Longest common subsequence [can have gap]

✓ Longest common substring [can't have gap, continuous]

○ print LCS

✓ shortest common subsequence (after merging contain both subsequence)

○ print SCS

✓ min # of insertion & deletion

✓ Longest repeating subsequence

length of longest subsequence of a which is substring to subsequence [after matching count how many times a appears as subsequence in b]

✓ longest palindromic subsequence

substring

min # of del in string to make it palindrome

min # of insertion in string to make it palindrome

Base case

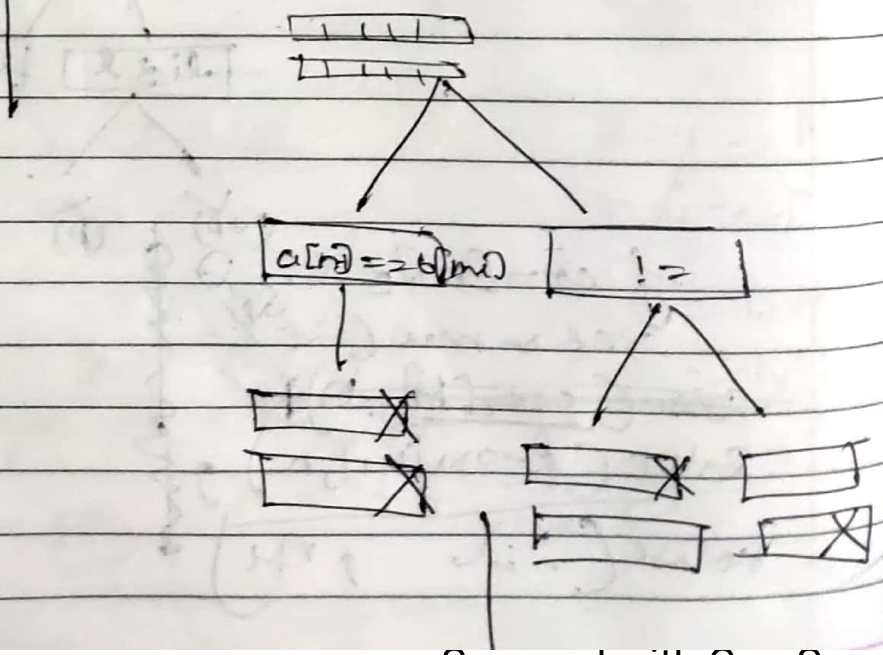
a = u ——— u → 0

b = u ——— u

if len(a) == 0 or len(b) == 0

return 0

choice diagram



if(a[n-1] == b[m-1])

return 1 + solve(n-1, m-1)

if (——) != ——

return max (

solve(n-1, m),

solve(n, m-1))