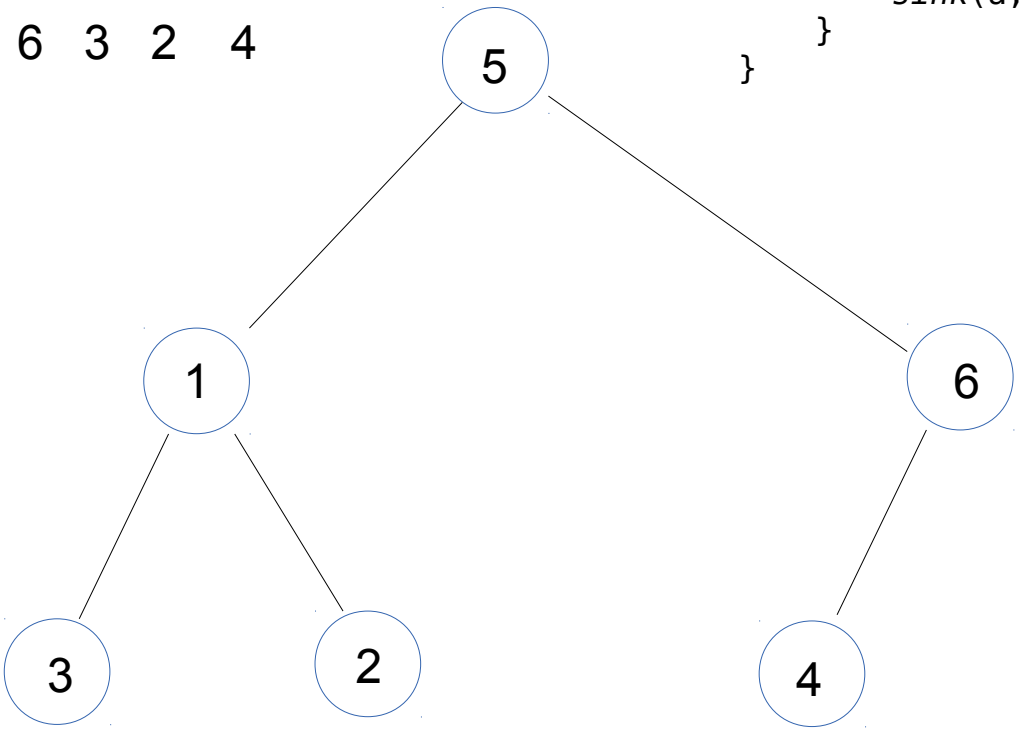
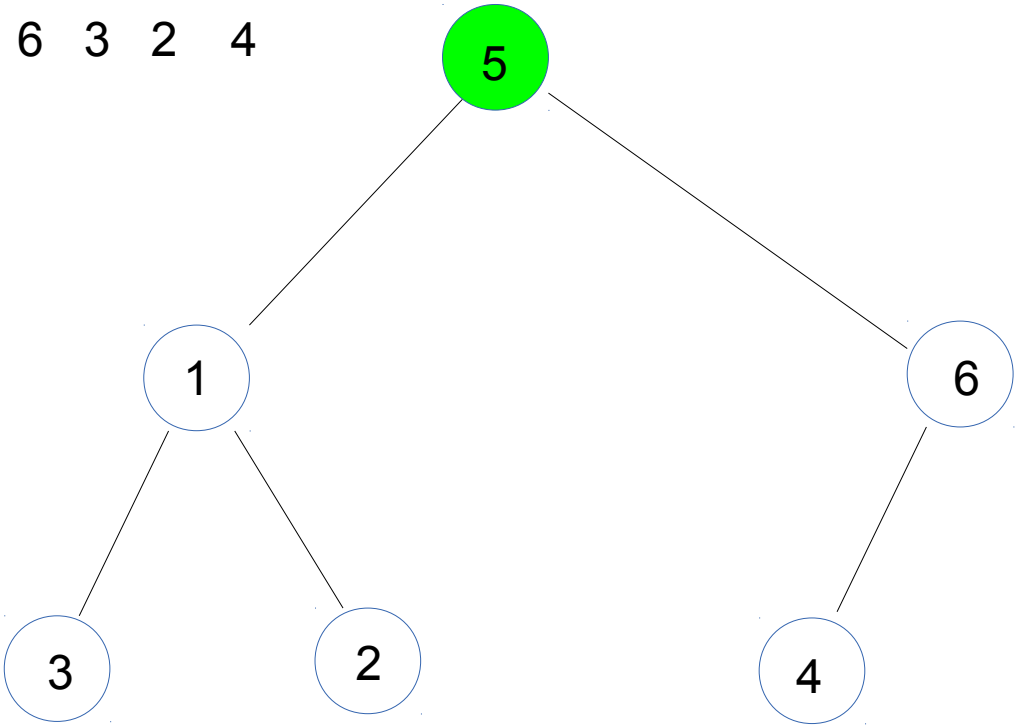
```
void heapSort(double[] a) {
    for (int i = 1; i < a.length; i++)
        swim(a, i);
    for (int i = a.length - 1; i > 0; i--) {
        swap(a, 0, i);
        sink(a, i);
    }
}
```

