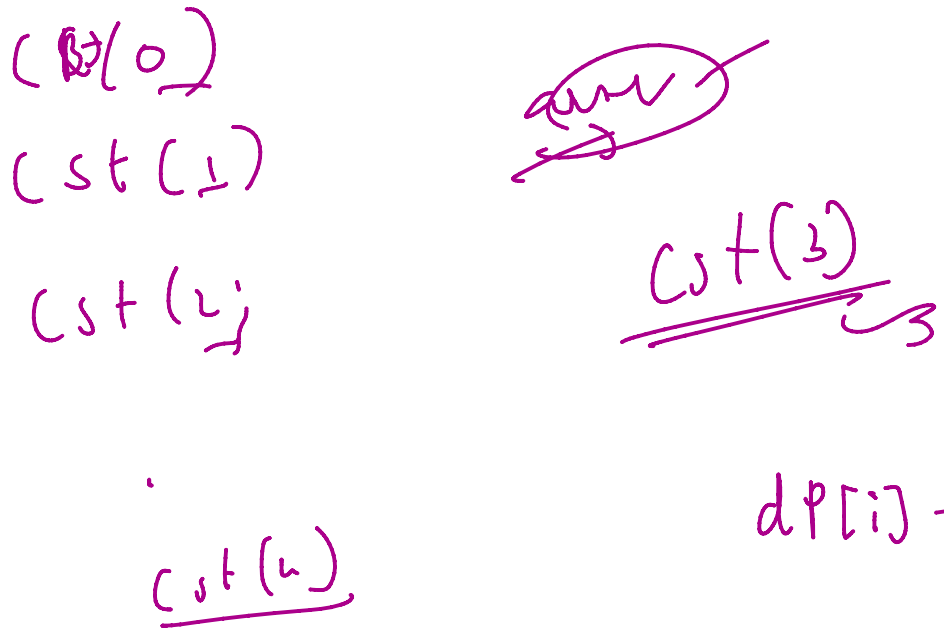
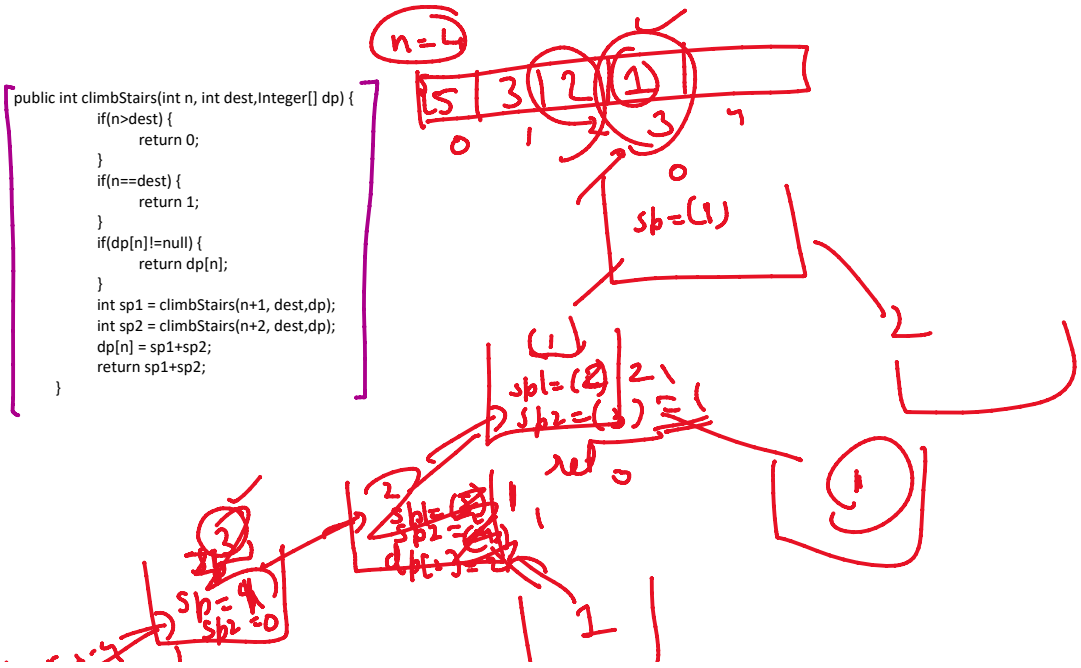
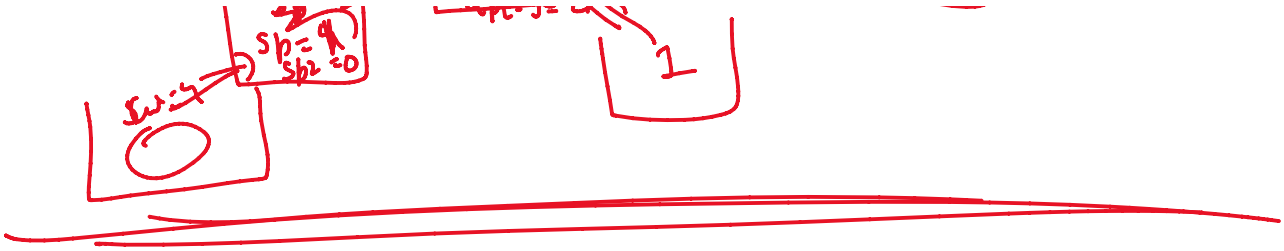
27 May 2023 14:55



