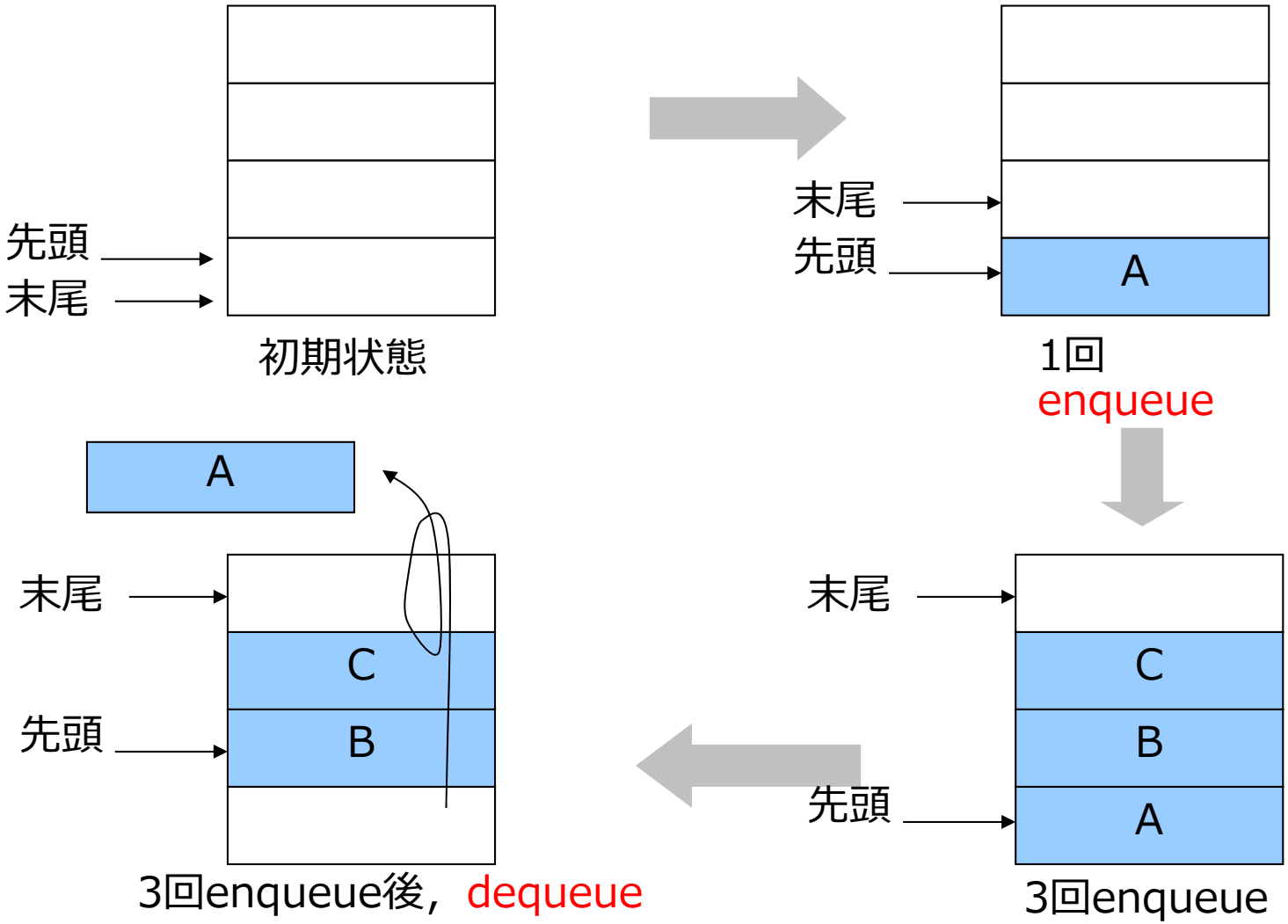
第7週
配列を使ったキュー

キューとは

- データを保持するデータ構造
  - 先入れ先だし (First In First Out: FIFO)
  - スタックとは取り出す順番が反対

- キューに関する変数
  - 末尾: データの挿入口を指す
  - 先頭: データの出口を指す

キューの操作


先頭
末尾
初期状態
末尾
先頭
A
1回
enqueue
末尾
C
B
先頭
A
3回enqueue
A
末尾
C
先頭
B
3回enqueue後，dequeue

