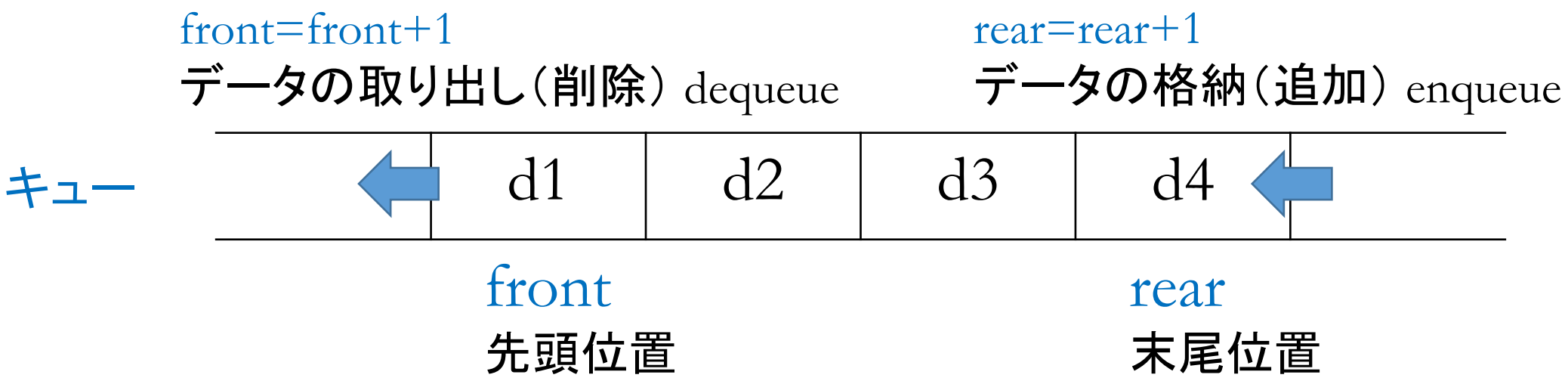
キュー

リングバッファ・連結リスト・優先度付きキュー

キュー（Queue）とは

- データを一時的に保存しておくためのもの
- キューは（順番を待つ人などの）行列で、待ち行列ともいう
- 待ち行列なので、データが列の一方の端から取り出され、もう一方の端から格納される
- すなわち、データは先入れ先出し（FIFO: First In First Out）の構造で保持する

キューの操作・用語


front=front+1
データの取り出し（削除） dequeue
rear=rear+1
データの格納（追加） enqueue
キュー
d1
d2
d3
d4
front
先頭位置
rear
末尾位置

キューの使用例

- 計算機OSのタスク管理（スケジューリング）に使用されている
- プリンタには印刷キューと呼ばれるものがある
  - 複数のPCから印刷指示（ジョブ）があった場合、ジョブをキューで順番に記憶し出力（印刷）する
- 予約システムのキャンセル待ち処理

実装方法と問題点

- 配列
  - データを取り出す/格納する度にfront/rear値に1足す
  - そのため、データの取り出しと格納を繰り返すと、前のほうに使えないメモリ領域がどんどん増えてしまいデータ格納ができなくなる
- リングバッファ
- 連結リスト
- 優先度付きキュー