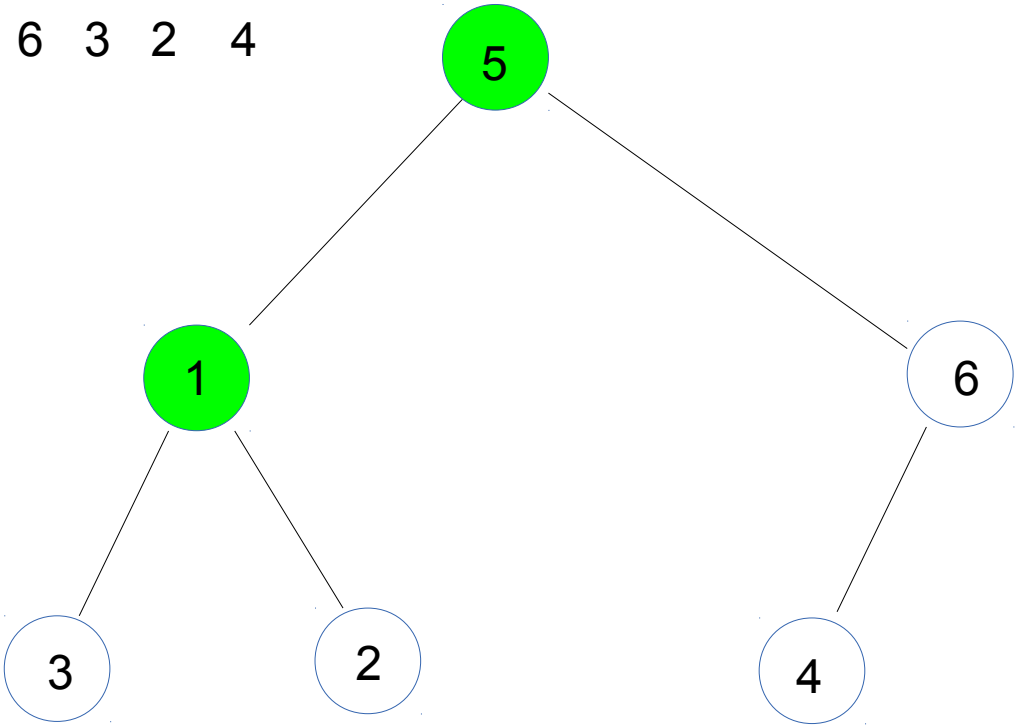
5 1 6 3 2 4



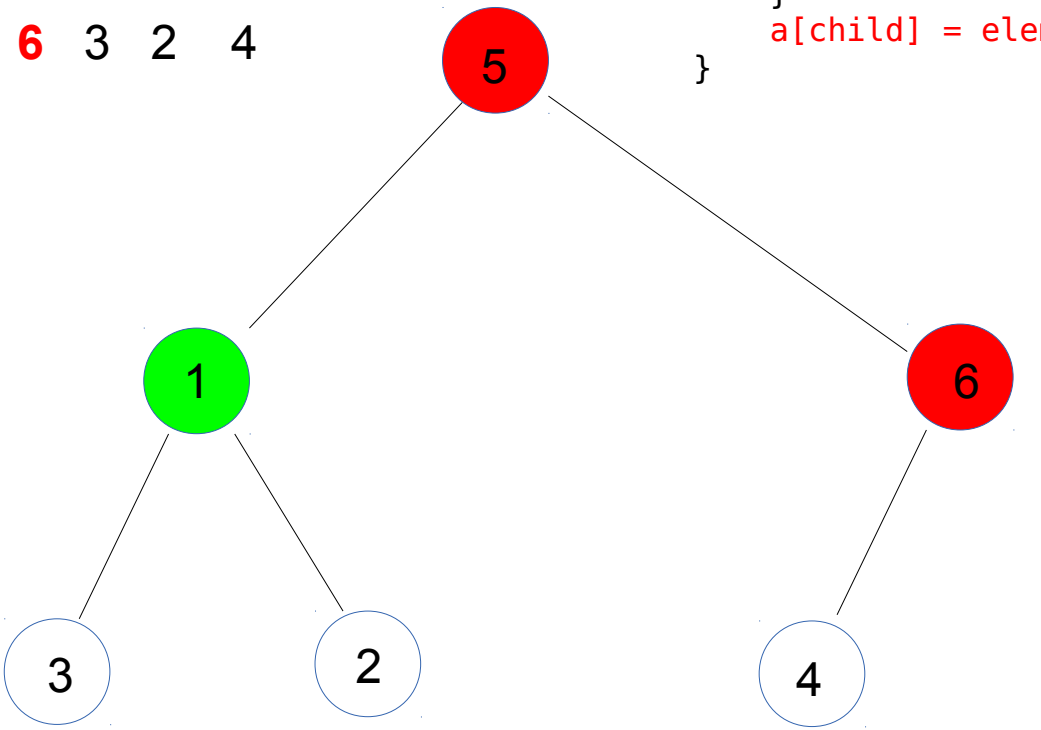
5 1 6 3 2 4



5 **1** 6 3 2 4

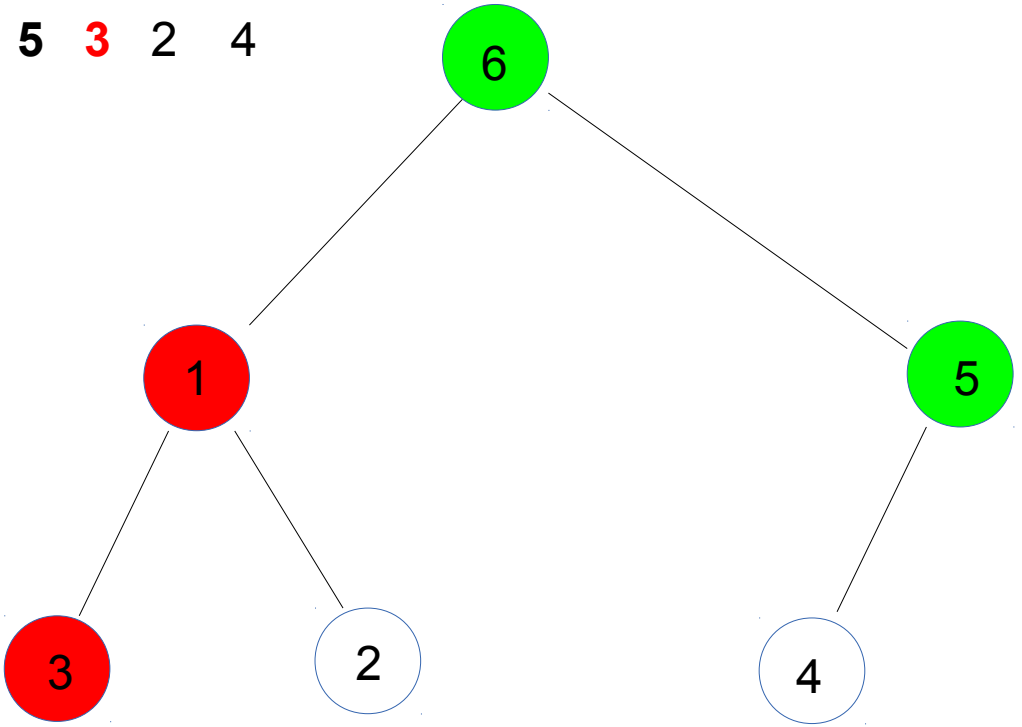


5 1 6 3 2 4



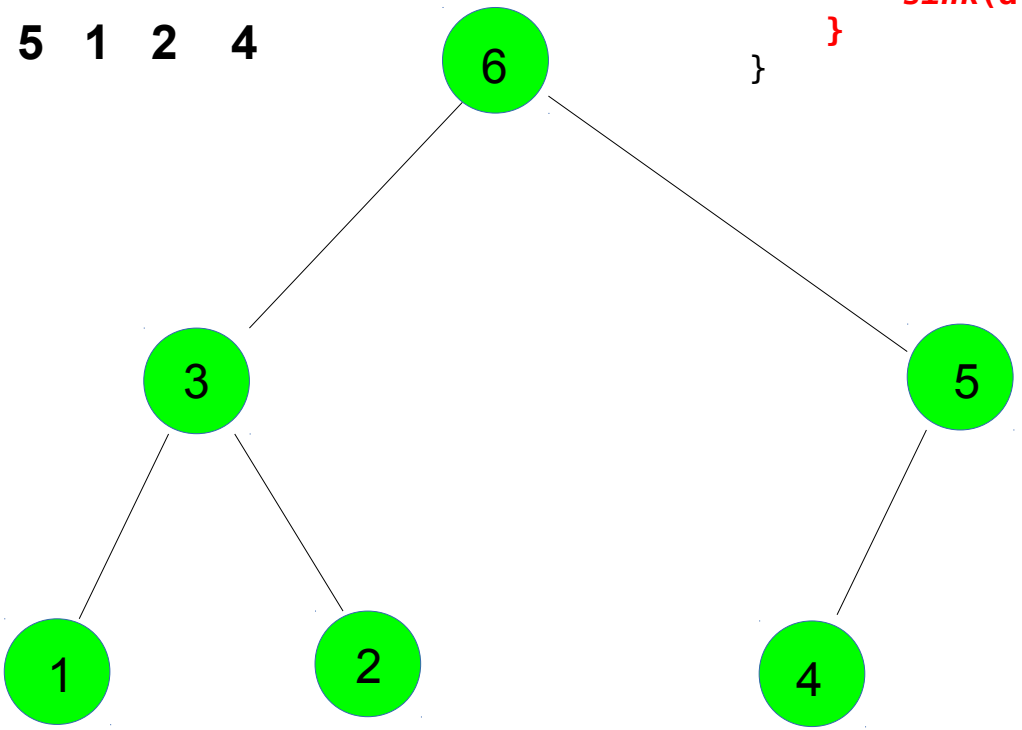

```
void swim(double[] a, int child) {
    int parent = parentOf(child);
    double element = a[child];
    while (child != ROOT && a[parent] < element) {
