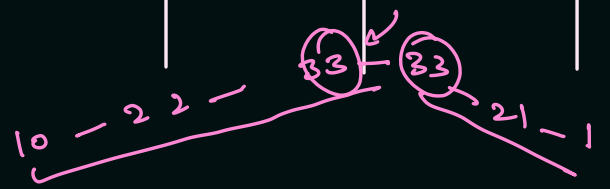
Longest Bitonic Subsequence

Bitonic Seq. → Increasing Decreasing.

	3	4	2	5	3	6	5	7	1
Bitonic length	1 9 10	1 21 22	1 8	1 21 33	1 21	1 41 50	1 41	1 60	1 80	1
	3	3	2	3	2	3	2	2	2	1 ← LDS
	10 0	2 2 1	9 2	3 3 3	21 4	50 5	41 6	60 7	80 8	1 9
LIS→	1	2	1	3	2	4	4	5	6	1
	10	10 22	9	10 22 33	9 21	10 22 33 50	10 22 33 41	10 22 33 41 60	10 22 33 41 60 80	1



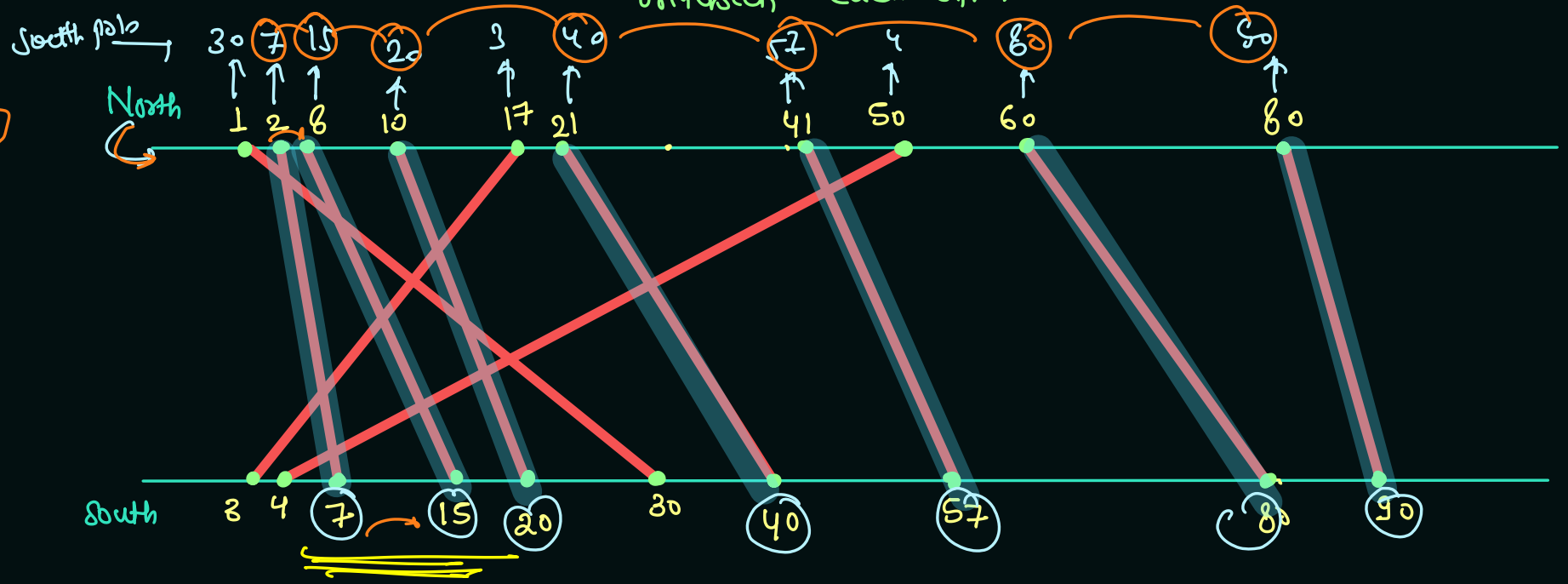
→ length of bitonic subseq

If peak is at i

LIS[i] + LDS[i] (-1) → peak up count peak twice

Maximum Non Overlapping bridges:→

Select maximum bridge such that no two selected bridge intersect each other.



N.	S.
10	20
2	7
8	15
17	3
21	40
50	4
41	57
60	80
80	90
1	30

Sequence of selected bridges.

↳ B1 B2 B3 - - - - .

Condition:

① B1.north < B2.north

② B1.south < B2.south

→ for no intersection.

Condition - ① → Sort array on the basis of noth pole.

N.	S.
10	20
2	7
8	15
17	3
21	40
50	4
41	57
60	80
80	90
1	30

sort on → basis of North pole.

B1.north < B2.north

1 - 30
2 - 7
8 - 15
10 - 20
17 - 3
21 - 40
41 - 57
50 - 4
60 - 80
80 - 90

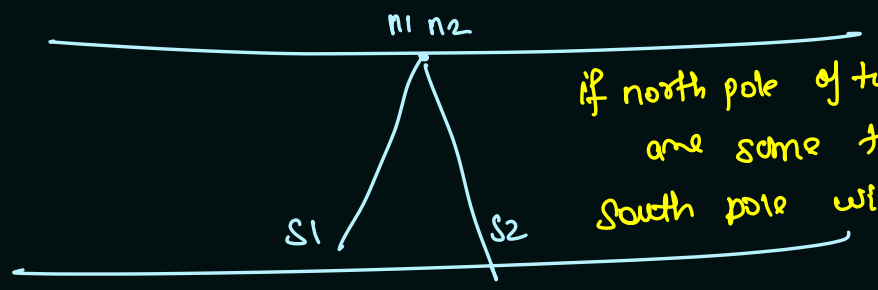
After sorting → B1.north < B2.north.

