

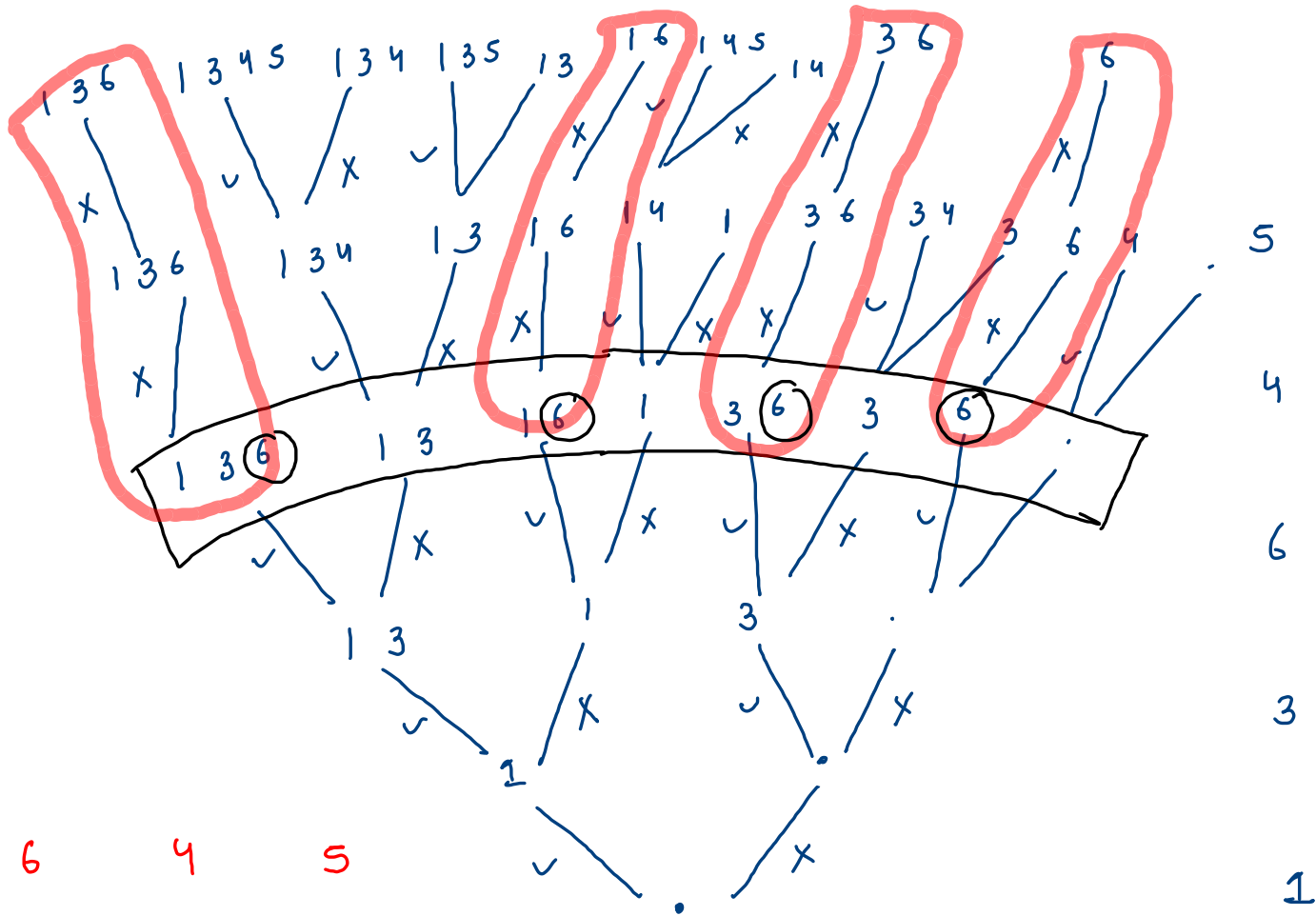
## Longest Increasing Subsequence

10    2 2    9    3 3    2 1    50    41    60    80    1

$2^{10}$  →  $x$  → select  
all subseq    increasing    the longest out of  $x$  subseq.