        a[child] = a[parent];
        child = parent; parent = parentOf(paremt);
    }
    a[child] = element;
}
```

6 **1** **5** **3** 2 4

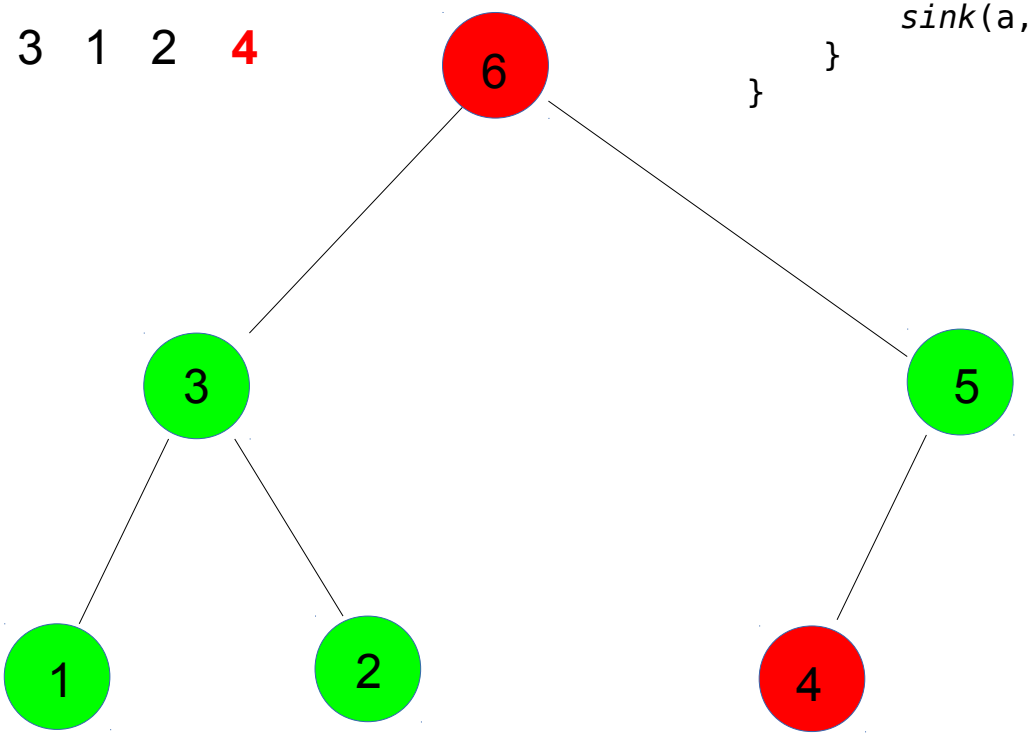


```
void heapSort(double[] a) {
    for (int i = 1; i < a.length; i++)
        swim(a, i);
    for (int i = a.length - 1; i > 0; i--) {
        swap(a, 0, i);
        sink(a, i);
    }
}
```

