

tags: 資料結構

HW6

題目給定四則運算前序以及後序表達式，要求分別轉成後序及前序，在性質上我們可以知道不管是前轉後或是後轉前的步驟都一樣，只是擺放符號及字符的順序相反，因此此處僅講解其中一種做法。

pre2post

我們有前序表達式 $++A/^CDE*FG$ ，要轉成後序，這邊我們從後到前依序取出字符並放入 stack，當遇到符號時就出棧兩次並組成新字符重新入棧，詳細操作如下。

| | |
|--------------------------|----------------------|
| G in stk | -> G |
| F in stk | -> G F |
| (* add F and G) in stk | -> FG* |
| E in stk | -> FG* E |
| D in stk | -> FG* E D |
| C in stk | -> FG* E D C |
| (^ add C and D) in stk | -> FG* E CD^ |
| (/ add CD^ and E) in stk | -> FG* CD^E/ |
| A in stk | -> FG* CD^E/ A |
| (+ add A and CD^E/) | -> FG* ACD^E/+ |
| (+ add ACD^E/+ and FG*) | -> ACD^E/+FG*+ |

pseudo code 如下，需要注意的地方有在 pre2post 是 $stk_{top} + stk_{top} + operator$ 但 post2pre 則是 $operator + stk_{top} + stk_{top}$ 。

```
1  stack<string> stk;
2  for (int i=str.length()-1; i>=0; i--) {
3      if (is_operator(str[i]))
4          tmp = stk.top(), stk.pop(),
5          tmp += stk.top(), stk.pop(),
6          tmp += str[i],
7          stk.push(tmp);
8      else
9          tmp = str[i], stk.push(tmp);
10 }
11 ans = stk.top();
```

最後根據上述 pseudo code 我們可以知道因為只會跑過一次迴圈，所以時間複雜度為 $O(N)$ ，而在 stack 部分我們最長只會入棧整串 string，因此空間複雜度也是 $O(N)$ ， N 為字串長度。