リングバッファ

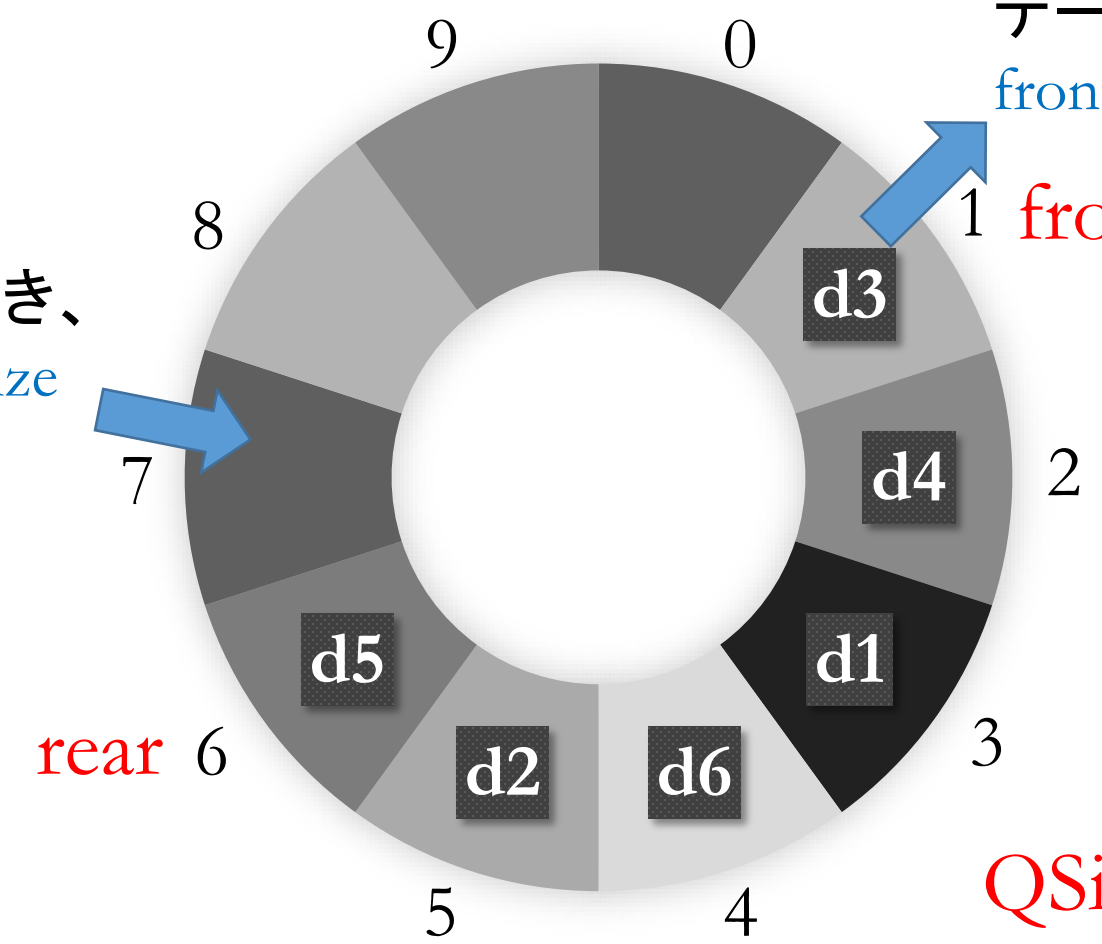
リングバッファ（Ring buffer, Circular queue）

- 配列の最後尾と先頭を繋げて「円状」の配列でキューを実装する方法
- 「円状」なので、front/rearの値が巡回するので、どんどん増えることがなく、待ち状態のデータの個数がキューのサイズを超えない限り、メモリが足りなくなる現象は発生しない

リングバッファ（Ring buffer, Circular queue）

データを取り出すとき、
front=(front+1) % QSize

データを格納するとき、
rear=(rear+1) % QSize



QSize=10

キューの実装（キューの関数⇒queue.h）

```
typedef  int data_t;

#include <stdlib.h>
#define QSize 1000
int queue[QSize], front, rear, qcnt;

void InitQueue(void)
{front=0; rear=-1; qcnt=0;}

int QueueEmpty(void)
{
  if(qcnt==0) return 1;
  else return 0;
}
```

```
void EnQueue(data_t x){
  if(qcnt==QSize) {puts("queue is full"); exit(-1);}
  qcnt++;
  rear=(rear+1) % QSize;
  queue[rear]=x;}

data_t DeQueue(void){
  data_t x;
  if(qcnt==0) {puts("queue is empty"); exit(-1);}
  x=queue[front];
  qcnt--;
  front=(front+1) % QSize;
  return x;}
```

manaba小テスト：06-1

- 8分
- 8点

manaba小テスト：06-2

- 8分
- 10点

manaba小テスト:06-3

- 12分
- 12点

キューを用いたプログラミング

- グラフの幅優先探索（2Qで紹介）
- 次のプログラム

キューを利用したプログラムの作成

- 文字列を入力したら空白で単語を検出しそれを括弧付けて出力してくれるプログラムを作成したい。

  例：This is a pen → [This][is][a][pen]

考え方・手順

1. 1行の文字列を読み込む
2. 読み込んだ文字列の先頭から一文字ずつ以下の処理を行なう
   2.1 空白と改行でなければ文字をエンキューする。
   2.2 そうでなければ，キューが空になるまでデキュー・出力し続ける。ただし、その前後に‘[’と‘]’を出力する

プログラム

```
#include <stdio.h>
#include "queue.h"

#define N 128

int main(void)
{
  char line[N];
  int i;

  InitQueue();
  while(fgets(line, N, stdin)!=NULL){
    for(i=0; line[i] != '¥0'; i++){
      if(line[i] != ' ' && line[i] != '¥n')
        ?1;
      else{
        putchar('[');
        while(!QueueEmpty())
          putchar(?2);
        putchar(']');}
    }
    printf("¥n");
  }
  return 0;
}
```

manaba小テスト：06-4

- 5分
- 4点