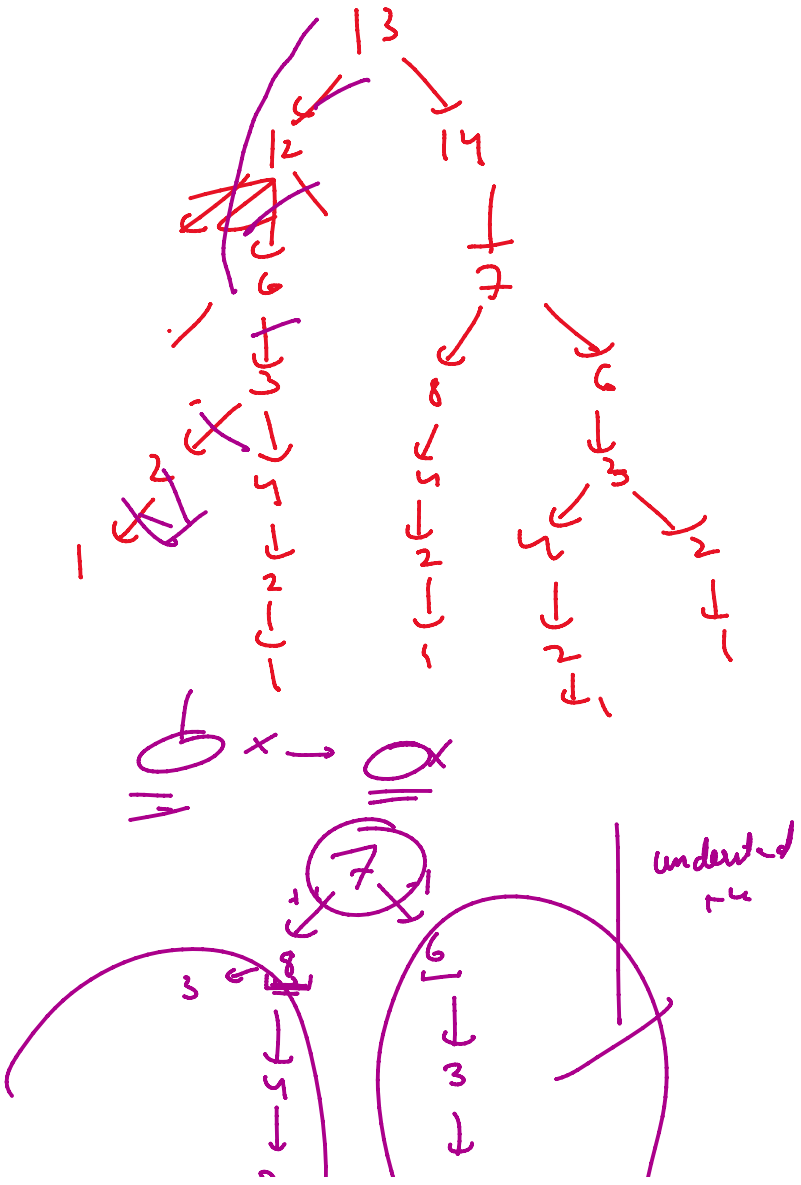

```
public int climbStairs(int n, int dest,Integer[] dp) {
		if(n>dest) {
			return 0;
		}
		if(n==dest) {
			return 1;
		}
		if(dp[n]!=null) {
			return dp[n];
		}
		int sp1 = climbStairs(n+1, dest,dp);
		int sp2 = climbStairs(n+2, dest,dp);
		dp[n] = sp1+sp2;
		return sp1+sp2;
	}
```