6 3 5 1 2 4

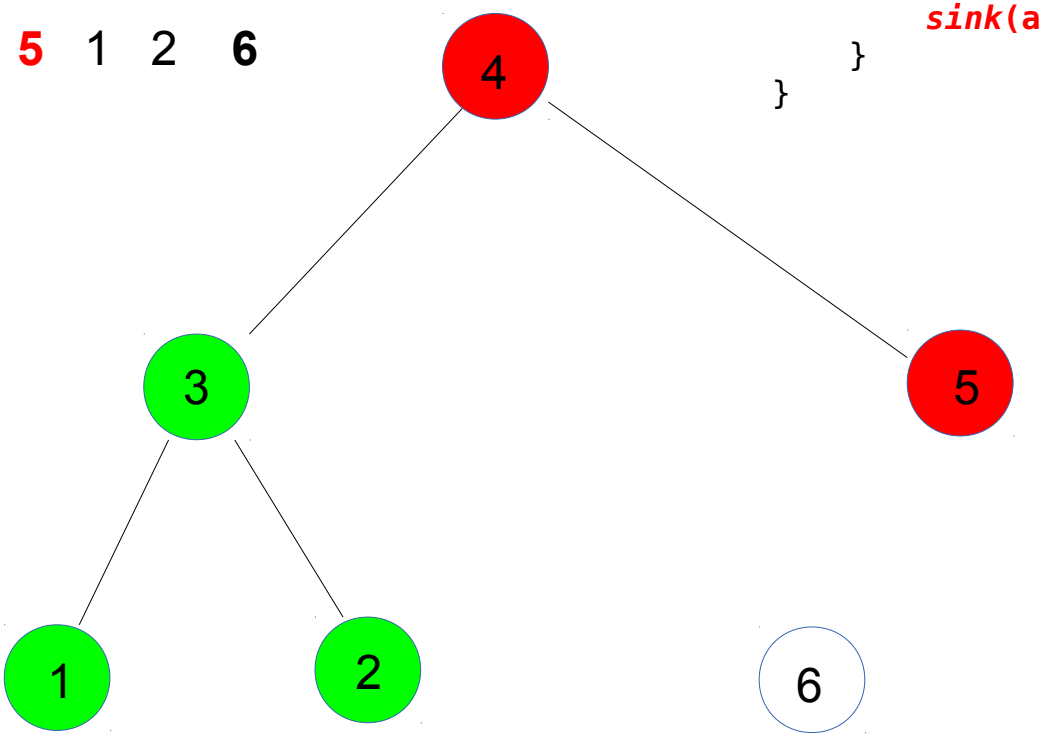


6 5 3 1 2 **4**



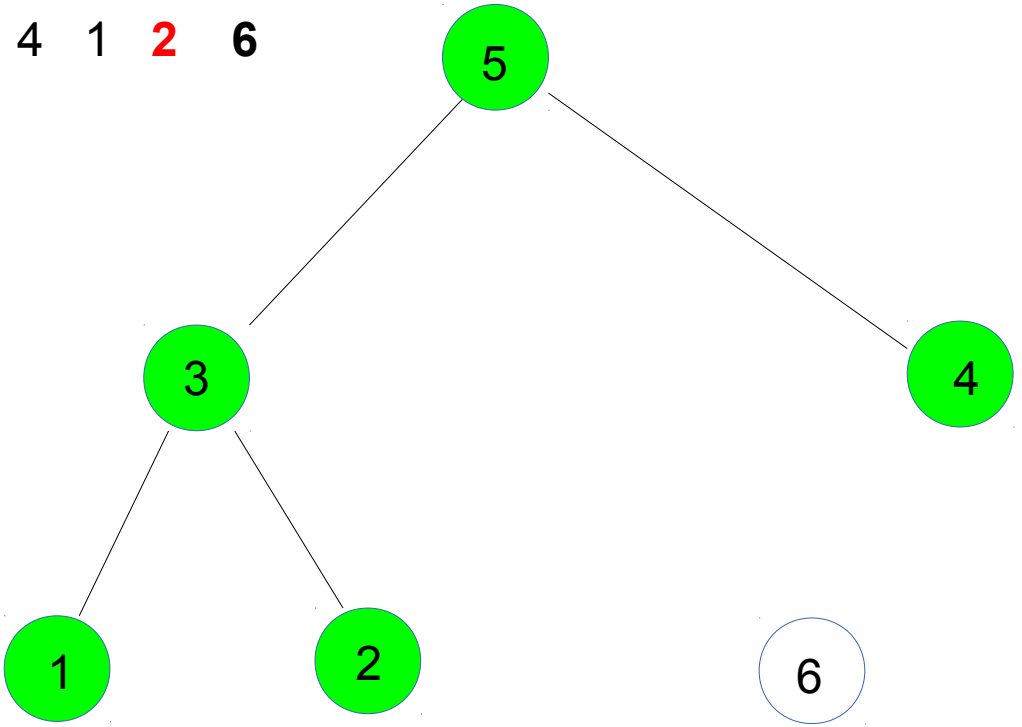

```
void heapSort(double[] a) {
    for (int i = 1; i < a.length; i++)
        swim(a, i);
    for (int i = a.length - 1; i > 0; i--) {
        swap(a, 0, i);
        sink(a, i);
    }
}
```

