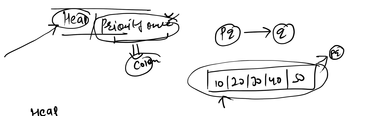
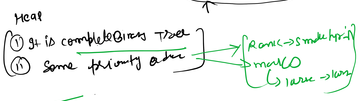
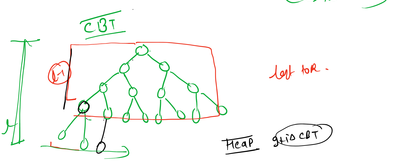
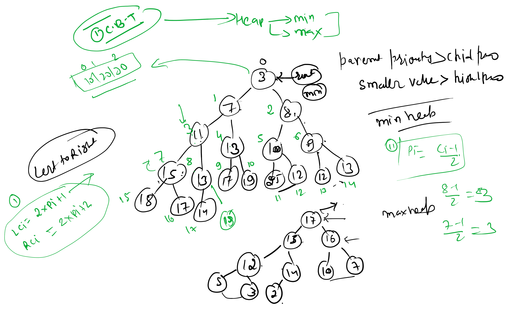
Lec-36

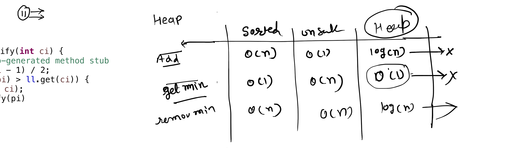
03/12/24 10:06 AM





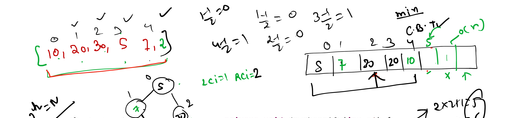


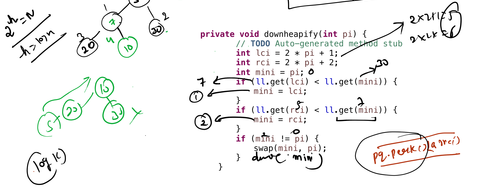




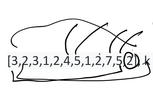

```
private void upheapify(int ci) {
        // TODO Auto-generated method stub
        int pi = (ci - 1) / 2;
        if (ll.get(pi) > ll.get(ci)) {
            swap(pi, ci);
            upheapify(pi)

        }
    }
```



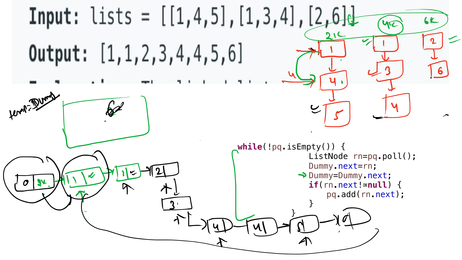



```
private void downheapify(int pi) {
        // TODO Auto-generated method stub
        int lci = 2 * pi + 1;
        int rci = 2 * pi + 2;
        int mini = pi;
        if (ll.get(lci) < ll.get(mini)) {
            mini = lci;
        }
        if (ll.get(rci) < ll.get(mini)) {
            mini = rci;
        }
        if (mini != pi) {
            swap(mini, pi);
            downheapify(mini);
        }
    }
```

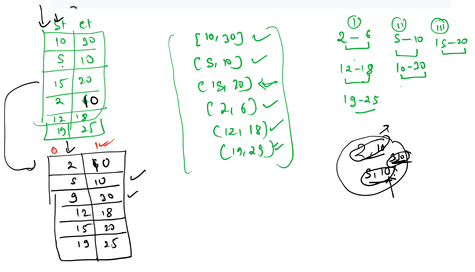


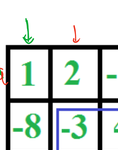
Input: lists = [[1,4,5],[1,3,4],[2,6]]

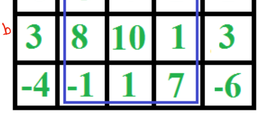
Output: [1,1,2,3,4,4,5,6]

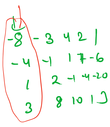


intervals = [[0,30],[5,10],[15,20]]










```
public static int Mini_Room(int[][] arr) {
        Arrays.sort(arr, (a, b) -> a[0] - b[0]);
    }
```

```
public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[][] arr = { { 9, 30 }, { 2, 10 }, { 15, 20 }, { 5, 12 }, { 12, 18 }, { 19, 25 } };

    }

    public static int Mini_Room(int[][] arr) {
        Arrays.sort(arr, (a, b) -> a[0] - b[0]);
        PriorityQueue<int[]> pq = new PriorityQueue<>((a, b) -> a[1] - b[1]);
        int room = 1;
        pq.add(arr[0]);
        for (int i = 1; i < arr.length; i++) {
            if (arr[i][0] < pq.peek()[1]) {
                pq.add(arr[i]);
                room = Math.max(room, pq.size());
            } else {
                pq.poll();
                pq.add(arr[i]);
            }
        }
```