

D.P. Data S2

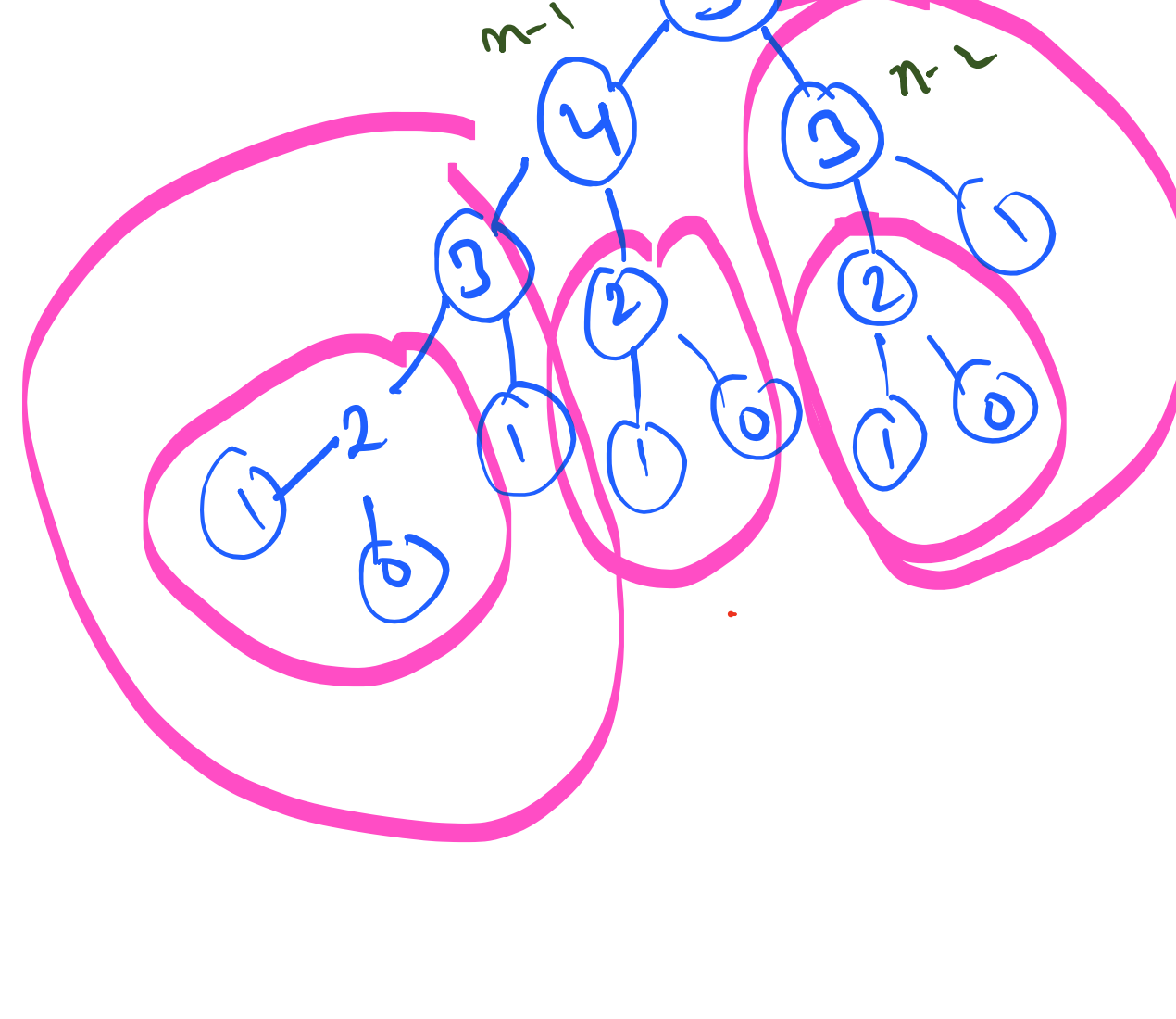
$2^n \rightarrow O(n)$

$10! = 1 \times 2 \times 3 \times \dots \times 10$

$12! = 1 \times 2 \times 3 \times \dots \times 12$

$= 10! \times 11 \times 12$

#



$[i] \rightarrow$ Problem-overlapping

$\times [i] \rightarrow$ optimal sub S2

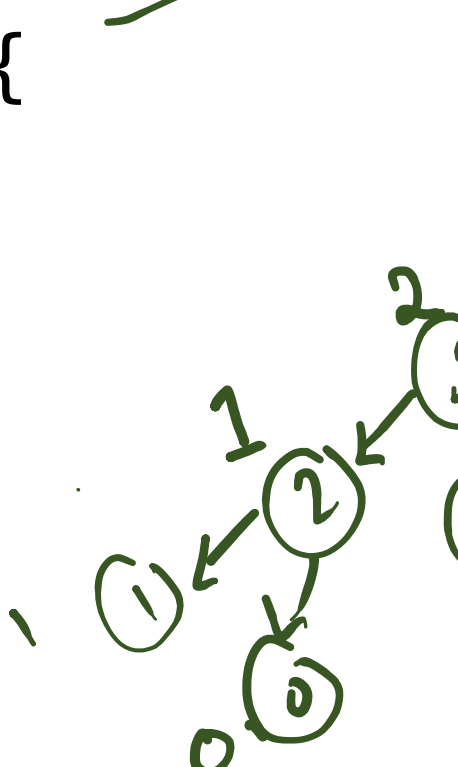


① Top-down \rightarrow Recursive \rightarrow memoization

② Bottom-up \rightarrow Iterative \rightarrow tabulation

(NM)

```
public static int fib(int n) {
    if (n == 0 || n == 1) {
        return n;
    }
    int f1 = fib(n-1);
    int f2 = fib(n-2);
    return f1 + f2;
}
```



$2 \rightarrow 1$

$3 \rightarrow 2$

$4 \rightarrow 3$

$5 \rightarrow 5$

```
public static int fibTD(int n, int [] dp) {
    if (n == 0 || n == 1) {
        return n;
    }
    if (dp[n] != 0) {
        return dp[n];
    }
    int f1 = fibTD(n-1, dp);
    int f2 = fibTD(n-2, dp);
    return dp[n] = f1 + f2;
}
```



0 1 2 3 4 5

```
public static int fibTD(int n, int [] dp) {
    if (n == 0 || n == 1) {
        return n;
    }
    if (dp[n] != 0) {
        return dp[n];
    }
    int f1 = fibTD(n-1, dp);
    int f2 = fibTD(n-2, dp);
    return dp[n] = f1 + f2;
}
```

