Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8077427	00:19:42	Stack Construction	✔ **Accepted**	0.32 s	C++
	✔ ✔ ✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
stack.cpp	2386 bytes	b9b51457d9b5ed70e023b55de752c5874ead4eff	download

Edit and resubmit this submission.

stack.cpp

```
1  //https://open.kattis.com/problems/stack
2  #include <bits/stdc++.h>
3  const long long Maxi  = 9e18;
4  class Solution {
5  public:
6    long long mfind(std::map<char, long long> &id, char c) {
7      auto it = id.find(c);
8      if (it == id.end()) {
9        id.insert({c, id.size()});
10       return id.size() - 1;
11     }
12     else{
13       return it->second;
14     }
15   }
16
17   void Solve_Storing(long long &v, long long l, long long r) {
18     if (l == Maxi || r == Maxi) {
19       return;
20     }
21     else{
22       v = fmin(v, l+r);
23     }
24   }
25
26   void StackConstruction()
27   {
28     std::string S;
29     std::getline(std::cin, S);
30     map<char, long long> id;
31     vector<std::vector<long long>> pos(256,std::vector<long long>());
32     std::vector<long long> mpos(S.length(), 0);
33     for (long long i = 0; i < (long long)S.length(); ++i) {
```

Help

```
34        long long mid = mfind(id, S[i]);
35        mpos[i] = pos[mid].size();
36        pos[mid].push_back(i);
37      }
38      std::vector<std::vector<long long>> dp(S.length(), std::vector<long long>(S.length(),
    Maxi));
39
40      for (long long r = 0; r < (long long)S.length(); r++)
41      {
42        for (long long l = r; l >= 0; --l)
43        {
44          dp[l][r] = 3LL * (r - l + 1LL);
45          for (long long m = l; m < r; ++m)
46          Solve_Storing(dp[l][r], dp[l][m], dp[m+1][r]);
47          if (S[l] == S[r])
48          {
49            std::vector<long long> &p = pos[mfind(id, S[l])];
50            long long li = mpos[l], ri = mpos[r];
51            std::vector<long long> tdp(ri - li + 1,Maxi);
52            tdp[0] = 3LL;
53            for (long long i = li + 1; i <= ri; ++i)
54            {
55              tdp[i - li] = tdp[0] + 1LL;
56              if (p[li] + 1 < p[i])
57                tdp[i - li] += dp[p[li]+1][p[i]-1];
58              for (long long j = li + 1; j < i; ++j)
59              {
60                long long tv = tdp[j - li] + 1LL;
61                if (p[j] + 1 < p[i])
62                  tv += dp[p[j]+1][p[i]-1];
63                if (tv < tdp[i - li]) tdp[i - li] = tv;
64              }
65            }
66            if (tdp.back() < dp[l][r])
67              dp[l][r] = tdp.back();
68            }
69          }
70        }
71        std::cout << dp[0].back() <<"\n";
72    }
73  };
74
75  int main(){
76    std::ios_base::sync_with_stdio(false);
77    std::cin.tie(0);
78    int N;
79    Solution S1= Solution();
80    std::cin >> N >> std::ws;
81    while (N--){
82      S1.StackConstruction();
83    }
84    return 0;
85  }
86
87
88
```

Help