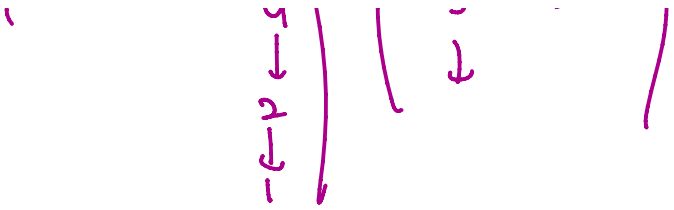
From <https://leetcode.com/problems/climbing-stairs/submissions/958161408/>

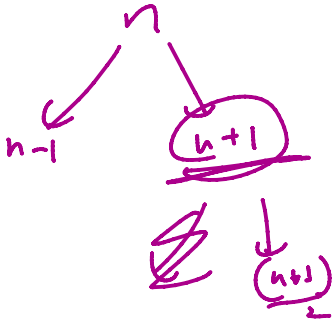




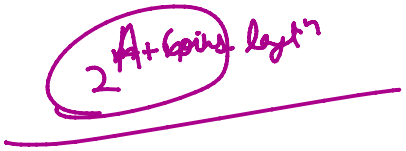
Given a positive number N your task is to bring this number to 1 by performing only a set of operations
. The operations can be either dividing the number by 2 only if the number is even or you can add
or subtract 1 only if the number is odd.
More Precisely:
1) N=N/2 (if N is even)
2)N=N+1/ N=N-1 (if N is odd)
Your task is to minimize these number of operations.
**Input Format**
A single positive integer N
**Constraints**
n<=100000
**Output Format**
Print on a single line the minimum number of steps needed to reach 1 by performing the given operations.
**Sample Input**
8
**Sample Output**
3
Explanation
8->4->2->1

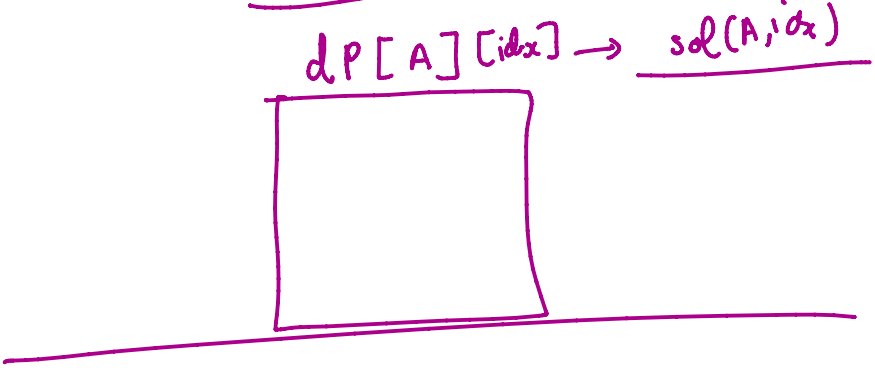







BU → Convert BP to smaller prob such that you can use your DP, ~~stack~~ data struc again




log C. Subseq.

text1 = "abcde", text2 = "ace"