DYNAMIC PROGRAMMING

1. Number of Binary String in which two consecutive zeros can’t occur maximum size upto 7.

Logic:-

0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| String end with 0 | **0** | **1** 0 | **1** 10 | **2** 010  110 | **3** | **4** | **5** | **6** |
| String end with 1 | **0** | **1**  1 | **2** 01  11 | **3**  011  111  101 | **4** | **5** | **6** | **7** |

Ans:- there will be 13 String… 7+6=13;

1. In LCS if last indexed char matched in both String then need to check for remaining char in both string if it didn’t matched then check first whole String in 2nd remaining String + check 2nd whole String in first remaining String.

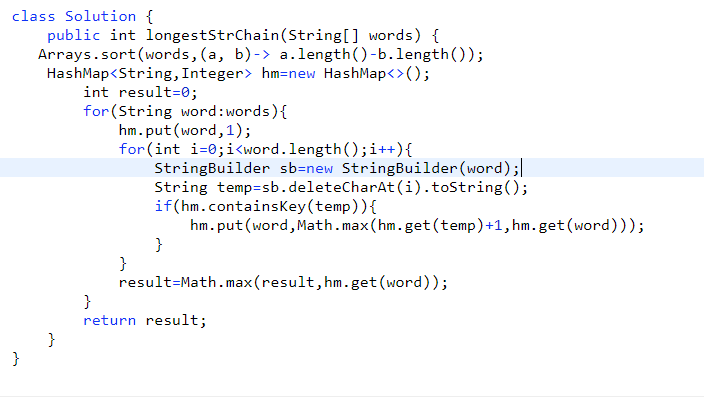
ABED GBCD

ABE GBC

AB in GBC

GB in ABC

1. For longest increasing subsequence Need to check for longest increasing subsequence which ends for particular indexed element.
2. Recursion leap of faith --- break the big cases in base cases.
3. For count of palindromic substring , in dp matrix the base case always would be diagonal element. And traversing also based on diagonal traversing
4. Longest palindromic subsequence is extended version of longest palindromic string
5. Longest alternative subsequence …Consider 2 variable 1 contain length alternative sequence in which 1st index value is greater that 0th index and 2nd variable contains las in which 1st index value is smaller than 0th index and at last print max of these 2.
6. Longest String Chain #LeetCode Question

  
***Greedy Approach***

