スタックとキュー

山本昌志*

2007 年 7 月 10 日

概 要

データ構造のスタックとキューについて学習する．スタックやキューの実装方法はいろいろあるが，ここでは教科書に沿って説明する．

1 本日の学習内容

1.1 前回の復習

前回は，リストというデータ構造を学習した．イメージは，図 1 のようなものであった．配列とは異なり，ランダムアクセスはできないが，要素の追加や削除が容易なデータ構造である．

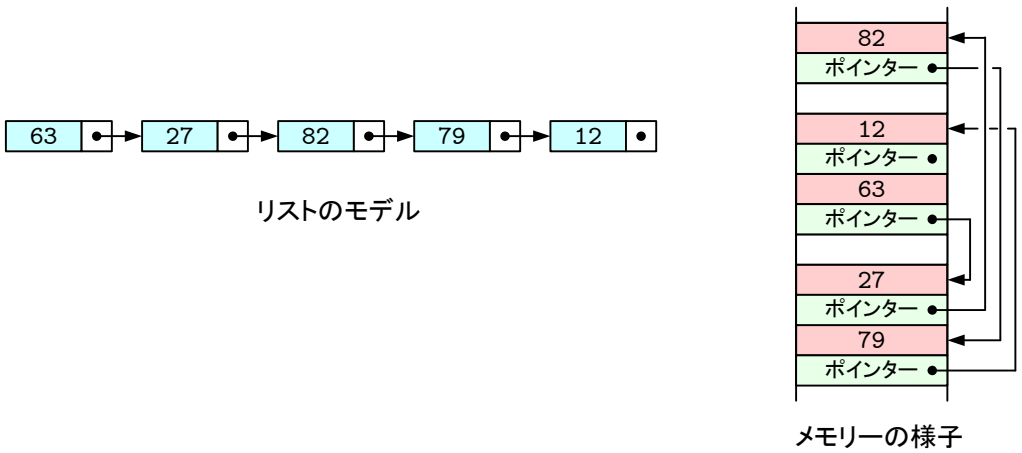

図 1: リスト

1.2 本日の学習内容

本日の学習内容は，データ構造のスタックとキューである．教科書 [1] の pp.184–189 が範囲である．ここでの学習のゴールは以下の通りである．

*独立行政法人　秋田工業高等専門学校　電気情報工学科

- スタックをイメージでき，それを使うことができる．
- キューがイメージでき，それを使うことができる．
- 逆ポーランド記法の計算ができる．

上級者，あるいはプロを目指すものはスタックやキューの使い方にとどまらず，実装方法もきちんと理解すること．

2　スタック

最後に入れたデータをはじめに取り出すスタックと呼ばれるデータ構造について説明する．

2.1　イメージ

スタック (stack) を辞書で調べてみると，次のような意味が書かれている．

1. (干し草などの) 大きな山，積みわら (haystack);(物のきちんとした) 積み重ね
2. (図書館などの) 書棚の列，書架；((the 〜s)) (図書館の)(閉架) 書庫
3. (屋上の) 組み合わせ煙突 (chimney stack); 煙突，煙出し
4. 《コンピューター》スタック:最後に入れたデータを最初に取り出せるようにしたデータ構造

もちろん，情報科学の分野で使われるのは最後の意味で，図 2 のようなデータ構造である．データ構造であるから，データを蓄えることと，それを取り出すことができる．スタックの特徴は，最後に入れたデータが一番最初に取り出されることにある．取り出されるデータは，格納されている最新のデータで，最後に入れられたものが最初に取り出されることから，LIFO(last in first out, 後入れ先出し) と呼ばれる．スタックの途中のデータを取り出すことは許されないのである．