配列によるキューの実現

キューの入口・出口をしっかり管理すること！

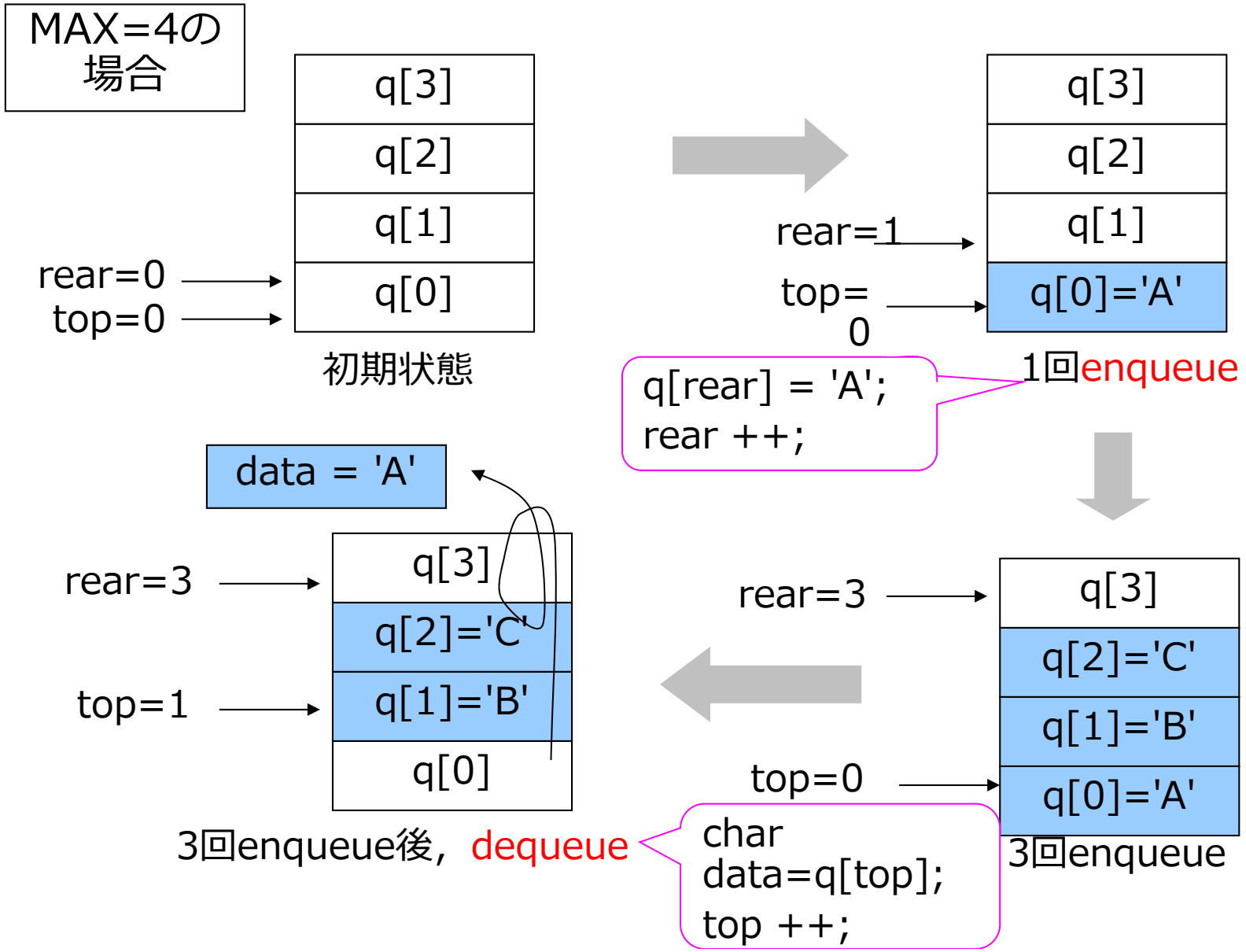
- **REAR**：入口を表す変数
  - 次にenqueueされた値が格納される添え字
    - REARの位置には何も入っていない
- **TOP**：出口を表す変数
  - 最も古い値が格納されている添え字

配列によるキューの表現

- 文字型データ (char型) を積んでいくキューを考える
- char q[MAX];　　キュー用配列
- int top;　　キュー先頭(データ出口)
- int rear;　　キュー末尾(データ入り口)