4 3 5 1 2 6

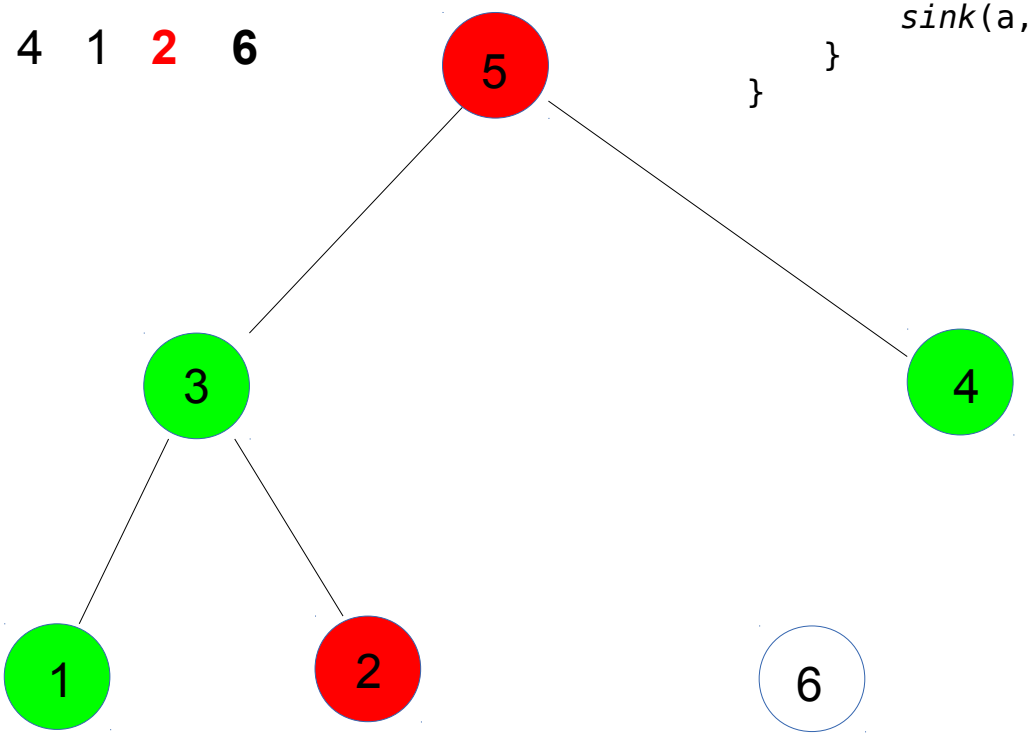



```
void heapSort(double[] a) {
    for (int i = 1; i < a.length; i++)
        swim(a, i);
    for (int i = a.length - 1; i > 0; i--) {
        swap(a, 0, i);
        sink(a, i);
    }
}
```

