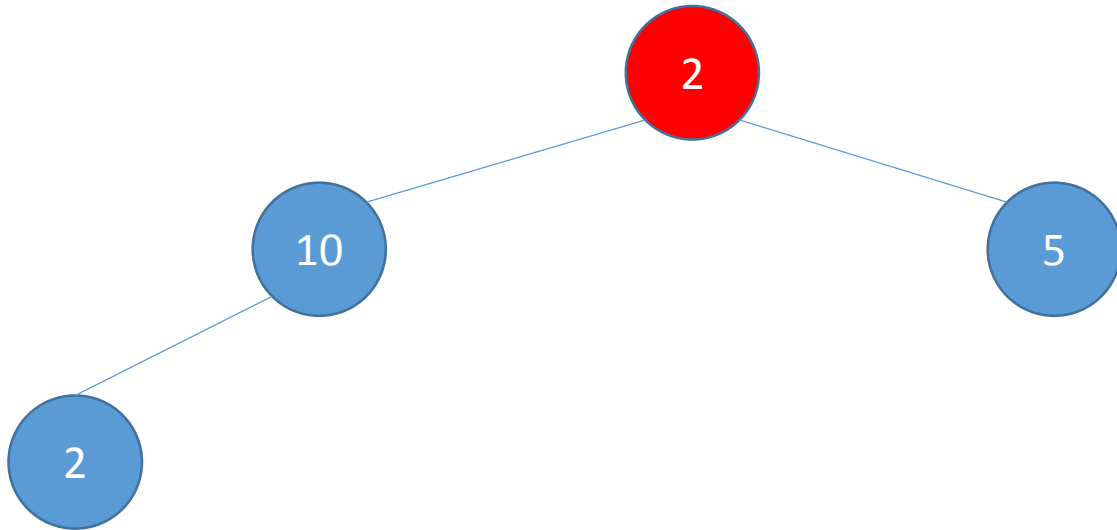
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 4

tempKey = 25

X	2	10	5	2	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



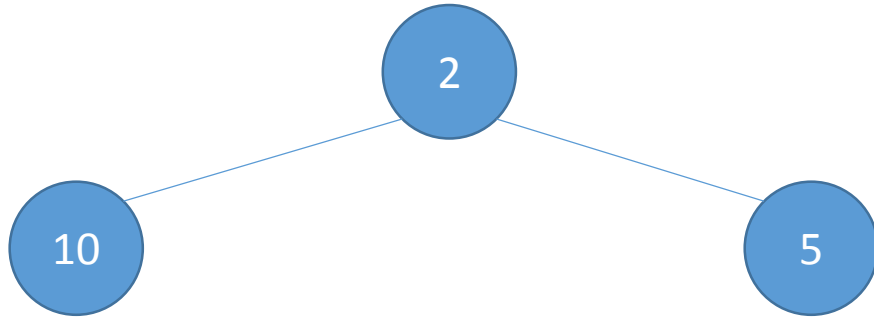
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 4
tempKey = 25

