プログラミング基礎演習
第11回

金（5限）

長谷川禎彦

連絡事項

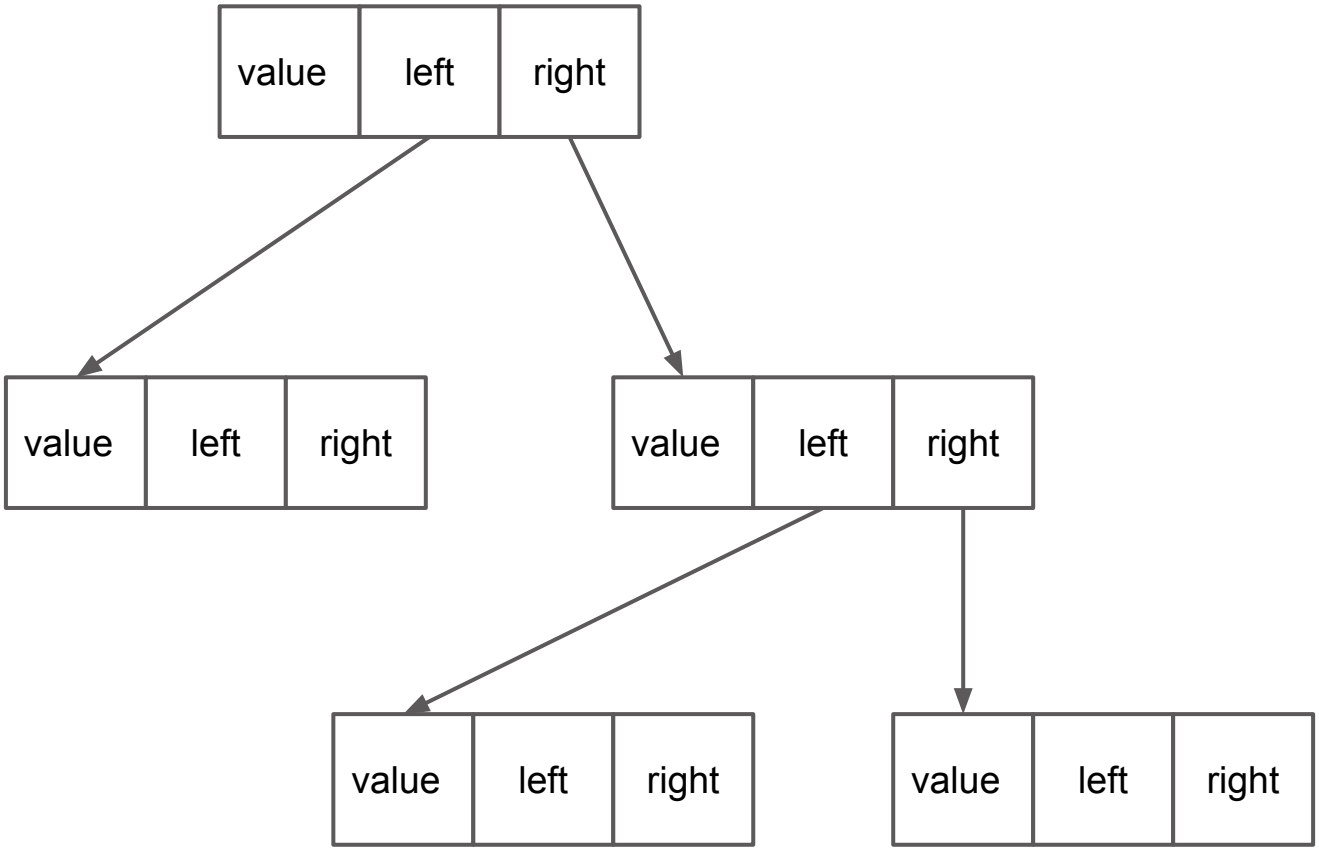
- 12/19小課題提出者一覧はホームページ
  - プログラム例もホームページ
  - 課題2は課題3のサブセットなので，載せていない
- レポート提出者一覧はホームページ
  - zipファイル破損で読めなかったため，レポート再送が必要な人がいるが一覧には含まれている
  - zip破損による要再提出の人はメールで提出すること