5 3 4 1 **2** **6**

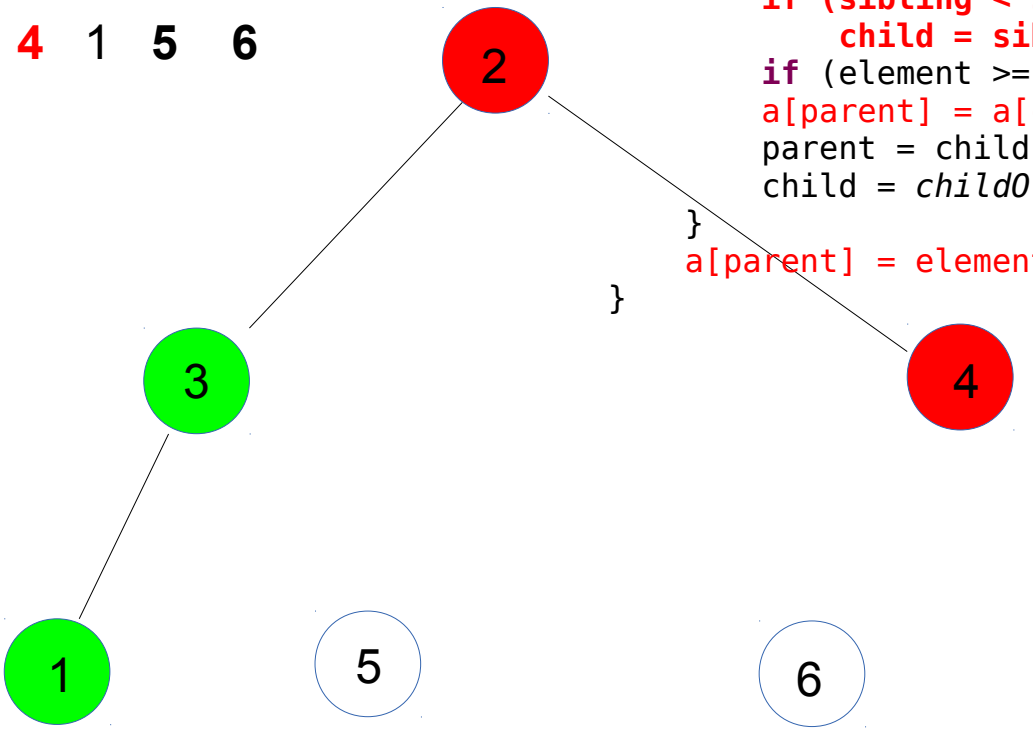


5 3 4 1 **2** **6**



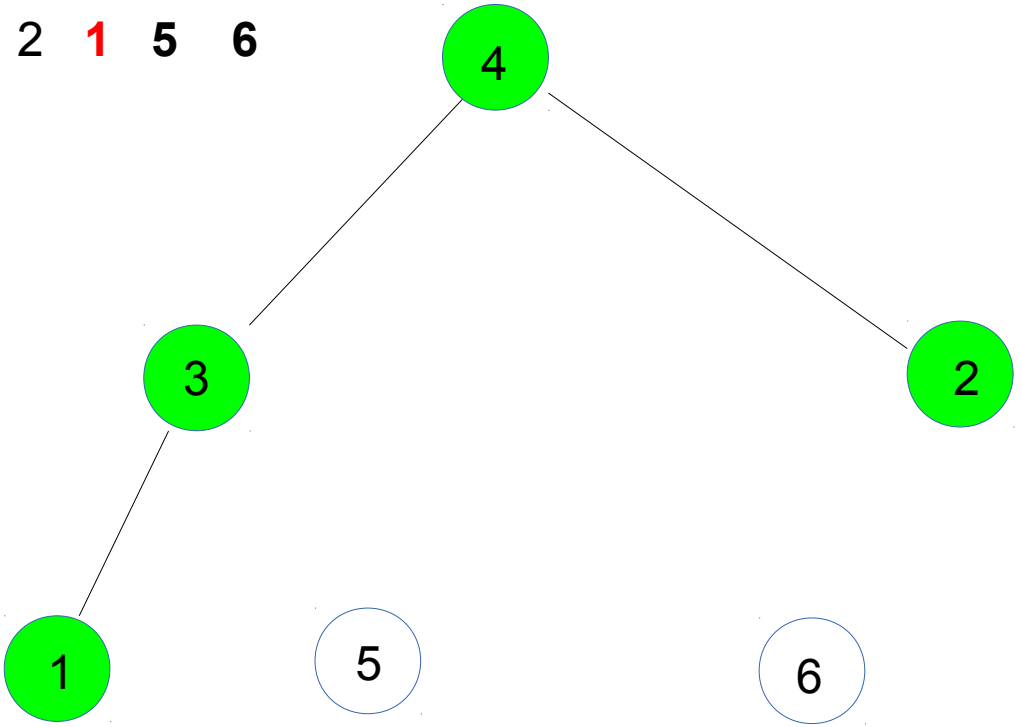

```
heapSort(double[] a) {
    for (int i = 1; i < a.length; i++)
        swim(a, i);
    for (int i = a.length - 1; i > 0; i--) {
        swap(a, 0, i);
        sink(a, i);
    }
}
```

2 3 **4** 1 **5** **6**

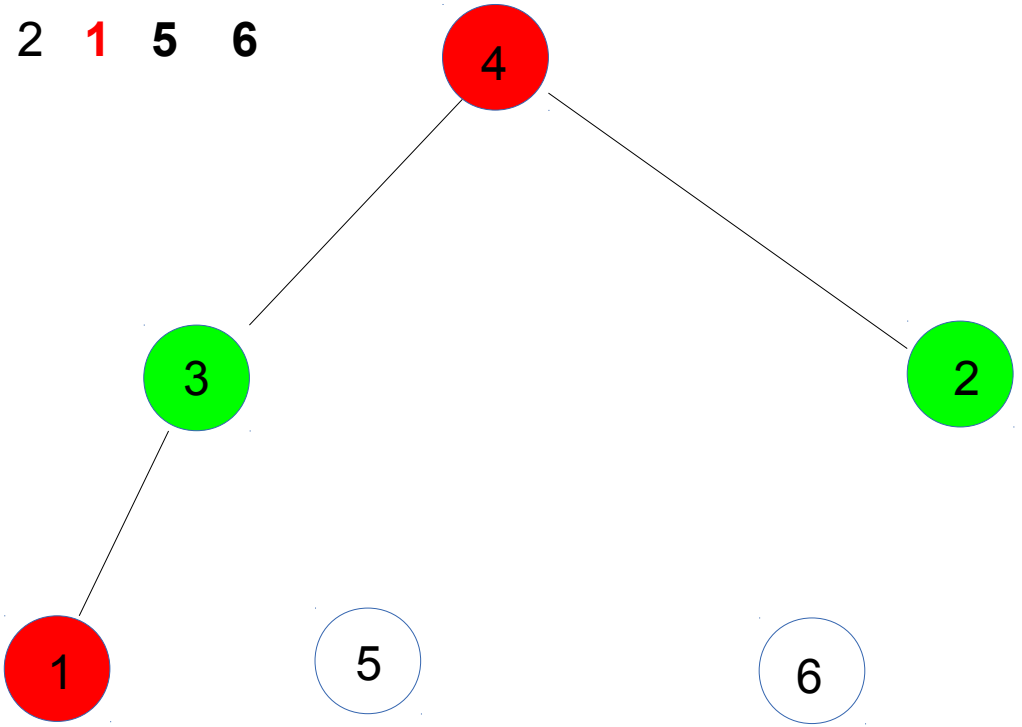



```
void sink(double[] a, int size) {
    int parent = ROOT;
    int child = childOf(ROOT);
    double element = a[ROOT];
    while (child < size) {
        int sibling = child + 1;
        if (sibling < size && a[sibling] > a[child])
            child = sibling;
        if (element >= a[child]) break;
        a[parent] = a[child];
        parent = child;
        child = childOf(child);
    }
    a[parent] = element;
}
```