Guarantee → sorting

B1.south B2.south

B1.south < B2.south → sequence.

Longest increasing subseq. on South pole.



if north pole of two bridges are same then min. South pole will encounter for

Southpole → 30 7 15 20 3 40 57 4 80 90

1 1 2 3 1 4 5 2 6 7

30 7 7 15 7 15 20 3 7 15 20 40 57 80 3 4 7 15 20 40 57 80

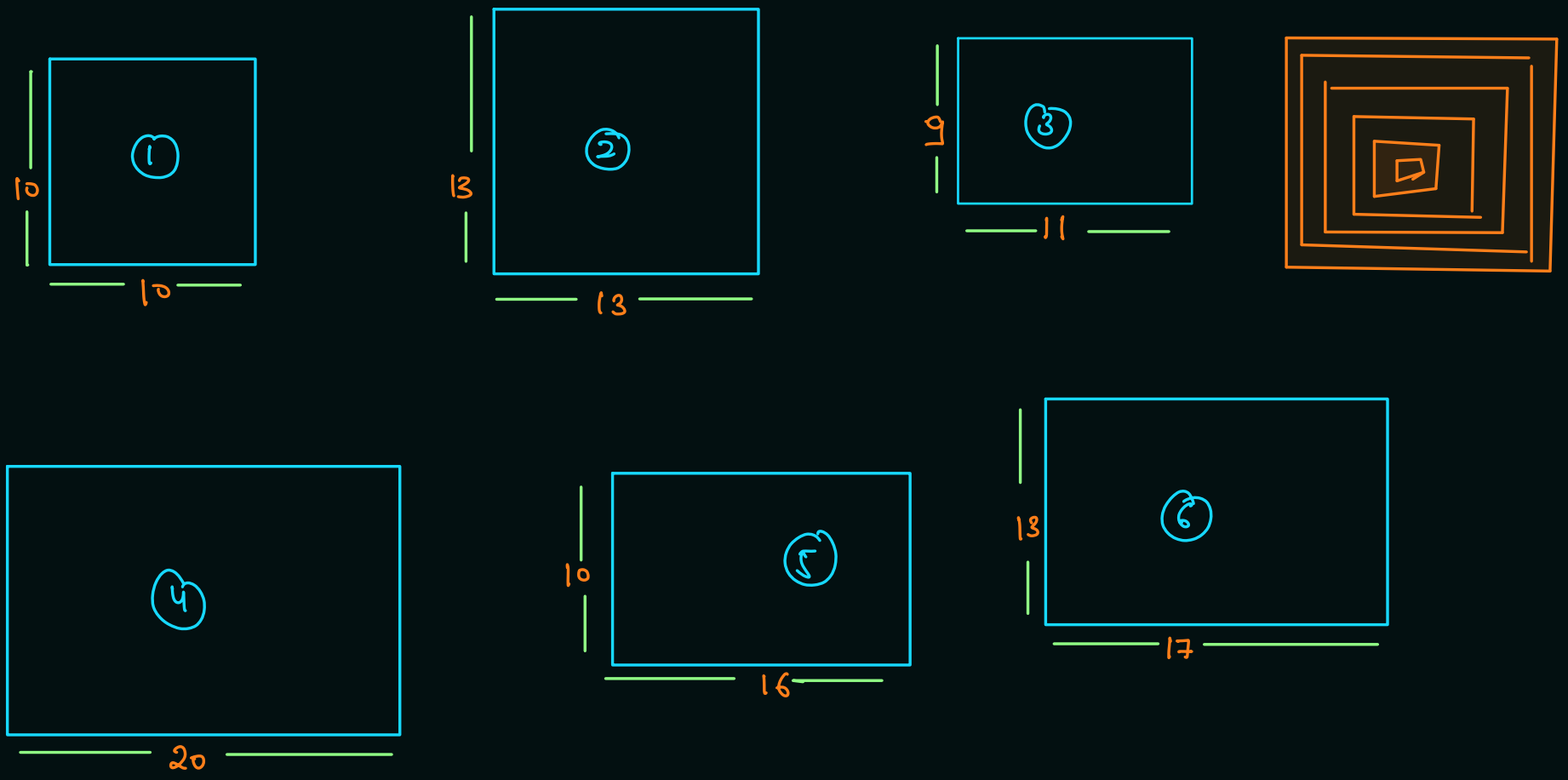
Same. 90

Russian Doll Envelope :-

width height

Rotations are Not allowed, width and height is not interchangable.

Envelopes = [w, h] → [10, 10], [13][13] [9, 11] [15, 20], [10, 16] [13, 17]



Envelopes = [w, h] → [10, 10], [13][13] [9, 11] [15, 20], [10, 16] [13, 17]

Condition for Nesting of Envelopes :-

E_1 E_2.

$E_1.h < E_2.h$. ⟶ Parameter ①

$E_1.w < E_2.w$ ⟶ " ②

Freeze Parameter ① ⟶ Sort on the basis of height]
Find LIS on width ⟶]

} ⟶ conclusion ↓

Count of Possible nesting of Envelope.