

資料結構實習

11/03 作業報告

Stack 實作 (e, s) segment

班級：資訊二甲

學號：D1109023

姓名：楊孟憲

Contents

1	引言	3
2	題目敘述	4
3	作法	5
4	執行結果	6
5	心得與討論	7

1 引言

Stack 是一個只有一個開口的單向輸入輸出的資料結構，在日常生活中可以使用該資料結構維護許多演算法。插入以及拿取都只需要 $O(1)$ 的時間。以下是利用 LinkedList 實作 Stack 的方法。(以下是使用 C 實作 Stack 的 push/pop)

```
1  /*****
2  # File Name: a.cpp
3  # Author: Vincent
4  # Mail: mengxian0913@gmail.com
5  # Created Time: 四 11/16 21:06:37 2023
6  *****/
7
8  #pragma GCC optimize("O3")
9  #include <bits/stdc++.h>
10 using namespace std;
11 #define int long long
12 #define ff first
13 #define ss second
14 #define fastIO cin.tie(nullptr)->sync_with_stdio(false);
15 #define INF 0x7FFFFFFF
16 #define pb push_back
17 #define all(aa) aa.begin(),aa.end()
18 #define MOD (int)(1e9+7)
19
20 void solve(){
21     int n;
22     cin >> n;
23     cout << n << "\n";
24     return;
```

```
25 }  
26  
27 signed main(){  
28     fastIO  
29     solve();  
30     return 0;  
31 }
```

2 題目敘述

題意說明： 如果在一個字串當中其頭一個字母為 E，最後一個字母為 S，而兩個字母中間不包含任何 E 或 S 字母的話，則稱為 ES 字串。請利用堆疊（Stacks）的原理，撰寫出一個程式，從檔案（input.txt）讀入一段文章或是字串，然後消去所有可能的 ES 字串，使得消去後的字串輸出不包含任何 ES 字串。

3 作法

使用 C++ 的 STL Stack 實作。遍歷字串的每一個字元，當前字元為's' 時，判斷 stack 是否為空並且如果前面有'e' 的話 (cnt > 0)，就不斷地把先前的字元從 stack 拿出來，直到 now == 'e'，並且將 cnt-1，這能夠確保之後的 e, s 區間能被正確的移除。否則，就把's' 放進 stack 裡。

如果當前字元不為's' 的話，就放進 stack 裡，並且判斷如果當前字元為'e' 的話，就 cnt++;

範例程式碼

```
1  /*****
2  # File Name: a.cpp
3  # Author: Vincent
4  # Mail: mengxian0913@gmail.com
5  # Created Time: 四 11/16 21:06:37 2023
6  *****/
7
8  #pragma GCC optimize("O3")
9  #include <bits/stdc++.h>
10 using namespace std;
11 #define int long long
12 #define ff first
13 #define ss second
14 #define fastIO cin.tie(nullptr)->sync_with_stdio(false);
15 #define INF 0x7FFFFFFF
16 #define pb push_back
17 #define all(aa) aa.begin(),aa.end()
18 #define MOD (int)(1e9+7)
```

```
19 |
20 | void solve(){
21 |     int n;
22 |     cin >> n;
23 |     cout << n << "\n";
24 |     return;
25 | }
26 |
27 | signed main(){
28 |     fastIO
29 |     solve();
30 |     return 0;
31 | }
```

4 執行結果

輸入輸出結果：



Figure 1: 輸入 3

5 心得與討論

這次的作業利用 Stack 的原理來完成，是一個常見的模板題，實作起來沒有什麼問題，這讓我更加了解 Stack 的運作原理。