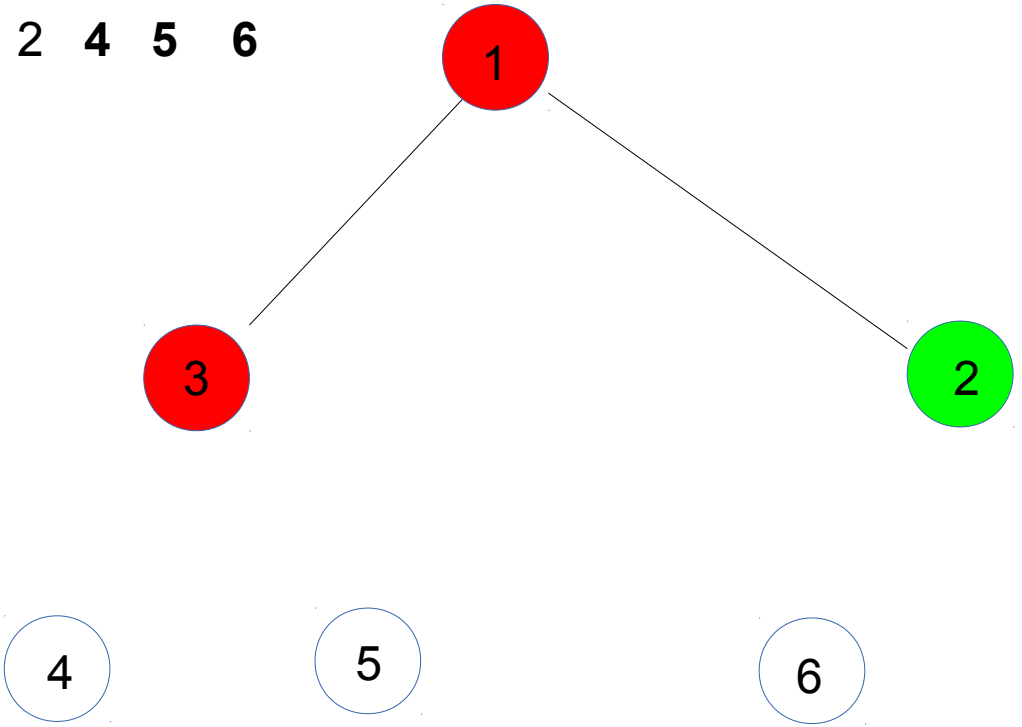
4 3 2 **1** **5** **6**



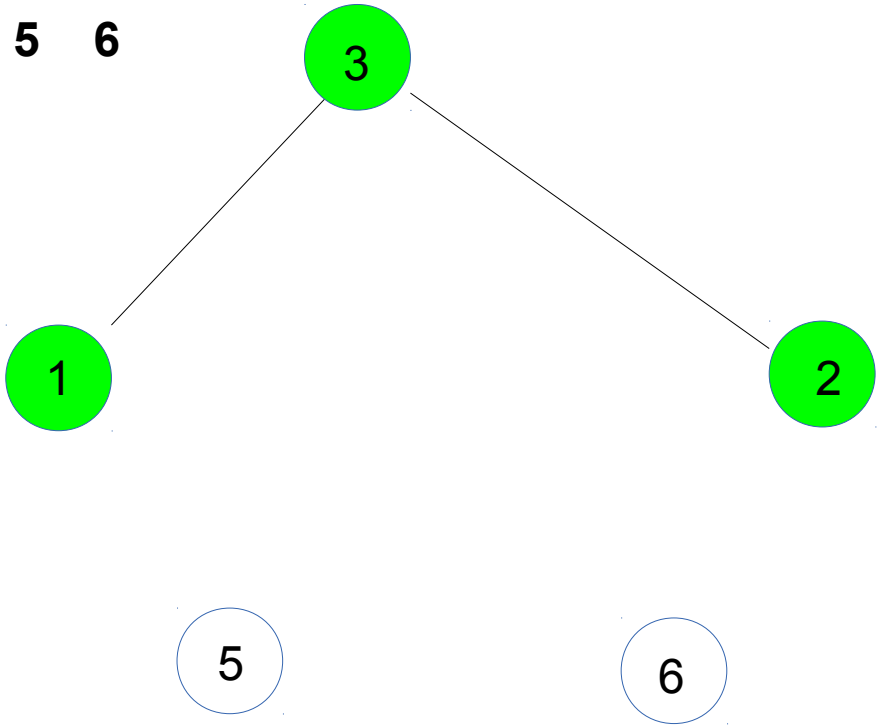
4 3 2 **1** **5** **6**



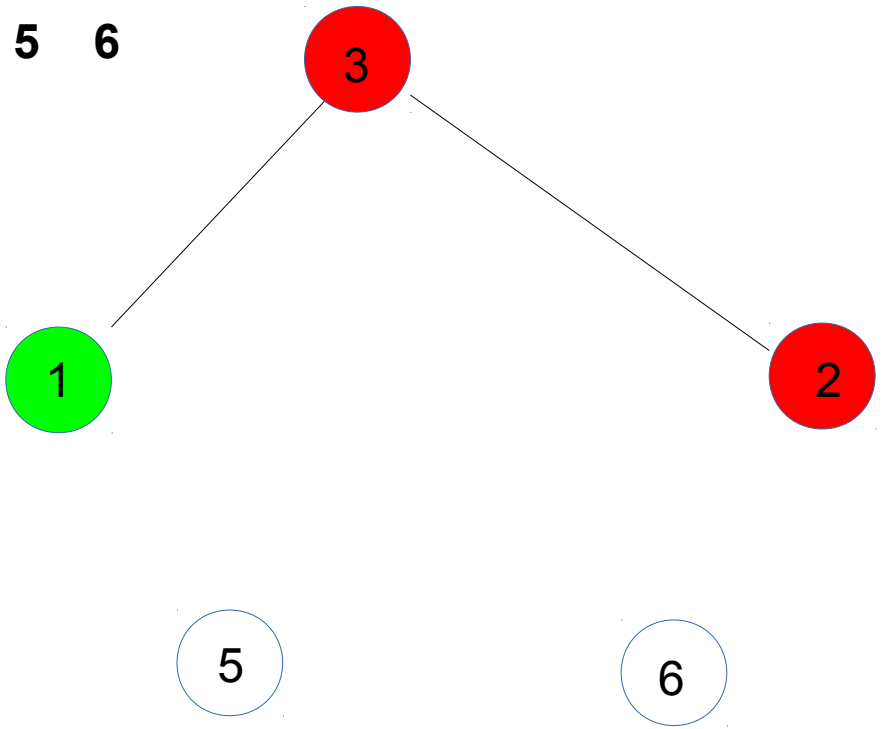
1 3 2 **4 5 6**



