**SOLUTION**

// SIMILIAR TO LONGEST COMMON SUBSEQUENCE DP

class Solution {

public:

Solution(){

ios::sync\_with\_stdio(false);std::cin.tie(nullptr);std::cout.tie(nullptr); }

int maxUncrossedLines(vector<int>& A, vector<int>& B) {

int m = A.size();

int n = B.size();

if(m == 0 || n == 0)

return 0;

vector<vector<int>> dp(m+1,vector<int>(n+1,0));

for(int i= 1;i < m+1; i++){

for(int j = 1; j < n+1; j++){

dp[i][j] = A[i-1] == B[j-1] ? 1 + dp[i-1][j-1] : max(dp[i-1][j], dp[i][j-1]);

}

}

return dp[m][n];

}

};

**TIME COMPLEXITY= O(M\*N)**

**SPACE COMPLEXITY= O(M\*N)**