

---

---

---

---

---

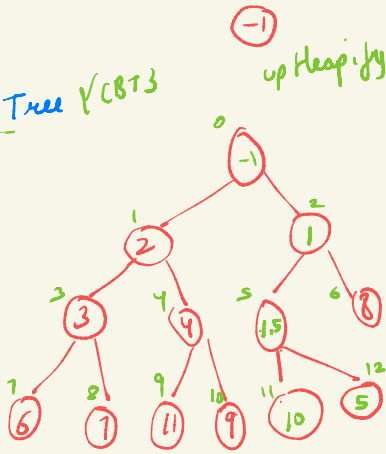


# # Heap {min p2}

Complete Binary Tree (CBT)

HoP  
Heap order property

why?  
what?  
how?



→ every parent will be smaller than child

{ -1, 2, 1, 3, 4, 15, 8, 6, 7, 11, 9, 10, 12, 5 }

$$pi = 4$$

$$lci = 2 * pi + 1;$$

$$rci = 2 * pi + 2;$$

$$lci = 9$$

$$rci = 10$$

$$pi = \frac{12-1}{2} \Rightarrow \underline{\underline{5}}$$

$$pi = \frac{5-1}{2} \Rightarrow \underline{\underline{2}}$$

swap (ci, pi)

(ci)

$$\{ pi = \frac{ci-1}{2} \}$$

```
static class Heap {
    ArrayList<Integer> data;

    public Heap(){
        data = new ArrayList<>();
    }

    public void swap(int i, int j){
        int valAtI = data.get(i);
        int valAtJ = data.get(j);

        data.set(i, valAtJ);
        data.set(j, valAtI);
    }

    public void upHeapify(int ci){
        int pi = (ci-1)/2;

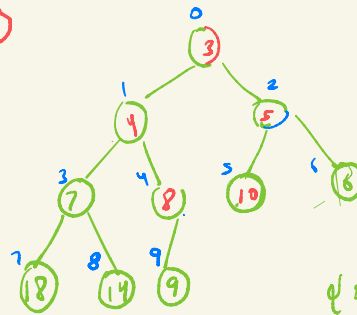
        if(data.get(pi) > data.get(ci)){
            swap(ci, pi);
            upHeapify(pi);
        }
    }

    // add
    public void add(int val){
        data.add(val);

        upHeapify(data.size()-1);
    }
}
```

data → { 3, 4, 5, 7, 8, 10, 6, 18, 14, 9 }

(-1)



downHeapify

{ 0, 13 }

$$pi = 6$$

$$lci = 3$$

$$rci = 4$$

$$pi = 0$$

$$lci = 2 * pi + 1 = 1$$

$$rci = 2 * pi + 2 = 2$$

{ 1, 4 }

```

private void downHeapify(int pi){
    int lci = 2*pi + 1;
    int rci = 2*pi + 2;

    int mini = pi;

    if(lci < data.size() && data.get(lci) < data.get(mini)){
        mini = lci;
    }

    if(rci < data.size() && data.get(rci) < data.get(mini)){
        mini = rci;
    }

    if(mini != pi){
        swap(pi, mini);
        downHeapify(mini);
    }
}

// remove
public int remove() {
    if(data.size() == 0){
        System.out.println("There is no element in heap");
        return -1;
    }

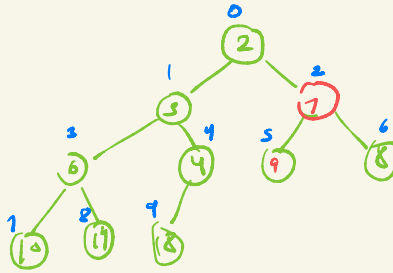
    swap(0, data.size()-1);
    int rv = data.remove(data.size()-1);

    downHeapify(0);

    return rv;
}

```

<sup>0 1 2 3 4 5 6 7 8 9</sup>  
 { 2, 3, 7, 6, 4, 9, 8, 10, 17, 18 }



rv = -1

mini = ~~5~~ 8

pi = ~~0~~ 3  
 lci = ~~1~~ 7  
 rci = ~~2~~ 8