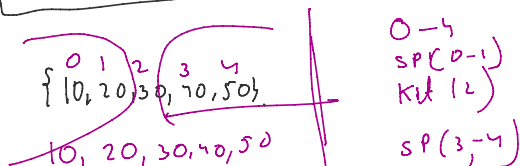
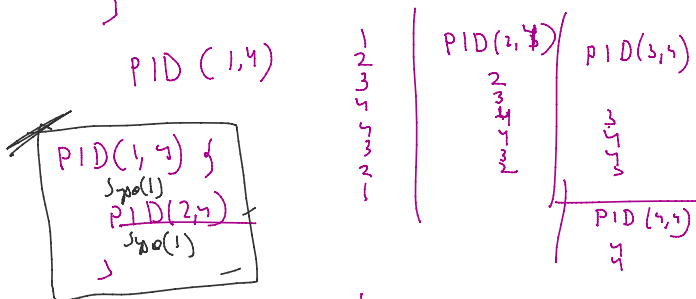
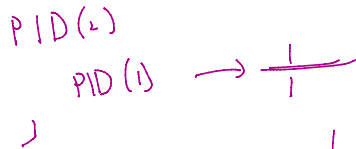
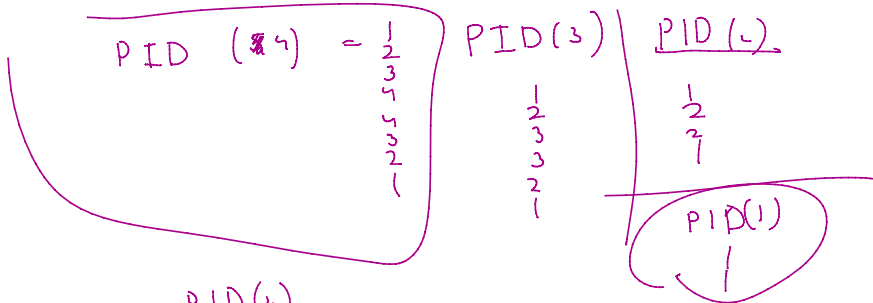
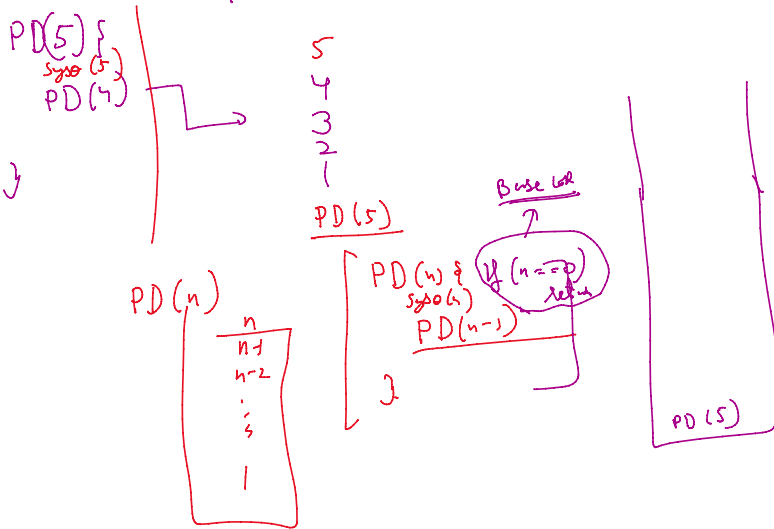
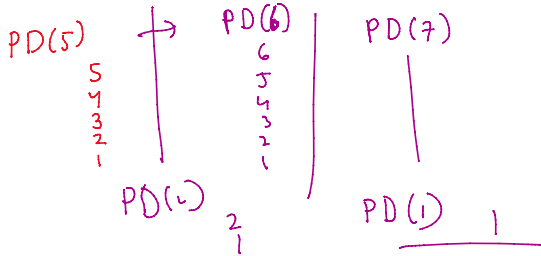


- ~~Identify~~ Identify S.P
- Identify S.P
- Assume S.P
- Bigger solⁿ using smaller ~~smaller~~



~~{10, 20, 30, 40, 50}~~ ~~Kit 12~~
~~10, 20, 30, 40, 50~~ ~~SP(3-4)~~

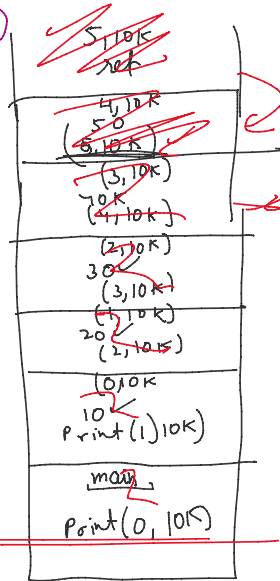
(s, e)
 rad 0-1
 (s, idx-1)
 idx = (s, e) ← arr[idx]
 (idx+1, e)

```
public static void main(String[] args) {
    int[] arr = { 10, 20, 30, 40, 50 };
    print(0, arr);
}

public static void print(int s, int[] arr) {

    System.out.println(arr[s]);
    print(s+1, arr);
}
}
```