$2^n$   
↓  
increasing subseq

1 3 6 4 5



2d

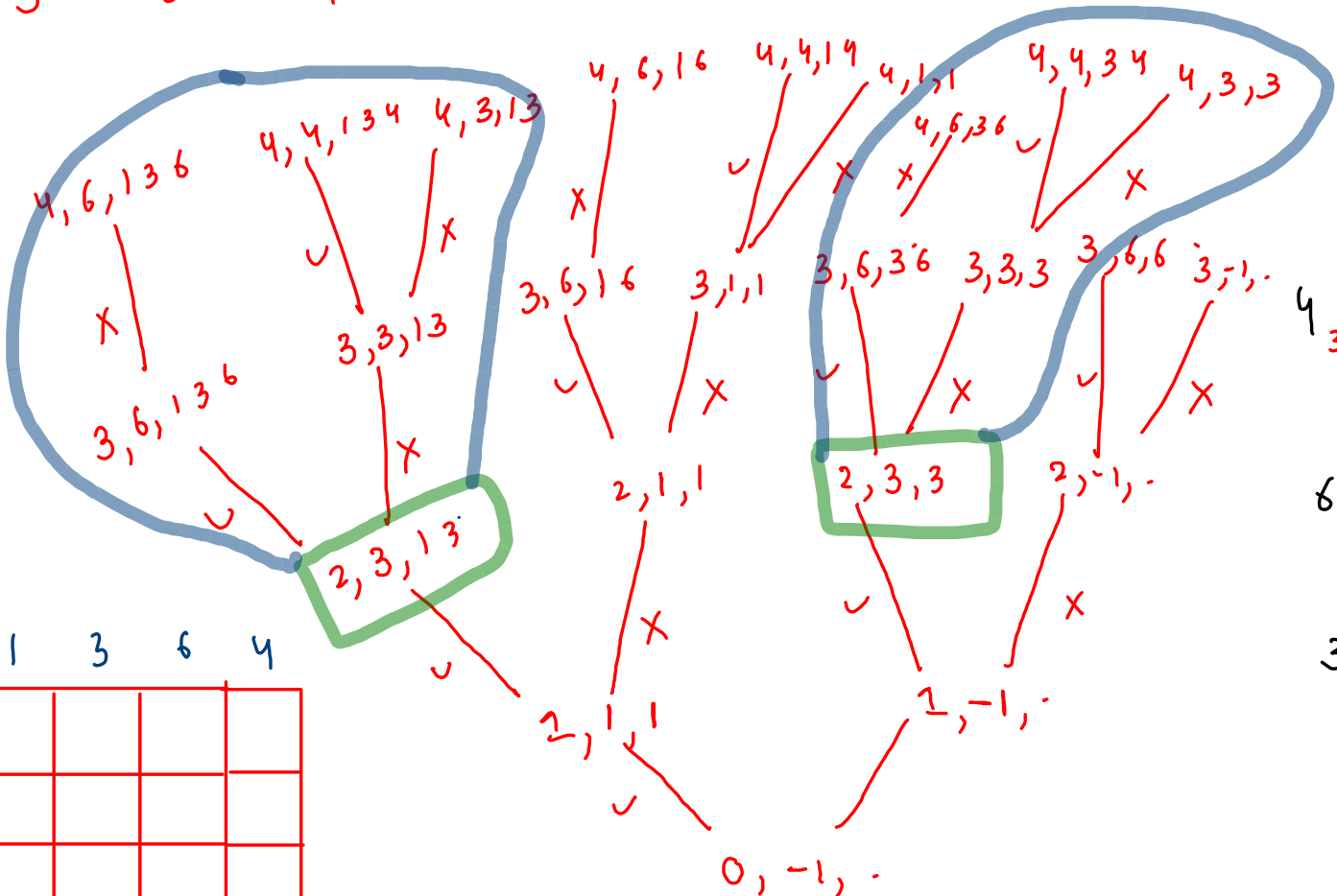
1 dim  $\rightarrow$  idx

2 dim  $\rightarrow$  last de

3

1

1 3 6 4



$n^2$

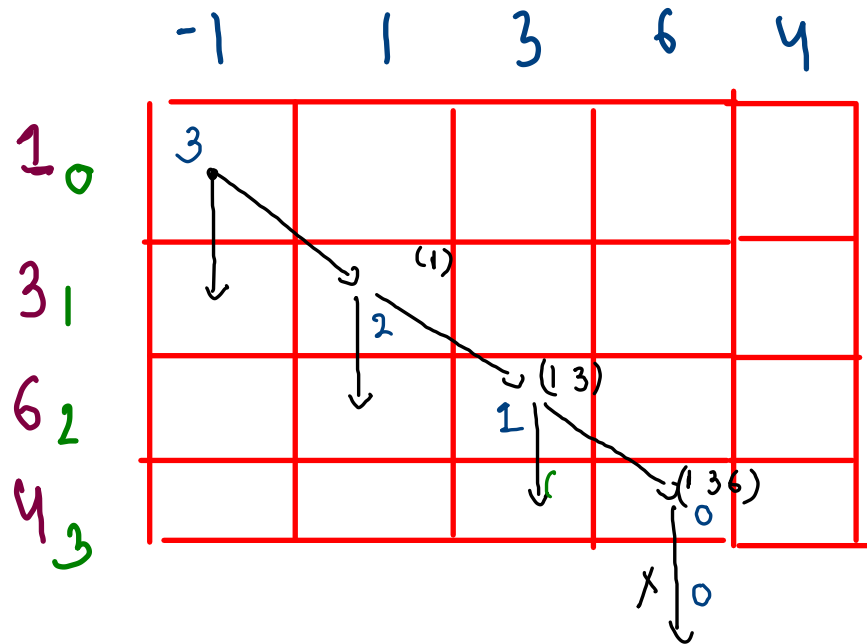
(i) dp

(ii) recursion

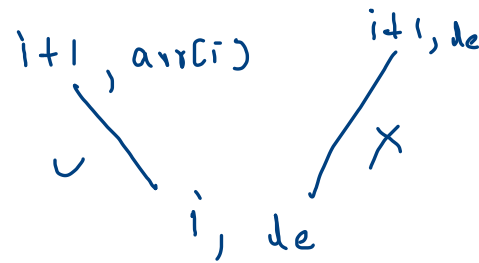
2, 3 → 1

ci, de, ss

	-1	1	3	6	4
1 <sub>0</sub>					
3 <sub>1</sub>					
6 <sub>2</sub>					
4 <sub>3</sub>					



ques bank



$$dp[i][de] = \max(dp[i+1][arr[i]] + 1, dp[i+1][de])$$

$10_0$	$2\ 2_1$	$9_2$	$3\ 3_3$	$2\ 1_4$	$50_5$	$41_6$	$60_7$	$80_8$	$1_9$
1	2	1	3	2	4	4	5	6	1
10	10 22	9	10 22 33	10 21	10 22 33 50	10 22 33 41	10 22 33 50 60	10 22 33 50 60 80	1