今日の学習事項

- 木構造
  - 連結リストの一般化
- 再帰＋ポインタの組み合わせ

- 第二回のレポートはほとんど今日の内容と同じ

木構造


value
left
right
value
left
right
value
left
right
value
left
right
value
left
right

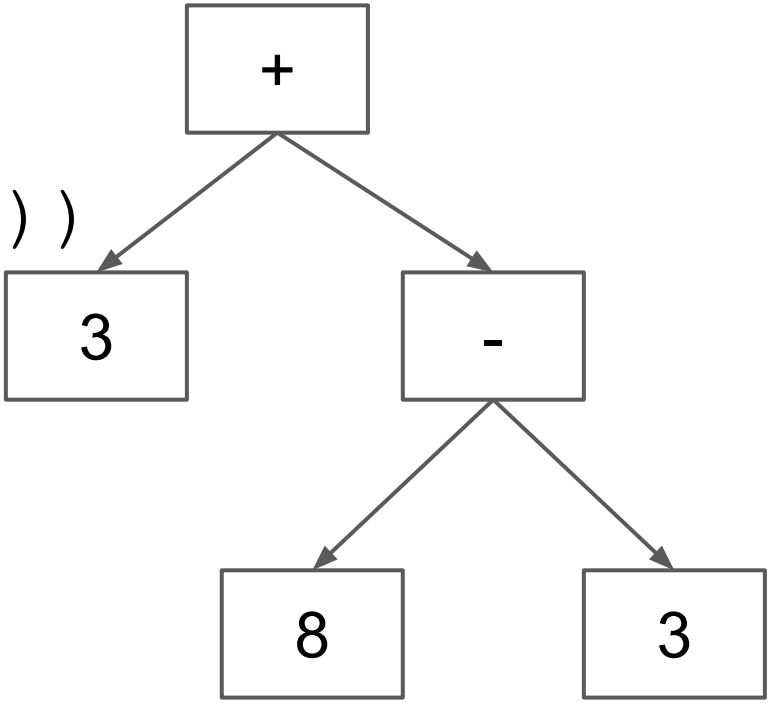
木構造のノードの構造体

```
struct node {
   char value;
   struct node *left;
   struct node *right;
}
```

式の表現と木構造

```
+ 3 - 8 3
= plus 3 minus 8 3
= plus(3, minus(8, 3))
= 3 + (8 - 3)
```

＋とーは引数が2であることが分かっているので，括弧がなくても曖昧さはない



課題1

式を表す文字列に従って木構造を生成せよ．また深さ優先探索によって，木構造を表示させよ（つまりもとの文字列がそのまま表示される）

課題1

ここでは簡単のため，数字は一桁の非負の整数に限定する.

+ 3 − 8 3

を

+3−83

のように詰めて書ける（パースがとても楽）.

関数は四則演算（'+', '-', '*', '/'）のみを考える（常に二分木）.

```
#include <stdio.h>
#include <stdlib.h>
struct node {
  char value;
  struct node *left;
  struct node *right;
};
void parse(struct node *n) {
/*埋める*/
}
struct node *create_tree(int *pos, char *s) {
/*埋める*/
}
int main() {
  char *str = "+/9*84-*84+73"; // 22.281250
  int p = 0;
  struct node *root = create_tree(&p, str);
  parse(root);
}
```

このひな形は使わなくても良い

create_node

- struct node *create_node(int *pos, char *s)
- s : 文字列全体（例 : +-833など）
- *pos : 現在の位置
  - ポインタであることが重要
- 返値 : 作成したノードのポインタ

s = “+-833”, *pos = 1の場合，’-’に相当するノード作成する．‘-’を生成したら，関数を出ずに8と3のノードも作る．8を作ったらreturnする．

