题意：给定一个长度为n的字符串，共有m次询问，每次询问给两个字符 $x, y$ 和长度 $I$，求以字符串 $xy$ 结尾的长度为 $I$ 的不同子序列有多少个，对 $1e9+7$ 取模。

其中，$n \leq 1e3, m < 1e5$。

```
#include<bits/stdc++.h>
using i8 = signed char;
using u8 = unsigned char;
using i16 = signed short int;
using u16 = unsigned short int;
using i32 = signed int;
using u32 = unsigned int;
using f32 = float;
using i64 = signed long long;
using u64 = unsigned long long;
using f64 = double;
using i128 = __int128_t;
using u128 = __uint128_t;
using f128 = long double;
using namespace std;
const i64 mod = 1e9 + 7;
const i64 maxn = 1e6 + 5;
const i64 inf = 0x3f3f3f3f3f3f3f3f;
void solve() {
  int n; string s; std::cin >> n >> s;
  s = " " + s;
  vector<int>idx(30, -1);
  for (int i = 1; i <= n; i++) {
    idx[s[i] - 'a'] = i;
  }
  vector g(n + 1, vector<vector<pair<int, int>>>(30));
  int q; std::cin >> q;
  for (int i = 0; i < q; i++) {
    char x, y; int l;
    std::cin >> x >> y >> l;
    g[l][y - 'a'].push_back({i, x - 'a'});
  }
  vector<i64>ans(q);
  vector dp(n + 1, vector<i64>(30, 0));
  for (i64 i = 0; i <= n; i++)dp[i][26] = 1;
  for (int i = 0; i < n; i++) {
    vector dp2(n + 1, vector<i64>(30, 0));
    auto inner = g[i + 1]; // i 为长度,查询长度为l - 1 = i 的串
    for (int j = 1; j <= n; j++) {
      int p = s[j] - 'a';
      if (idx[p] == j and inner[p].size()) {
        auto k = inner[p];
        for (auto [id, pre] : k) {
          int tmp = dp[j][pre];
          ans[id] = tmp;
        }
        inner[p].clear();
      }
      int sz = i >= 1 ? 26 : 27; // i + 1为当前处理的长度
      for (int k = 0; k < sz; k++) {
```

```
        dp2[j][p] = (dp2[j][p] + dp[j - 1][k]) % mod;
      }
      for (int k = 0; k <= 25; k++) {
        if (k != p) {
          dp2[j][k] = dp2[j - 1][k];
        }
      }
    }
    dp = dp2;
  }
  for (int i = 0; i < q; i++) {
    std::cout  << ans[i] << "\n";;
  }
}
int main() {
  int T; std::cin >> T;
  while (T--)
    solve();
}
```