Train VII

Oct 25. Ye Cao

Problem 1.

Longest Palindrome Subsequence

Suffix Problem [i:]
Prefix Problem [:i]

Suffix Problem Example
Black Jack - Deck of cards dealt from top
Text Justification - Word line always begins from front

Third Type of problem [i:j] subsequence

(GeeksforGeeks) -> 5
EEKEE
EESEE
EEFEE

Brute force :
1. Choose start end index i,j -> $\binom{n}{2}$ possibili
2. Try all combination of subsequences with i,j
3. Check if it's palindrome

Already nonpolyno mial

DP Approach

type of DP : [i : j]

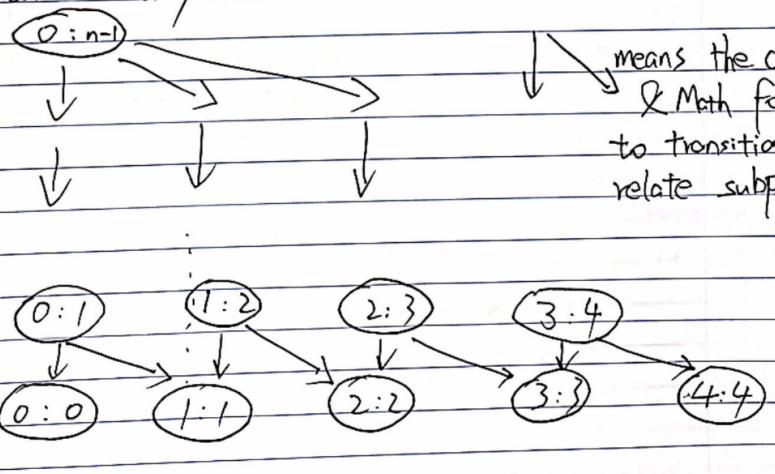
1. Define subproblem :
   - what's the length of the longest palindrome subsequence within the subsequence [i : j] inclusive.
   - Thus the result is solved by dp[0, n-1]

2. Guessing :
   - ~~Where~~ the length of longest palindrome lies ?
     i.e. what are i, j ?
   - What is the length of [i : j] ?
     i.e. L = 1, 2, 3, ..., n ?

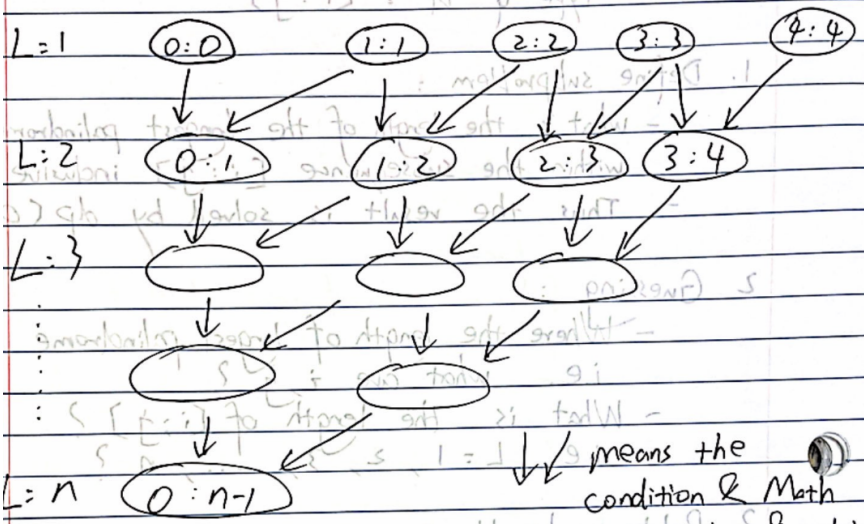
3. Relate subproblems

① Compare recursive + memorization & ② build DP-table + loop

① Recursion way + memo



② Build up DP table + Loops



means the condition & Math to transion & relate subproblems

Notice that the only difference is the order in which things are computed i.e. Bottom up starts from base case, whereas recursion starts from top

If you put two computational DAG (directed acyclic graph) together, you will find that they are symetric.

Now it's time to figure out the most important part i.e. transition function, which is the most tricky part

$$dp[i][j] = \begin{cases} s[i] = s[j] \ ? \ 2 : 1 & j - i = 1 \\ dp[i+1][j-1] + 2 & s[i] = s[j] \\ \max(dp[i][j-1], dp[i+1][j]) & s[i] \neq s[j] \end{cases}$$

Notice the transition equation has order. i.e. you need to check the condition from top to bottom rather than like usual math formula.

4. Solve the original problem

$dp[0][n-1]$ $O(1)$.