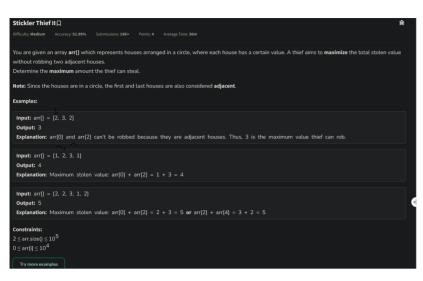
## Stickler Thief 11:



arr [] = (2,3,2)

1. 0 - robbid

2. 0 - not robbed

2. (1)

alo)

de (i)(0) = non (April 10), de (in)(0)

April stal

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

April (i) = non (April 10), a[1) × de (in)(0)

A

## Best Approach: Using Bottom - Up DP: - O(n) Time & O(n) Space

The idea is to fill the deptable based on previous values for each index, we either include it or exclude it to compute the maximum value. The table is filled in an iterative manner from j=2 to j=n-1.

The dynamic frogramming relation is as follows: -

The idea is to split the circular house robbery problem into two linear subproblems by considering two scenarios: one where we exclude the last house (array [0 - . - n-2]) and another where we exclude the first house (array [1 - - n-1))

New Section 6 Page 2