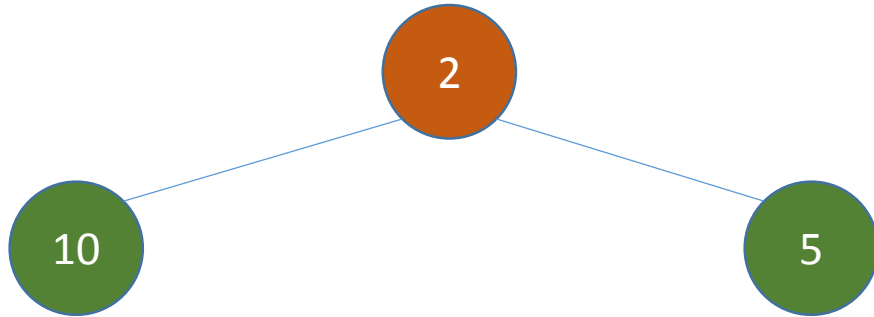
X	2	10	5	2	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
```

Size = 3
tempKey = 25

X	2	10	5	2	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

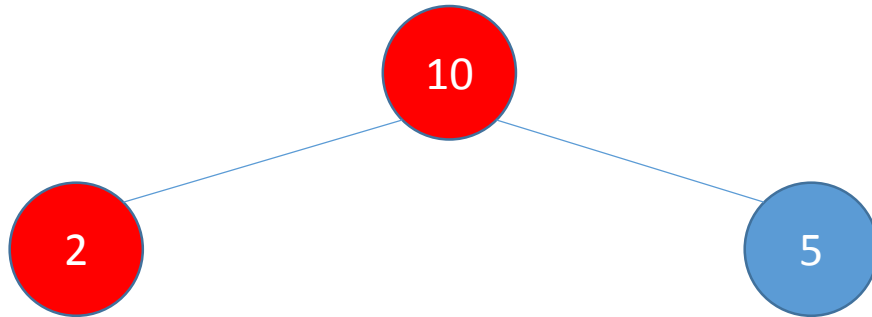


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 3
tempKey = 25

X	10	2	5	2	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

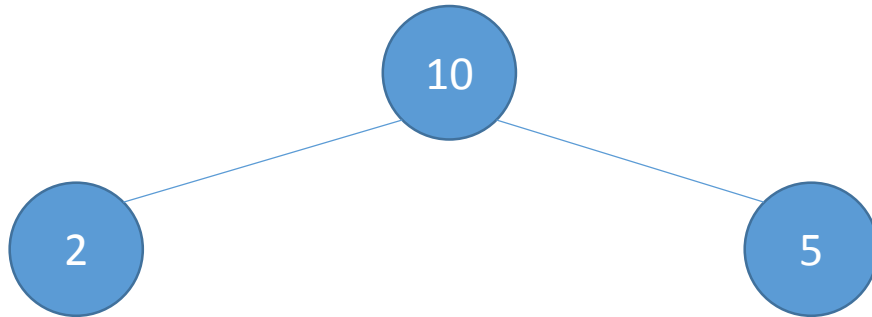


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 3
tempKey = 25

X	10	2	5	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

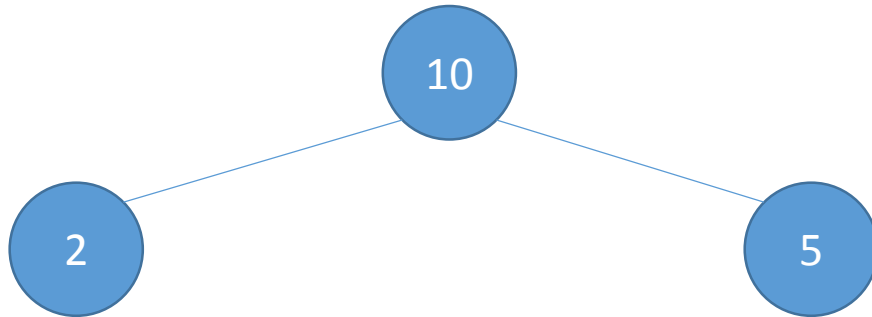


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 3
tempKey = 25

X	10	2	5	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



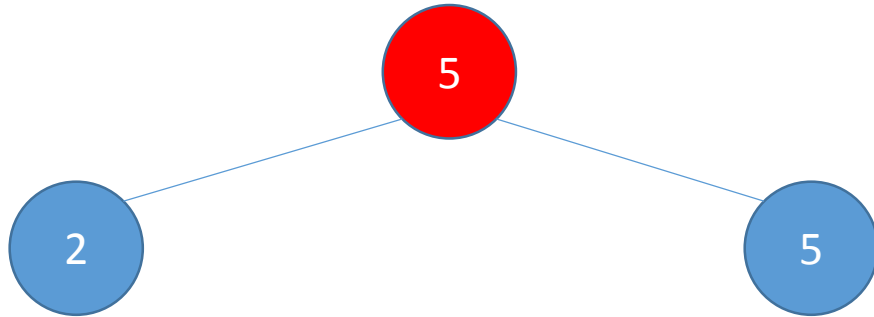
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 3

tempKey = 10

X	5	2	5	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



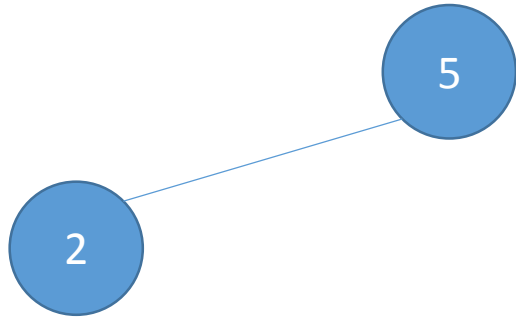
```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 3
tempKey = 10