0 1 2 3 4 5

$i=2 \quad i < n$

$dp[i] = dp[i-1] + dp[i-2]$

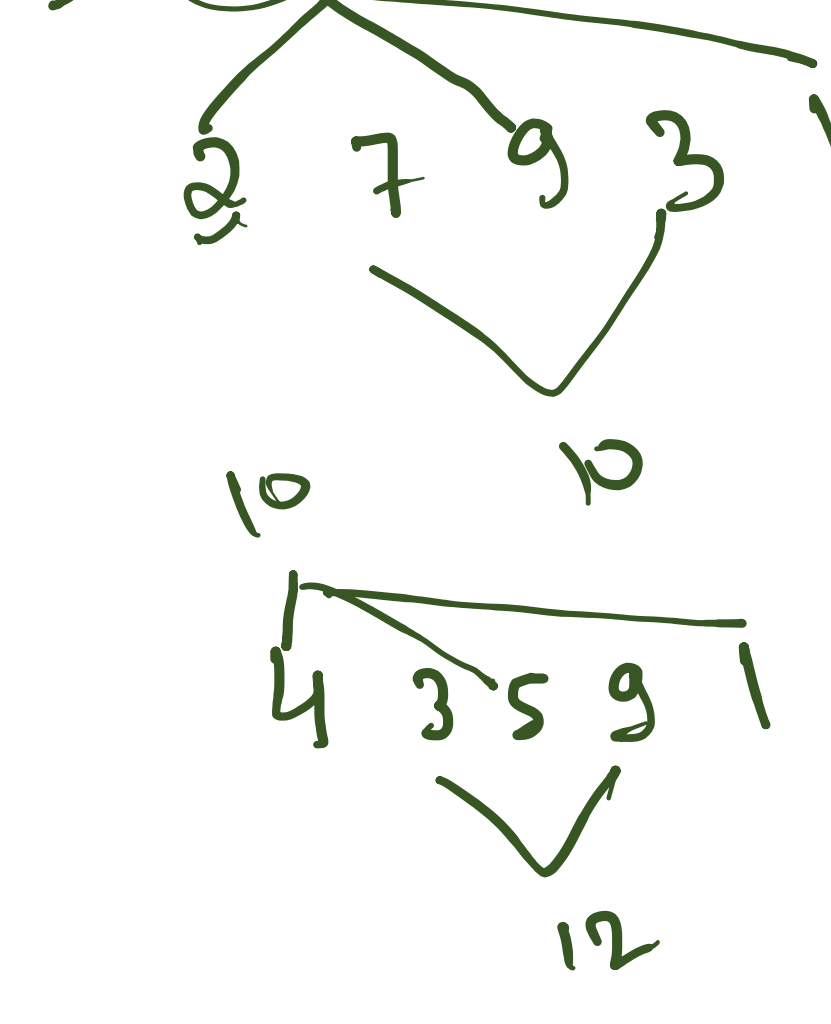
$i=2 \quad dp[2] = dp[1] + dp[0]$

$i=3 \quad dp[3] = dp[2] + dp[1]$

$i=4 \quad dp[4] = dp[3] + dp[2]$

$i=5 \quad dp[5] = dp[4] + dp[3]$

1, 2, 3, 1



12

7 9 3 1

10 10

4 3 5 9 1

12

0 1 2 3 4

4 3 5 9 1

$dp[i] = dp[i-1] + dp[i-2]$

$dp[2] = dp[1] + dp[0]$

$dp[3] = dp[2] + dp[1]$

$dp[4] = dp[3] + dp[2]$

$dp[5] = dp[4] + dp[3]$

```
public static int Robber(int[] arr, int i, int[] dp) {
    if (i >= arr.length) {
        return 0;
    }
    if (dp[i] != -1) {
        return dp[i];
    }
    int rob = arr[i] + Robber(arr, i+2, dp);
    int dont_rob = Robber(arr, i+1, dp);
    return dp[i] = Math.max(rob, dont_rob);
}
```