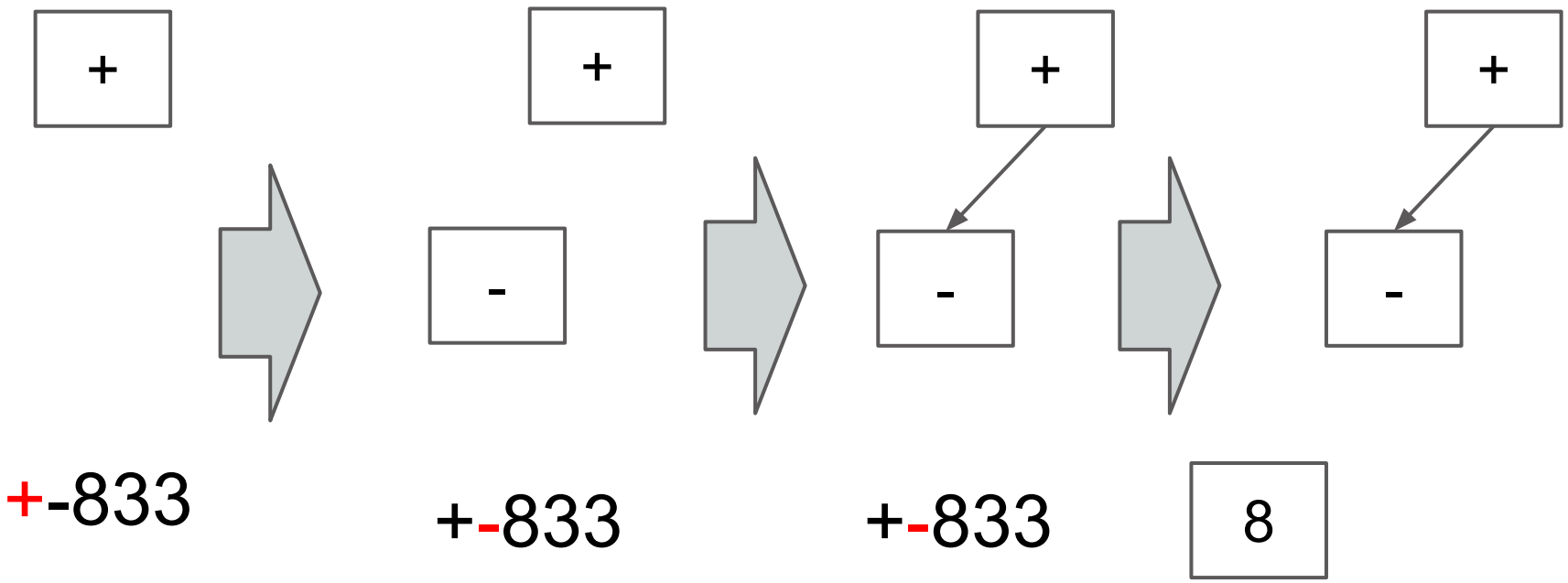
create_node

- create_nodeは再帰的に用いる
  - create_nodeの中でcreate_nodeを呼ぶ
    - 再帰で呼ぶ場合は，*posを一つ増やす
      - 次の記号を読む
  - 終端記号の場合はcreate_nodeを呼ばずに，returnする

parse

- void parse(struct node *n)
- ノードnのvalueをprintfし，さらにn->left, n->rightに関してparseを再帰的に呼ぶ