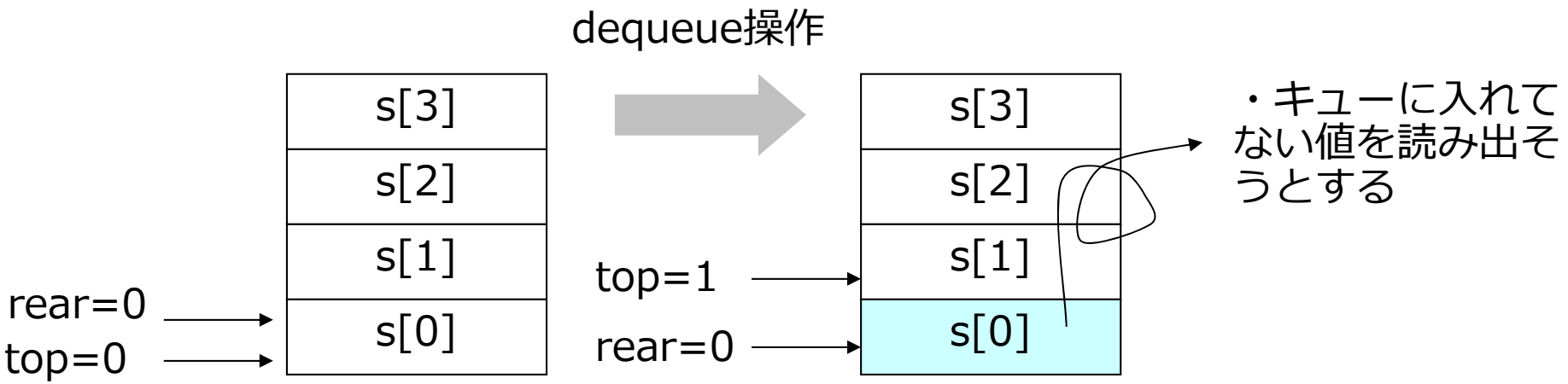
- 初期化： top=0, rear=0
- enqueue操作: q[rear] = data, rear++;
- dequeue操作: data = q[top], top++;