max den  $\rightarrow 6$

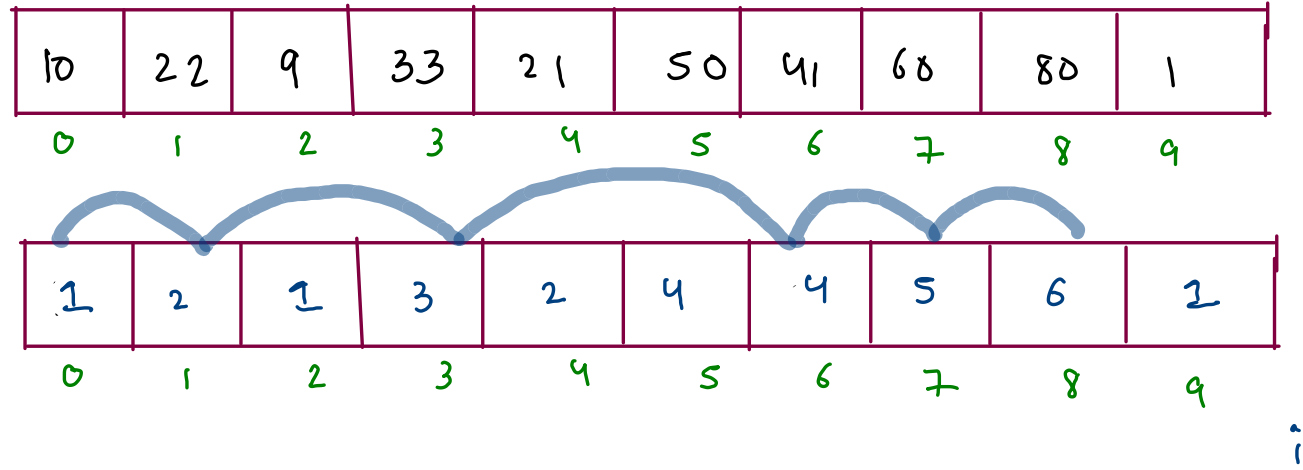
```

public static int LIS(int[] arr) {
    int[] dp = new int[arr.length];
    int omax = 0;

    for(int i=0; i < arr.length; i++) {
        int max = 0;
        for(int j=0; j < i; j++) {
            if(arr[j] < arr[i]) {
                max = Math.max(dp[j], max);
            }
        }
        dp[i] = max + 1;
        omax = Math.max(omax, dp[i]);
    }

    return omax;
}

```



$dp(i) \rightarrow$  0 to  $i$  elements  
 dis which ends  
 at  $i$ .

$omax = 1 \neq 2 \neq 3 \neq 4 \neq 5 \neq 6$

## Maximum Sum Increasing Subsequence

{1, 101, 2, 3, 100, 4, 5}

arr

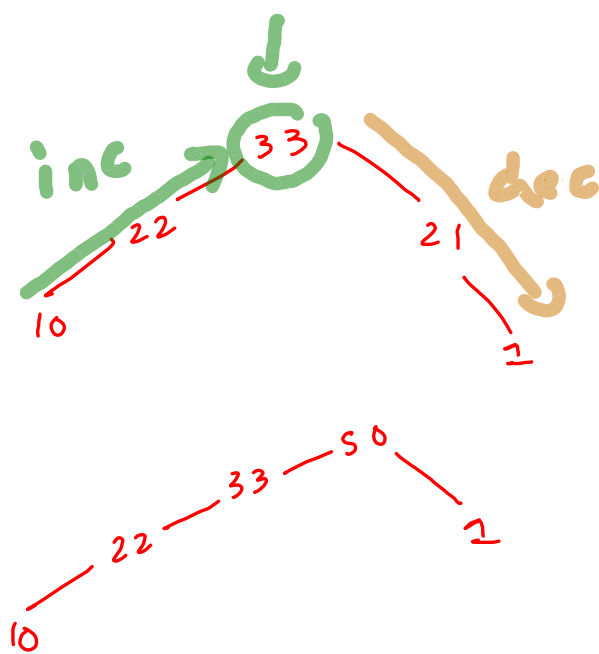
1	101	2	3	100	4	5
---	-----	---	---	-----	---	---

dp

1	101	3	6	106	10	15
---	-----	---	---	-----	----	----

Longest Bitonic Subsequence

10	22	9	33	21	50	41	60	80	1
0	1	2	3	4	5	6	7	8	9





10	22	9	33	21	50	41	60	80	1
0	1	2	3	4	5	6	7	8	9

lis

1	2	1	3	2	4	4	5	6	1
1	10	9	10	9	10	10	10	10	1
	22		22	21	22	22	22	22	
			33		33	33	33	33	
					50	41	41	41	
							60	60	
								80	



lds

3	3	2	3	2	3	2	2	2	1
10	22	9	33	21	50	41	60	80	1
9	9	1	21	1	41	1	1	1	
1	1		1		1				



$$lds : \quad 0 \max ( lis[i] + lds[i] - 1 )$$