スタックにデータを積むことをプッシュ(push) と，スタックからデータを取り出すことをポップ (pup) と呼ぶ．これらの英語の意味は，

push ＜人・物を＞押す，突く

pop ポンという音を立てる，ひょいとやって来る [出て行く]，急にはいる [出る], ひょっこり現れる

である．

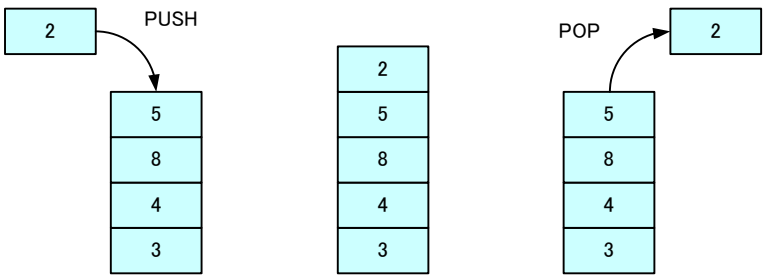

図 2: スタック

2.2 実際の応用

スタックは単純なデータ構造であるが，有用でいろいろな場面で使われる．例えば，次のような場合である．

- 関数を呼び出した場合，呼び出し元のデータをいったん保存するときのデータ構造として使われる．
- 算術式の評価 (教科書 [1] の pp.189–192)．これは逆ポーランド記法でかかれた数式を計算するプログラムである．

カッコの対応を調べるプログラムや逆ポーランド記法を用いた数式の計算 (p.114-115 List 4-3) などでも使われる．授業では説明しないので，興味のある者は自分でソースコードを解析するのが良いだろう．

2.3 スタックの実装

配列を使えば，スタックの実装は簡単である．教科書 [1] では，次のような構造体を宣言している．

```
typedef struct stack{
  int top;
  int size;
  int data[1];
}stack_t;
```

この構造体には，スタックに必要な情報がすべてがある．図 3 に示すようにメンバー変数 top は，次にデータを入れる配列の添え字を表している．size は配列の要素数を表し，このスタックに蓄えることができるデータ数となっている．

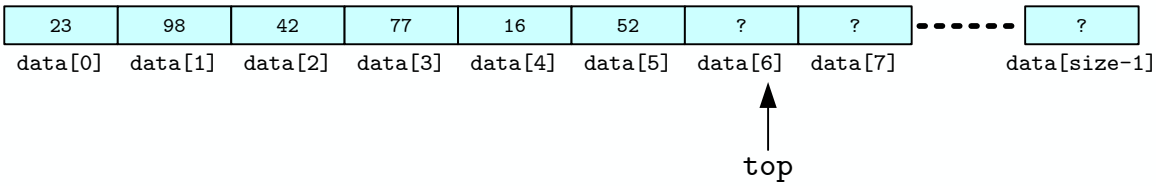

図 3: 教科書 [1] のスタックのイメージ．配列中の?はデータはあるが，意味のない—使い道のない—データを表している．

スタックを実際に使うためには記憶領域が確保と操作を行う必要がある．教科書の stack.c では，次の 3 つの関数でそれを行っている．記憶領域の確保を行う creat_stack とデータを格納する push()，データを取り出す pop() である．

3　キュー

最初に入れたデータをはじめに取り出すキューと呼ばれるデータ構造について説明する．

3.1　イメージ

待ち行列と呼ばれるキュー (queue) も辞書で調べてみた．それには，次のような意味がある．

1. 順番を待つ人・車などの）列 (line)；（待つ）一群の人々
2. 《コンピューター》待ち行列

これは，窓口に並ぶ順番待ちの行列の意味で，図 4 のようなデータ構造となっている．スタックではデータの挿入と取り出しが列の一方からのみであったの対して，キューは列の両端から行う．一方がデータの追加で一方がデータの取り出しとして使われる．キューでは，最初に入れたデータが一番最初に取り出されることにある．取り出されるデータは格納されている最古のデータで，最初に入れられたものが最初に取り出されることから，FIFO(first in first out, 先入れ先出し) と呼ばれる．スタック同様，スタックの途中のデータを取り出すことは許されないのである．

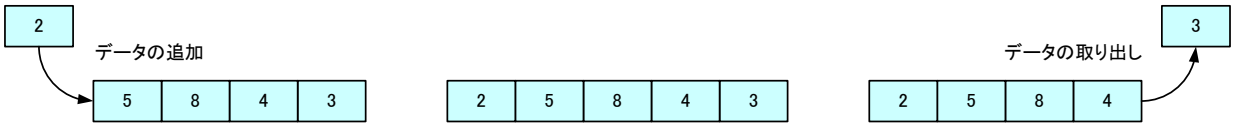

図 4: キュー

キューはスタックに比べて少しばかり，構造が複雑になっている．実際，それを直線的なイメージのメモリーにデータを追加しようとすると，以下のような問題が生じる．

- FIFO を続けると，いずれはメモリーの端に到達して，データの追加が出来なくなる．
- データを追加したり取り出したりする毎に，データの列を移動させることも考えられる．こうすると計算量が増加して不経済である．

これを防ぐために，図 5 のようなリングバッファと言うものが考えられた．

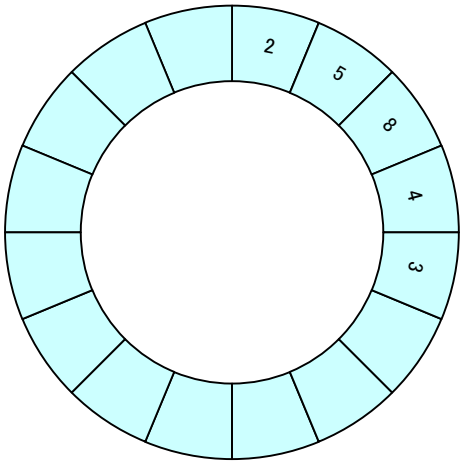

図 5: リングバッファー

3.2 実際の応用

これは，入れられた順序通りに処理すべきデータに使われる．たとえば，次のような応用がある．

- データをバッファーにためて，処理を行う場合．プリンターの処理などである．プリント待ちのデータをプリントキューと言うことがある．
- オンライントランザクション処理[1]の管理に用いれれる．この処理では，原則的には到着順に処理しなくてはならない．

3.3 キューの実装

配列を使えば，キューの実装も簡単である．教科書 [1] では，次のような構造体を宣言している．

```
typedef struct queue{
  int head;
  int tail;
  int size;
  int data[1];
}queue_t;
```

この構造体には，キューに必要な情報がすべてがある．図 6 に示すようにメンバー変数 head は，最新のデータを格納している配列の添え字を表している．tail はもっとも古いデータ—次に取り出すデータ—の次の配列の添え字を表していいる．size はこのスタックに蓄えることができるデータ数を表している．それは，配列の要素数に 1(番兵の分) を加えた値となっている．

[1]ネットワークに接続された複数のパソコンがホストコンピュータに処理要求を行い、ホストコンピュータがその要求にもとづいてデータを処理し、処理結果を即座にパソコンに送り返す処理方式。データベースの処理などに多く使われる．

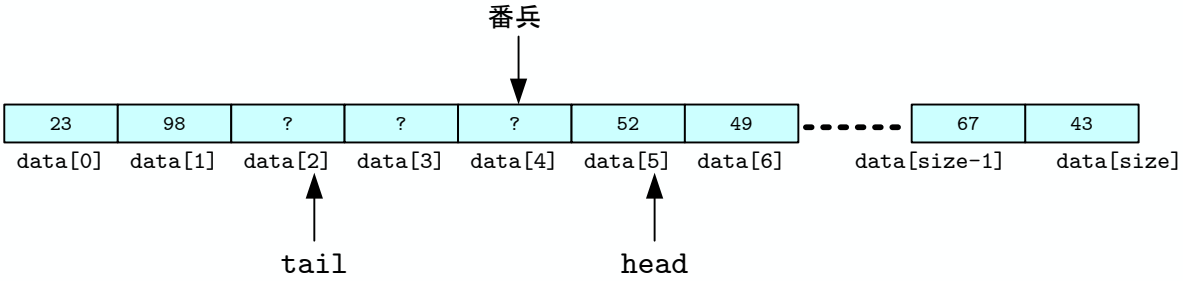

図 6: 教科書 [1] のキューのイメージ．配列中の?はデータはあるが，意味のない—使い道のない—データを表している．

キューを実際に使うためには記憶領域が確保と操作を行う必要がある．教科書の queue.c では，次の 3 つの関数でそれを行っている．記憶領域の確保を行う `creat_queue()` とデータを格納する `enqueue()`，データを取り出す `dequeue()` である．

4 プログラム作成の練習

[練習 1] 本日の講義ノートの WEB ページより,「stack.h」と「stack.c」,「RPN.c」をダウンロードしてコンパイルせよ．

```
gcc -o RPN stack.c RPN.c
```

このプログラムは 3 つのファイルからできており，逆ポーランド記法の電卓をシミュレートする．

- stack.h はヘッダーファイルで，複数のファイルで使用するものを宣言する．ここでは構造体の宣言やマクロ，プロトタイプ宣言を行っている．
- stack.c は，スタックを操作する関数の動作を記述している．
- 逆ポーランド記法の電卓プログラムの本体．

このように，ファイルを分けることにより，スタックを操作する関数が使いまわしできるようにしている．スタック操作する関数を他のプログラムで使いたければ,「stack.h」と「stack.c」をコピーし，一緒にコンパイルすればよい．

[練習 2] 逆ポーランド記法のプログラムを実行し，その動作を理解せよ．逆ポーランド記法については，教科書 [1] の p.189 に書いてある．HP 社の電卓は，逆ポーランド記法で入力する．

[練習 3] スタックを使って，キーボードから入力された整数を逆順に表示するプログラムを作成せよ.「stack.h」と「stack.c」をそのまま使うこと．

[**練習 4**] キューを使って，キーボードから入力された整数を入力された順序に表示するプログラムを作成せよ．ただし，本日の講義ノートの WEB ページより,「queue.h」と「queue.c」をダウンロードして使うこと．

5 課題

以下の課題を実施し，レポートとして提出すること．

[**問 1**] (復予) 教科書 [1] の pp.170–217 を 3 回読め．レポートには「3 回読んだ」と書け．

[**問 2**] (復) 本日配布したプリントを 2 回読め．レポートには「2 回読んだ」と書け．

[**問 3**] (復) スタックとキューについて，以下の操作を定義する．

PUSH(n) :スタックにデータ n をプッシュする．戻り値は無し．
POP() :スタックからデータをポップする．戻り値は，取り出した値．
ENQ(n) :キューにデータ n を追加する．戻り値はなし．
DEQ() :キューからデータを取り出す．戻り値は取り出した値．

空のスタックやキューに対して，以下の操作を行ったとき，データ構造の様子を図で示せ．

(1) PUSH(3)→PUSH(5)→POP()→PUSH(2)→PUSH(1)→POP()→POP()→PUSH(1)→POP()→PUSH(7)

(2) ENQ(6)→ENQ(2)→DEQ()→ENQ(7)→DEQ()→ENQ(3)→ENQ(1)→ENQ(2)→DEQ()→DEQ()

(3) PUSH(3)→ENQ(POP())→PUSH(8)→PUSH(5)→ENQ(POP())→PUSH(DEQ())→ENQ(2)

[**問 4**] (復) キーボードから整数を 10 個入力する．読み込んだ整数を入力順と逆順に表示するプログラムを作成せよ．ただし，教科書の「stack.h」と「stack.c」,「queue.h」と「queue.c」はそのまま使うとする．これらの教科書のプログラムは記述する必要はない．自分が書いたプログラムのみ書け．

[**問 5**] 前問のプログラムのコンパイル方法を示せ．

[**問 6**] 「stack.h」と「stack.c」では，どのようにしてスタックおよびその操作を行っているか? レポート用紙 1 枚程度にまとめよ．このプログラムを理解することが重要．

[**問 7**] 「queue.h」と「queue.c」では，どのようにしてキューおよびその操作を行っているか? レポート用紙 1 枚程度にまとめよ．このプログラムを理解することが重要．

[**問 8**] 逆ポーランド記法で書かれた，以下の計算結果を示せ．

(1) `12 45 +`

(2) `4 3 + 5 * 1 4 * / 3 2 + + 23 10 * 2 + 4 * - 8 +`

[**問 9**] (復) ここでの学習内容でわからないところがあれば，具体的に記述せよ．

提出要領はいつものとおり．ただし，期限は 7 月 17 日 (火)AM 8:45，課題名は「課題　スタックとキュー」とすること．

参考文献

[1] 内田智史監修, (株) システム計画研究所編. C 言語によるプログラミング　応用編　第 2 版. (株) オーム社, 2006.