X	5	2	5	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

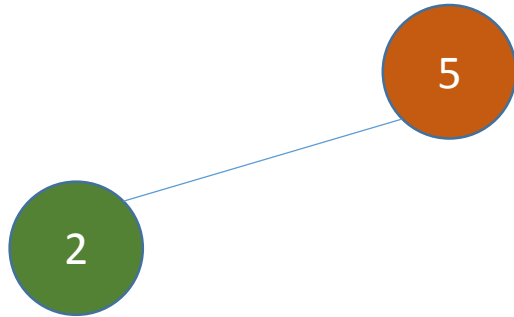


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 2
tempKey = 10

X	5	2	5	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

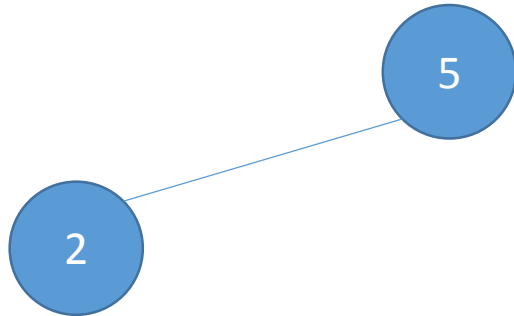


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 2
tempKey = 10

X	5	2	5	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

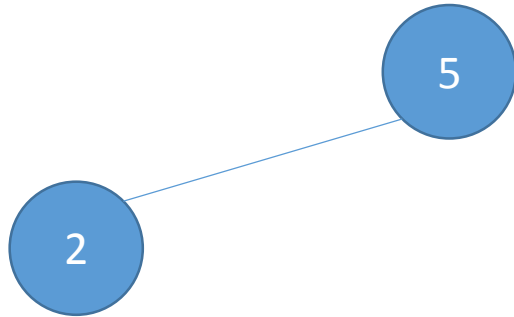


```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
  
```

Size = 2
tempKey = 10

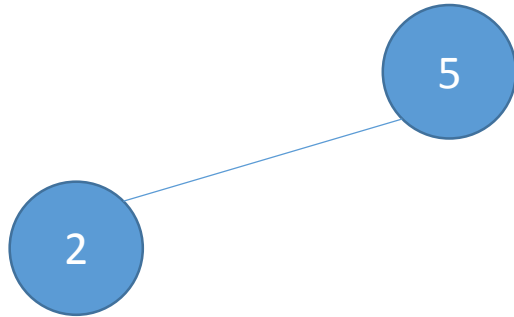
X	5	2	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
```

Size = 2
tempKey = 10

X	5	2	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

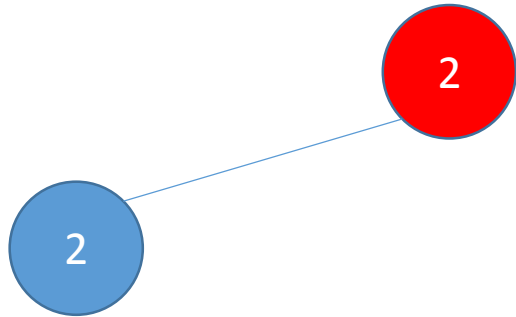


```
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
```

Size = 2

tempKey = 5

X	2	2	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 2
tempKey = 5

X	2	2	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```

public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}

```

Size = 1

tempKey = 5

X	2	2	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12



```
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
```

Size = 1
tempKey = 5

X	2	5	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

2

```
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
```

Size = 1
tempKey = 5

X	2	5	10	25	33	50	130	200	250	600		
0	1	2	3	4	5	6	7	8	9	10	11	12

2

```
public void sort(){
    int n= size;
    for(int i= 1; i < n; i++){
        int tmpKey= keys[1];
        T tmpData= data[1];
        keys[1]= keys[size];
        data[1]= data[size];
        size--;
        siftDown(1);
        keys[size+1]= tmpKey;
        data[size+1]= tmpData;
    }
}
```

Size = 1
tempKey = 5