10, 20, 30, 40, 50



```
public static void printInterval(int s, int e, int[] arr) {
    if(s > e) {
        return;
    }
    int mid = (s+e)/2;
    printInterval(s, mid-1, arr);
    System.out.println(arr[mid]);
    printInterval(mid+1, e, arr);
}
```

10, 20 / 30

main
 pI(0, 4)

s=0, e=4
 mid=2

(0, 1)
 → arr[2]
 (3, 4)

s=0, e=1

m=0

(0, -1)

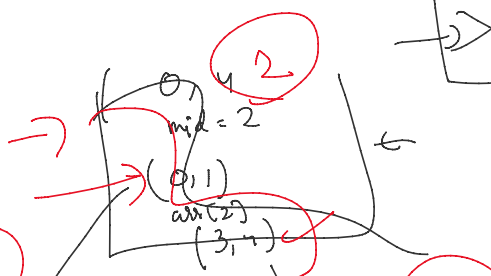
→ arr[0]

→ (1, 1)

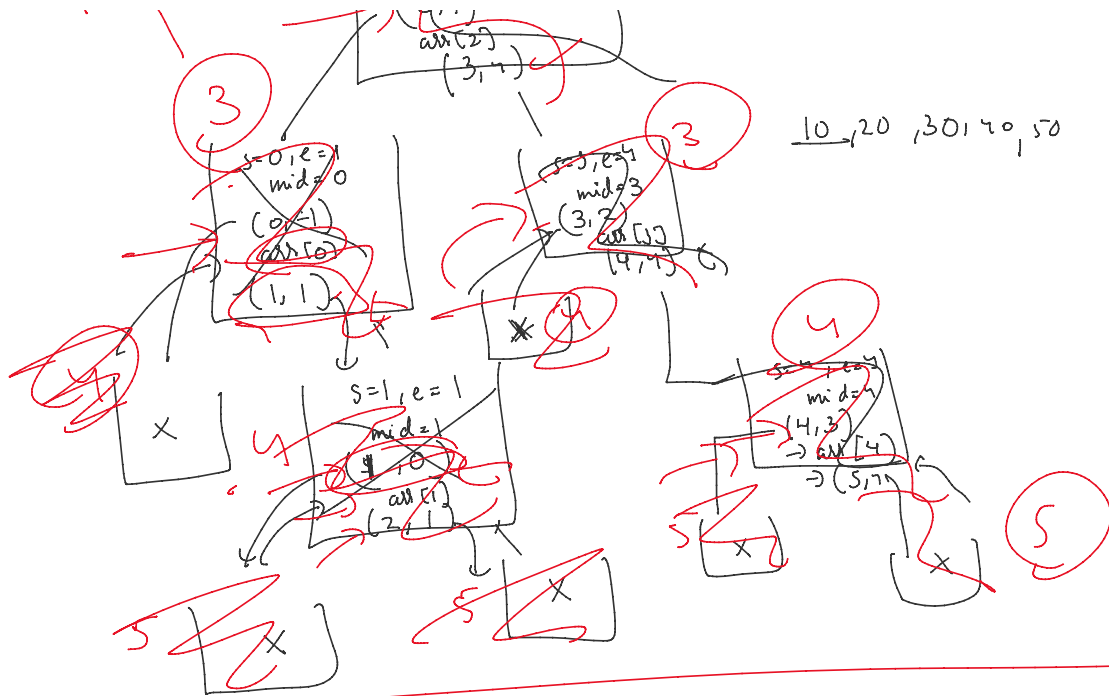
~~s=0, e=-1~~
~~s > e~~
~~t~~

s=1, e=1
 mi=1
 1, -1
 arr[1]
 (2, 1)

Main



[X] > [X]



{ 10, 20, 30, 40, 50 }

50, 40, 30, 20, 10

→ 40, 30, 20, 10

50, 40, 30, 20

{ 20, 1, 4, 9, 7 }

Max(0)

q = Max(1)

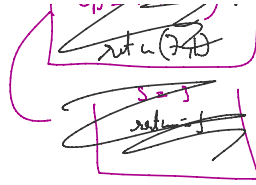
return (arr(5), 0)

```
public static int Max(int s, int []arr) {
    // {2,4,7,1}
    if(s==arr.length-1) {
        return arr[s];
    }
    int sp = Max(s+1, arr);
    return Math.max(arr[s], sp);
}
```

{ 20, 1, 4, 9, 7 }

Max(0, arr)
sp = Max(1)
arr(1,2)
s=1
sp = Max(2)
(4,3)
s=2
sp = Max(3)
return(7,0)
s=3

7



{ 5, 6, 5, 6, 6, 7 }

0 1 2 3 4 5

First occ
ali

6

