CSE12 - Lecture 21 - B00

PA7 due tomorrow
PA4 Late/Resubmit due Tomorrow
Friday → Exam 2
 ↳ from Exam 1 up to hash tables

Heap Applications

Median

```
class Tracker {
    PriorityQueue<Integer> pq1 = new PriorityQueue<>(Collections.reverseOrder(Integer::compare));
    PriorityQueue<Integer> pq2 = new PriorityQueue<>(Integer::compare);
    void add(int n) {
        if(pq2.size() == 0 && pq1.size() == 0) {
            pq2.add(n);
            return;
        }
        int current = get();
        if(n >= current) {
            pq2.add(n);
        }
        else {
            pq1.add(n);
        }
        int sizeDifference = pq2.size() - pq1.size();
        if(sizeDifference > 1) { pq1.add(pq2.poll()); }
        else if(sizeDifference < -1) { pq2.add(pq1.poll()); }
    }

    int get() {
        if(pq2.size() == pq1.size()) { return (pq2.peek() + pq1.peek()) / 2; }
        if(pq2.size() > pq1.size()) { return pq2.peek(); }
        else { return pq1.peek(); }
    }

    public String toString() {
        return "" + pq1 + " " + this.get() + " " + pq2;
    }
}
```
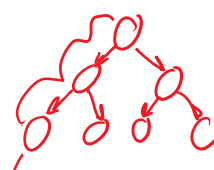
heap
high to low → max heap
low to high → min heap
sorted input

$m →$ # of elements

$\Theta(1)$

WC $\log_2(m)$ BC 1

$\Theta(1)$

$\frac{BC}{\log_2(m)}$ $\frac{WC}{2\log_2(m)}$
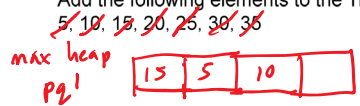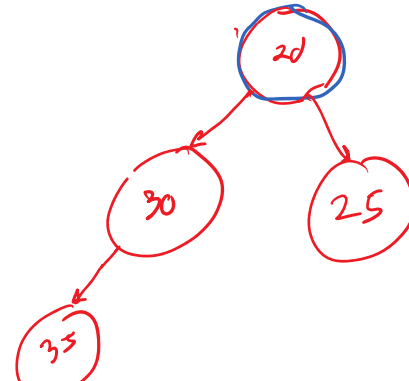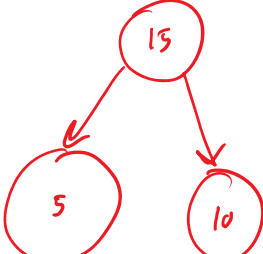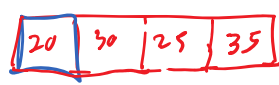
height
 ↳ $\log_2(m)$

add()
 ↳ $\log_2(m)$

poll()
 $\Theta(\log_2(m))$

Draw the picture and the arrays for the following:

Add the following elements to the Tracker (in this order):
5, 10, 15, 20, 25, 30, 35

max heap
pq1 | 15 | 5 | 10 | |

min heap
pq2 | 20 | 30 | 25 | 35 |

Value of get()? 20

What is the result of the call to get() after adding all the elements?

20

What is the run-time for the tracker?

add()   get()

Worst Case $\Theta(\log_2(N))$ $\Theta(1)$

Best Case:

Write a method to use the tracker:

```
int findNumber (Integer[] arr) {
    Median Tracker  tracker = new _____ |
    for (int i=0; i< arr.length; i++) {
        tracker.add(arr[i]);  } log_2(n)  N  ->  N * log_2(n)
    }
    return tracker.get();  |
}
```

What is the total run-time using the tracker:

$$\Theta(N * \log_2(n))$$

Using a PriorityQueue, write a Heap Sort method to perform an in-place sort of an array: