アルゴリズムとデータ構造

第2回　基本的なデータ構造

http://www-ui.is.s.u-tokyo.ac.jp/~takeo/course/

五十嵐 健夫

takeo@acm.org

前回の内容

イントロダクション
モデル化の例
計算量（OとΩ）

今回の内容

リスト
スタック
待ち行列
木

抽象データ型としての「列」

$a_1, a_2, a_3, \ldots, a_{1n}$　線形順序

insert, indexOf, get, remove, next, prev, clear, first, print, end

列の実現

配列による実現

○ ランダムアクセス　× 挿入と削除

ポインタによる実現（通常のリスト）

× ランダムアクセス　○ 挿入と削除

配列

○ ランダムアクセス　× 挿入と削除

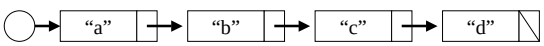
String[] labels = {“a”,”b”,”c” ,”d”}

“a”
“b”
“c”
”d”

リスト

× ランダムアクセス　○ 挿入と削除

```
LinkedList labels = new LinkedList();
labels.add(“a”);
labels.add(“b”);
labels.add(“c”);
labels.add(“d”);
```



スタック

clear, pop, push, empty

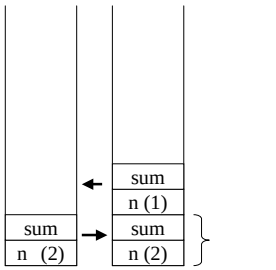
例（再帰呼び出し）

実装 （ポインタと配列）

スタックと再帰呼び出し

$$\sum_{i=1}^{n} i^2$$

```
int foo(int n){
   if (n == 1) return 1
   int sum = foo(n-1)+n*n;
   return sum;
}
```



活動レコード

待ち行列 (Queue)

clear, front, enqueue,
dequeue, empty

例（イベントキュー、データ転送）

実装 （ポインタと循環配列）

木 (Tree)

Insert, delete, member, etc.

階層構造を表す（例：住所）

実装 （ポインタ）

まとめ

配列
リスト
スタック
待ち行列
木