

Longest Repeated Subsequence

最长不重复子序列（LCS的变种），找到最长的重复子序列，且两个子序列里的相同元素不能是元素组中同一位置的元素

LCS + 不是同一个位置的判断

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

using namespace std;

// dp[i][j] 表示s[0 ~ i - 1]与s[0 ~ j - 1]的LRS
int LRS(string &s, int N) {
    vector<vector<int>> dp(N + 1, vector<int>(N + 1));
    for (int i = 1; i <= N; ++i) {
        for (int j = 1; j <= N; ++j) {
            if (s[i - 1] == s[j - 1] && (i - 1) != (j - 1)) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            } else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }
}

// print LRS
// int i = N, j = N;
// string res;
// while (i && j) {
//     if (dp[i][j] == dp[i - 1][j - 1] + 1) {
//         res += s[i - 1];
//         j--;
//         i--;
//     } else if (dp[i][j] == dp[i - 1][j]) {
//         i--;
//     } else {
//         j--;
//     }
// }
// reverse(res.begin(), res.end());
// printf("%s\n", res.c_str());
return dp[N][N];
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        string s;
```

```
        cin >> N >> s;  
        printf("%d\n", LRS(s, N));  
    }  
    return 0;  
}
```