

アルゴリズムとデータ構造

平衡木（その1）

演習1（必須課題）

- ランダムBST, スプレイBSTを用いた記号表を作成するプログラムを実装し，動作を確認せよ.

※ ST_BST2.cに追加すべき関数はMoodleから入手可能

この課題は提出不要

演習2（加点課題）

- 関数splayを修正し，既に記号表に存在する要素が入力された場合はsplay操作によってその接点が根となるように木を変形させるプログラムを実装せよ.
- 提出
 - 記号表に10個の要素を以下の順番で挿入して得られる木の形状を表示し，レポート本文として提出すること.
1 2 3 4 5 6 7 8 9 5
 - ソースコードを添付ファイルとして提出すること.

演習2: 補足

- 記号表に存在する要素についても処理するには, STtest.c の赤字の部分コメントアウトする.

```
for (M = 0, N = 0; N < maxN; N++)
{
    if (sw == 1) v = ITEMrand();
    else if (sw == 2) v = N+1;
    else if (ITEMscan(&v) == EOF) break;
    //      item = STsearch(v); if (item.key != NULLitem.key) continue;
    key(item) = v;
    STinsert(item); M++;
}
```

- 標準入力から記号表に要素を挿入するには...

```
$ gcc -o STtest STtest.c Item.c ST.c
$ ./STtest 10 0
```