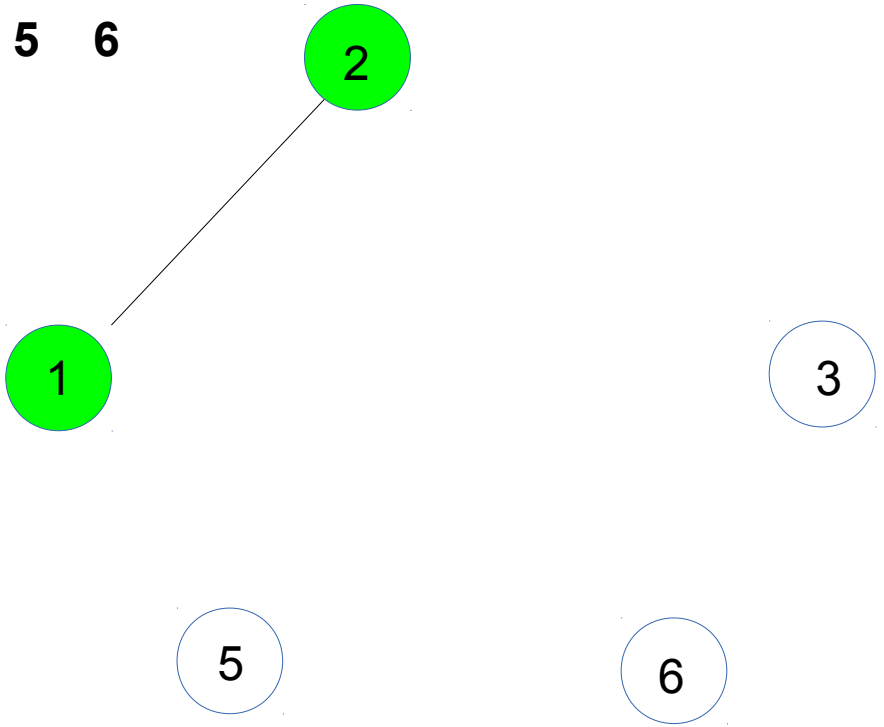
3 1 2 **4 5 6**



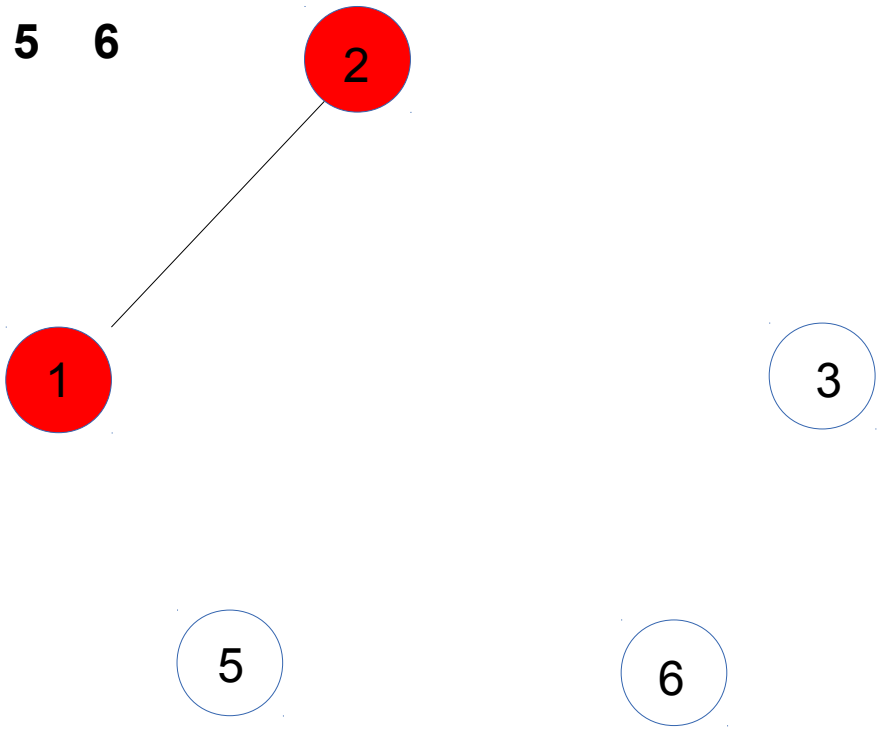
3 1 2 **4 5 6**



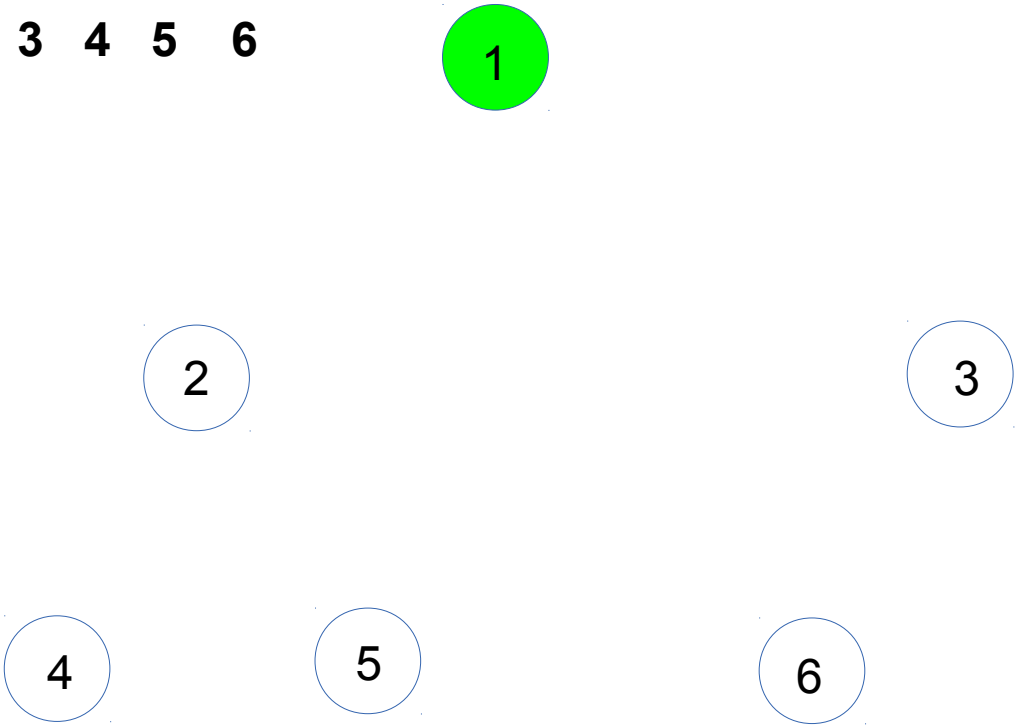
2 1 3 4 5 6



2 1 3 4 5 6



1 **2 3 4 5 6**



1 2 3 4 5 6

