

ACM-ICPC OB/OG Spring Contest 2014

Problem E - Parentheses

原案：岩田

解答：伊藤、保坂、森、山口

解説：伊藤

問題概要

- 文字列が n 個入力される
 - それぞれ、'('と')'で構成される文字列
- 文字列の順番を自由に並び替えていい
- 並び替えた後に繋ぎ合わせた文字列を括弧対応のとれた文字列にできるか答えよ

Sample Input 1

- 入力

1. ()(())

2.))()()()

3.)()()()

- 出力 : Yes

- 次の順番で並べたとき括弧対応をとれる

- $(\textcircled{1}) (\textcircled{2}) (\textcircled{3}) () () () () () ()$

Sample Input 2

- 入力

1.))0((

2.))((0)(

- 出力 : No

- ①→②でも②→①でも括弧対応はとれない

考察I

- 左からどのような順で繋げるか考えてみる
- **ネストの深さが変化しない文字列**は必ず使う
 - 状況を何も変えずに文字列を消費できるから
- **深さがマイナスにならないように**だけ注意

$$\begin{array}{ccc} \text{((} & + & \text{())} \\ \text{現在のネスト} & & \text{ネストの深さ} \\ \text{の深さ2} & & + 2 - 2 \end{array} = \begin{array}{c} \text{((} \text{())} \\ \text{ネストの深さは} \\ \text{2で変わらないまま} \end{array}$$

$$\begin{array}{ccc} \text{((} & + & \text{)))} \\ \text{現在のネスト} & & \text{ネストの深さ} \\ \text{の深さ2} & & - 3 + 3 \end{array} = \begin{array}{c} \text{深さが途中でマイナスに} \\ \text{なったため不採用} \end{array}$$

考察2

- ネストが浅くなる文字列より先に**深くなる文字列**を使うべき
 - ネストが浅くなる文字列によって後から全ての括弧をまとめて閉じられるから
 - 文字列を使う順番はいつでもいい
 - 深さがマイナスにならないようにだけ注意

$$\begin{array}{ccc} \text{((} & + & \text{))} \\ \text{現在のネスト} & & \text{ネストの深さ} \\ \text{の深さ2} & & -1+2 \end{array} = \begin{array}{ccc} \text{((} & \text{))} & \text{((} \\ \text{ネストの深さ3} & & \end{array}$$

考察3

- 残りの、ネストが浅くなる文字列は？
 - 考察2のように深さがマイナスにならない文字列を貪欲に繋げる方法はダメ
- ダメな例： ((、)、)) (という3つの文字列
 1. ((→)
 2. ((→)) (
- 深さが負ではないため、どちらも選択可能
 - 1を選ぶと、)) (を繋げなくなる
 - 2を選べば、)を繋げてYes

考察4

- ネストが浅くなる文字列は、**右から読めば考察2と同じように考えることが可能**



想定解法

- 貪欲法

- **ネストが浅くならない文字列**を貪欲に連結
- これを左側と右側から行う
- 左側のネストの深さと右側のネストの深さが一致すればYes

- ココで落ちる！

- **ネストの深さがマイナスになる文字列**は連結しないこと
 - (→))((→) は、上記の貪欲法を満たすがダメ

結果

- AC / Submit
 - 25 / 132 (19%)

- First Accept
 - 12:42
 - #ちよくだいさんのお金でたこ焼き
食べ放題