

HW program 3

41147009S 陳炫佑

README

等待輸入

輸入以0~9還有 + , - , * , / , (,) 組成的算式，總共不超過20個字

如果輸入正確

顯示前序、後序、level-order的此算式以及結果

EX:

```
Please enter an infix expression and press enter:
3+5*(2+8)/4
The postfix expression: 3528+*4/+
The prefix expression: +3/*5+284
The level-order expression: +3/*45+28
= 15.500000
請按任意鍵繼續 . . .
```

離開程式

輸入EOF(End of File), 在linux使用ctrl+Z

時間複雜度分析

輸入

時間複雜度: $O(N)$

原因:對於每個符號目前優先度，並對stack做處理，最後會剛好pop整個stack

並且數字也是每次都push到答案stack，在做轉換時pop出來兩個push回去，最後只剩一個，也就是該算式之binary tree，而stack的pop，push時間複雜度都是 $O(1)$

所以總共是 $O(N)$

顯示前序

時間複雜度: $O(N)$

原因:dfs該binary tree

dfs方式為:輸出目前符號->dfs(左子樹)->dfs(右子樹)

最後會走訪每個node，所以是 $O(N)$

顯示後序

時間複雜度: $O(N)$

原因:dfs該binary tree

dfs方式為:dfs(左子樹)->dfs(右子樹)->輸出目前符號

最後會走訪每個node，所以是 $O(N)$

顯示**level-order**

時間複雜度: $O(N)$

原因:bfs該binary tree

bfs方式為:將目前節點左右子節點丟入走訪queue裡面

再將目前節點pop並輸出該節點符號

最後會走訪每個node，所以是 $O(N)$

顯示答案

時間複雜度: $O(N)$

原因:

對於每個節點操作為:對(左子節點計算結果與右子節點計算結果)做該節點之符號操作，即`calculation(now.left) now.sign calculation(now.right)`

若為葉節點則當前符號為其運算結果

最後會走訪每個node，所以是 $O(N)$

錯誤判斷

時間複雜度: $O(N)$

原因:

對於每個符號/數字測試其前後合法與否，遍歷每個符號，所以是 $O(N)$