13 12 9 9 1

0 1 2 3 4

5 9 1 \rightarrow 9

$dp[1] = 9$

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

6 9

5 1 1

1 0 0

9 1 1

4 3 5 9 1

12 12

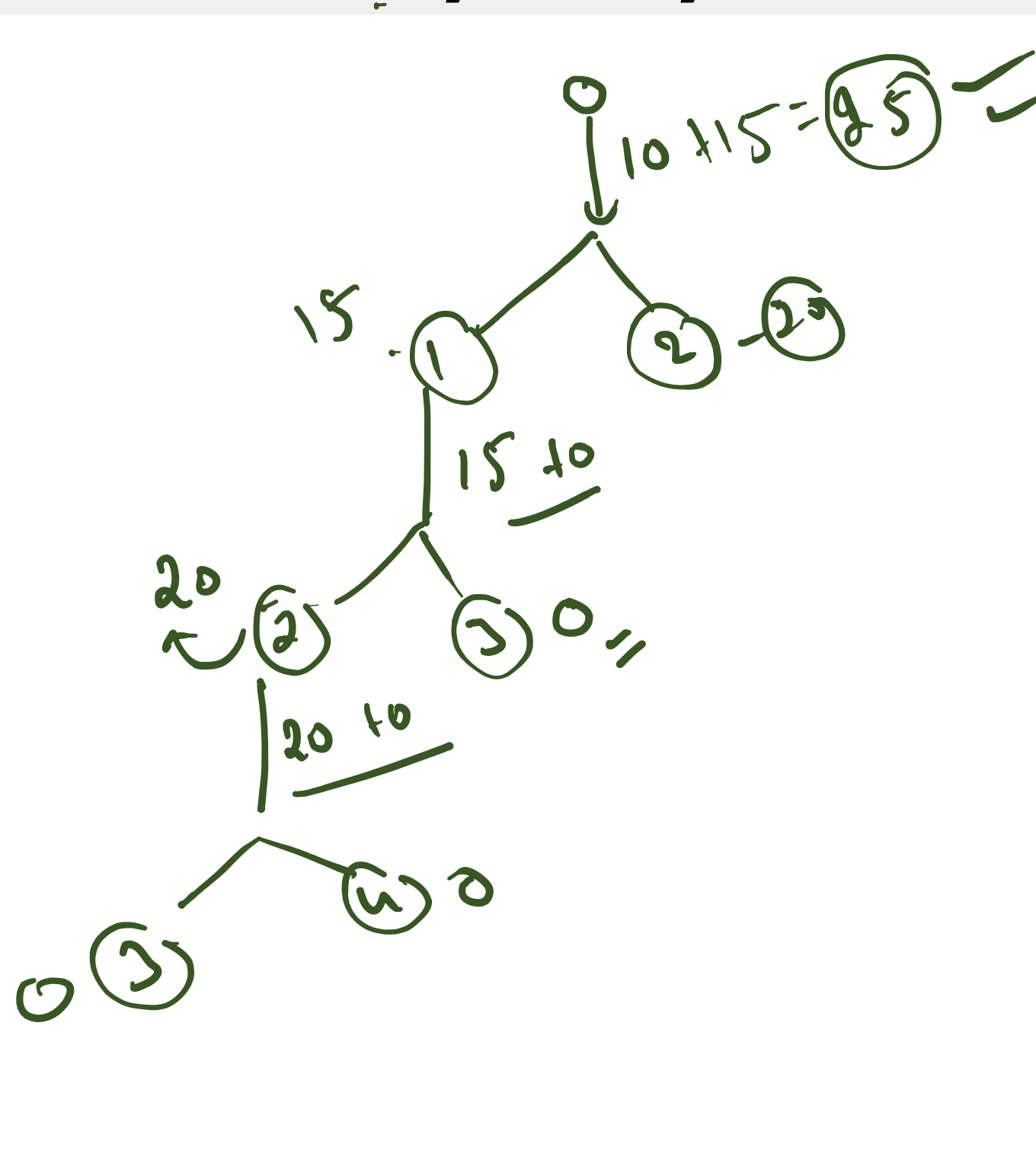
6 9

5 1 1

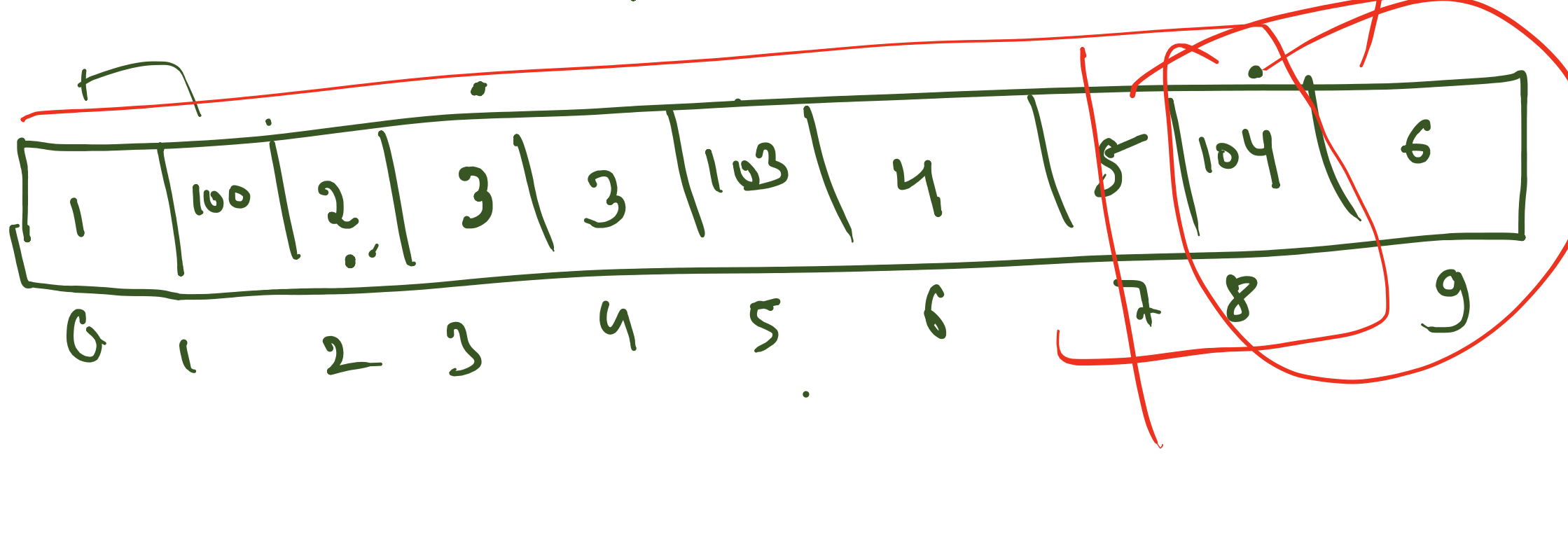
1 0 0

9 1 1

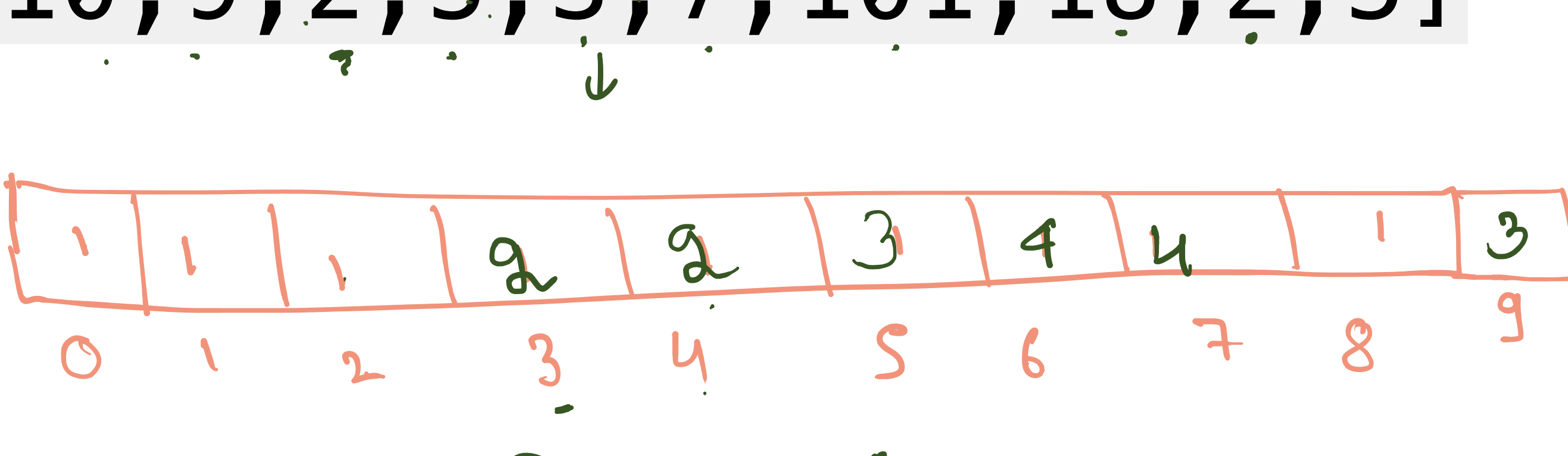
cost = [10, 15, 20]



Input: cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]



[10, 9, 2, 5, 3, 7, 101, 18, 2, 5]



2 > 1

2 3

2 5

2 7

2 3 7

2 5 7

2 3 8

2 5 7 8

2 5 101

2 3 9

For $i=1 \quad i < n \quad i++2$

For $j=i-1 \quad i > 0 \quad j--2$

If $arr[i] > arr[j]$

$x = arr[j]$

$arr[j] = \max(arr[j], x)$

100 = 248/9

int[] arr = { 0, 8, 4, 2, 12, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15 }; =

0: 0 [0]

8: 0 [0, 8]

4: 0 [0, 4]

2: 0 [0, 2]

11: 0 [0, 11]

10: 0 [0, 10]

6: 0 [0, 6]

14: 0 [0, 14]

1: 0 [0, 1]

9: 0 [0, 9]

5: 0 [0, 5]

13: 0 [0, 13]

3: 0 [0, 3]

11: 0 [0, 11]

7: 0 [0, 7]

15: 0 [0, 15]