例:+-833の場合



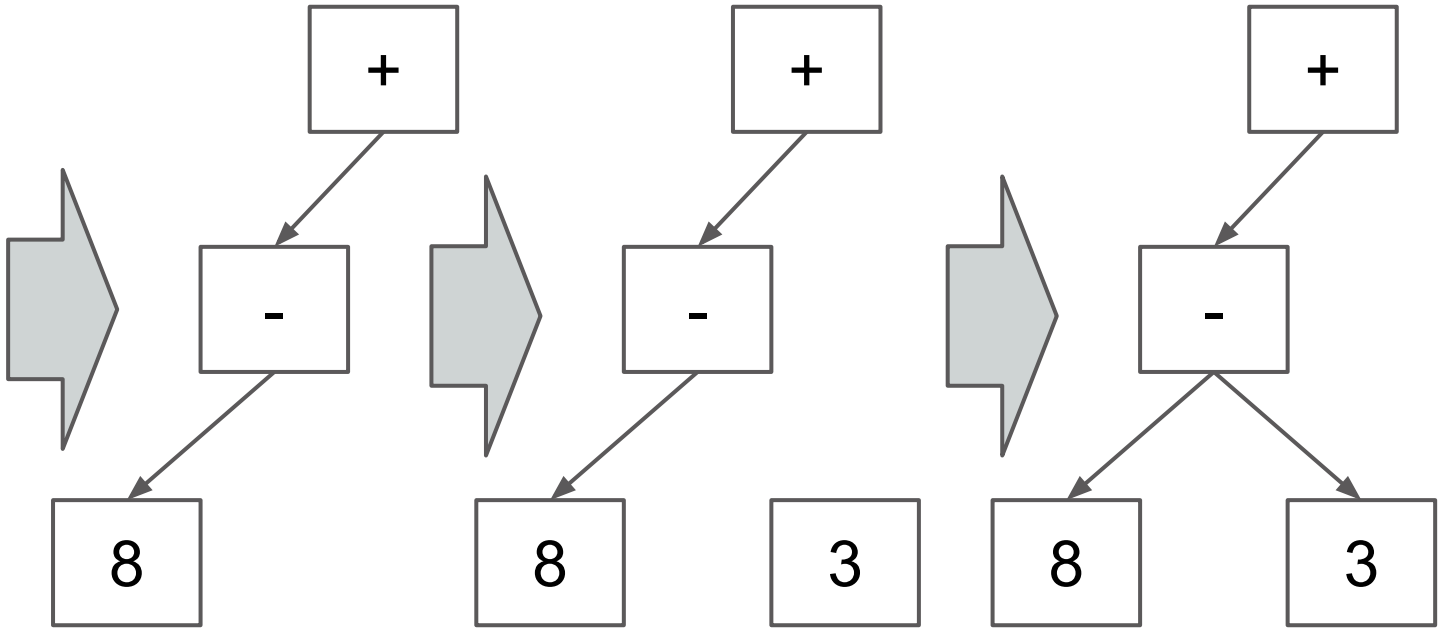
+-833

+-833

+-833

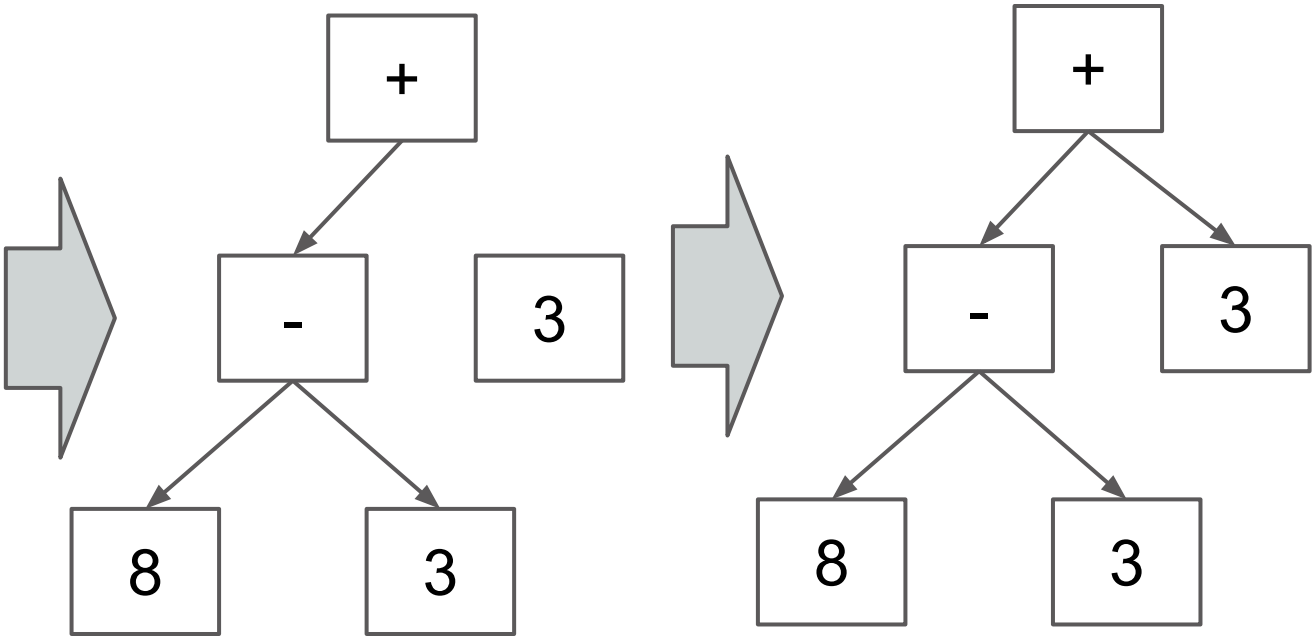
+-833

例:+-833の場合



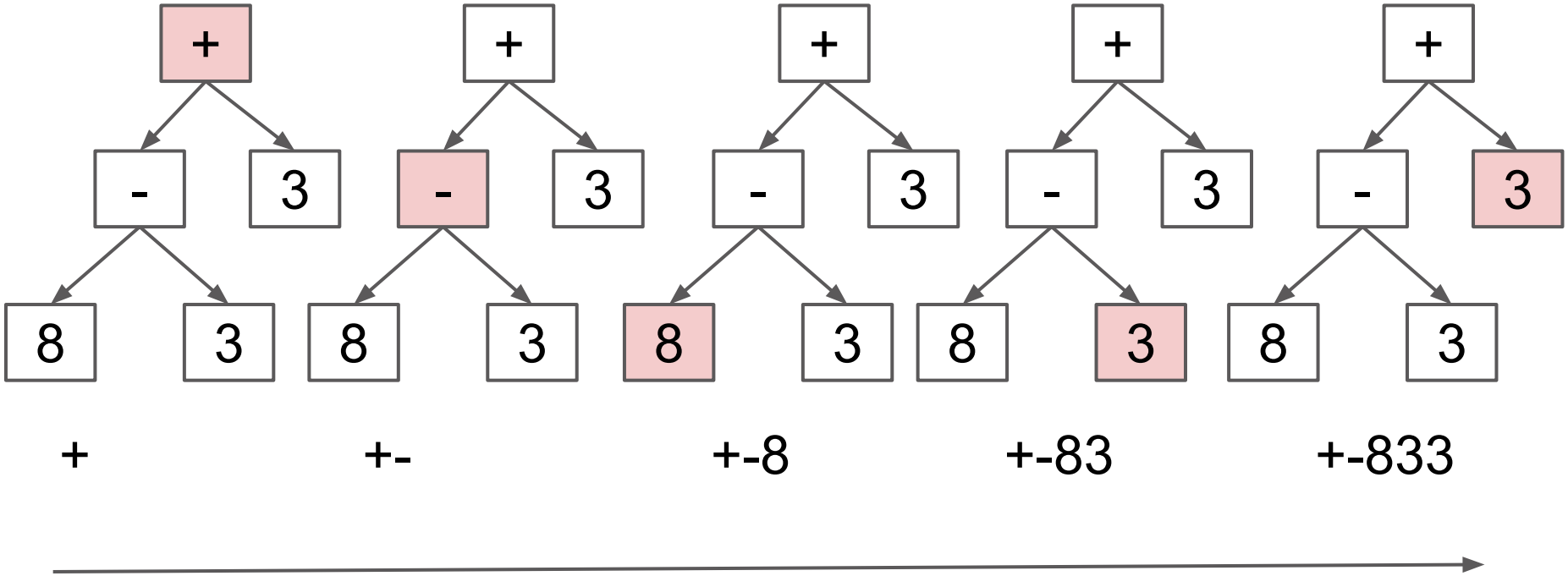
+-833　　+-833　　+-833

例:+-833の場合



+-833　　+-833

深さ優先探索



深さ優先探索も再帰呼び出しによって実装する

課題1

出力する際

```
+3*73
```

を

```
Plus[3,Times[7,3]]
```

のように出力すると，WolframAlphaでそのまま計算できる（チェックできる）

その場合四則演算はPlus, Subtract, Times, Divideであり，括弧は四角括弧にする

課題2 (自由課題)

作成した木構造を実際に計算せよ.
これも再帰によって計算する.

文字列と計算結果の例

```
+/9*84-*84+73
    =22.281250
+-8*2+-8/27-*5*3+-8*2+-8/22-*2*315//2/9755/*2/945
    =234.138095
```

課題提出方法

- 未完でも提出する
- 課題xの答えを`kadai0x.c`ファイルに書く.
  - xは1,2
- **締め切りは火曜日の23:59**
- **ファイルをzipファイルにまとめる**
  - **ファイル名:学籍番号.zip**
  - 例:学籍番号が341234の場合，341234.zipとする.
- 学籍番号.zipファイルのみを提出する
  - 提出は http://goo.gl/hXsfLl
  - フォームの「課題」から「小課題」を選択