

# Longest Common Subsequence

## 最长公共子序列

$dp[i][j]$  为  $S1[0] \sim S1[i-1]$  与  $S2[0] \sim S2[j-1]$  的最长公共子序列

对于  $S1[i-1]$  与  $S2[j-1]$ :

1.  $S1[i-1] == S2[j-1]$ , 则  $dp[i][j] = dp[i-1][j-1] + 1$  (问题转换为求  $dp[i-1][j-1]$ , 即  $S1[0 \sim i-2]$ ,  $S2[0 \sim j-2]$  的最长公共子序列)
2.  $S1[i-1] \neq S2[j-1]$ , 则  $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

```
#include <bits/stdc++.h>

using namespace std;

int LCS(string s1, string s2, int len1, int len2) {
    vector<vector<int>> dp(len1 + 1, vector<int>(len2 + 1));
    for (int i = 0; i <= len1; ++i) {
        for (int j = 0; j <= len2; ++j) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (s1[i - 1] == s2[j - 1]) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            } else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }
    return dp[len1][len2];
}

int main() {
    int T;
    scanf("%d\n", &T);
    while (T--) {
        int len1, len2;
        scanf("%d %d", &len1, &len2);
        string s1, s2;
        cin >> s1 >> s2;
        printf("%d\n", LCS(s1, s2, len1, len2));
    }
    return 0;
}
```