配列を使ったキュー操作: 必須課題7-2,3



dequeue操作時の注意

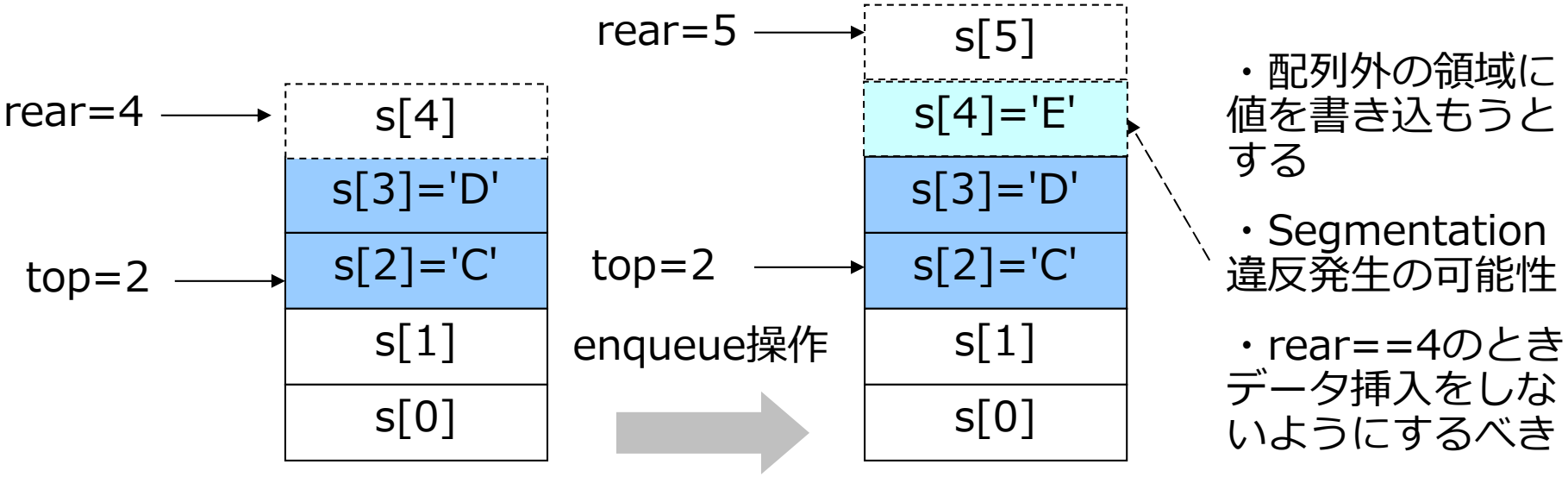
●空のキューから取り出さないようにする



以降，余力のある場合

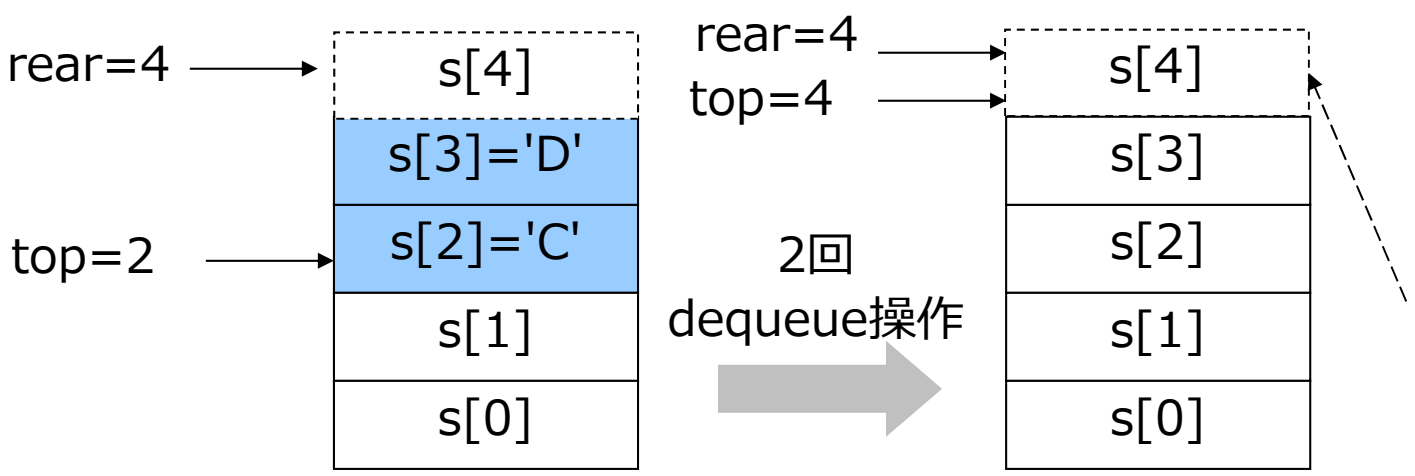
enqueue操作時の注意

●rearの値をチェックし，配列外の領域に入らないようにする



リングバッファ

●配列によるキューの勿体無い使い方．．．



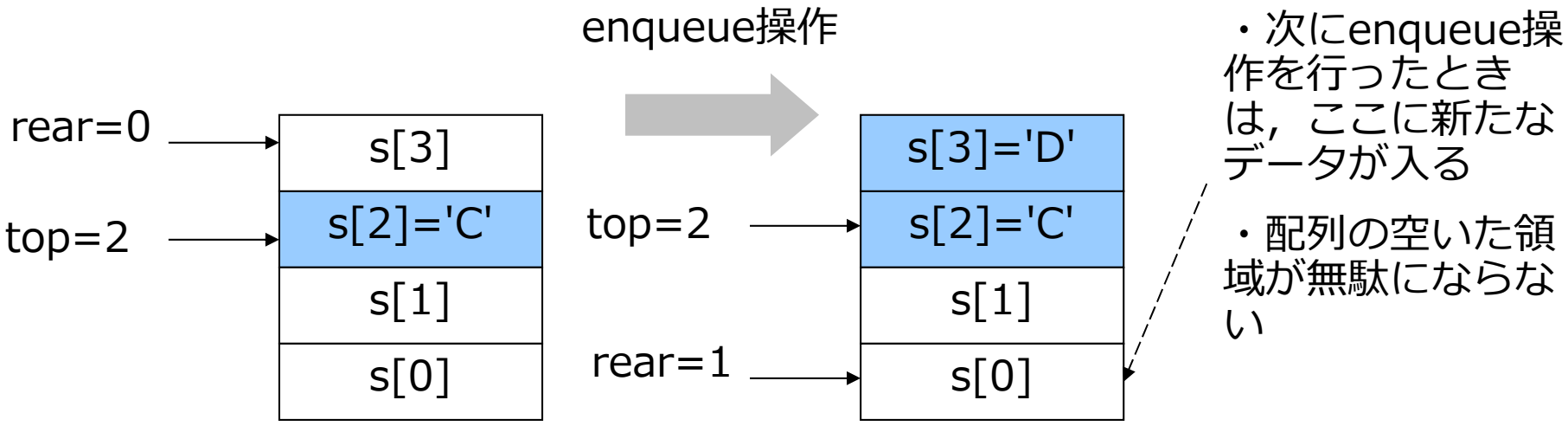
・配列外の領域への書き込みを避けるため，データ挿入を不可能にする必要

・しかし実際には，s[0]からs[3]は何のデータも入っていない．ここに新たなデータを挿入できるのでは？

リングバッファ

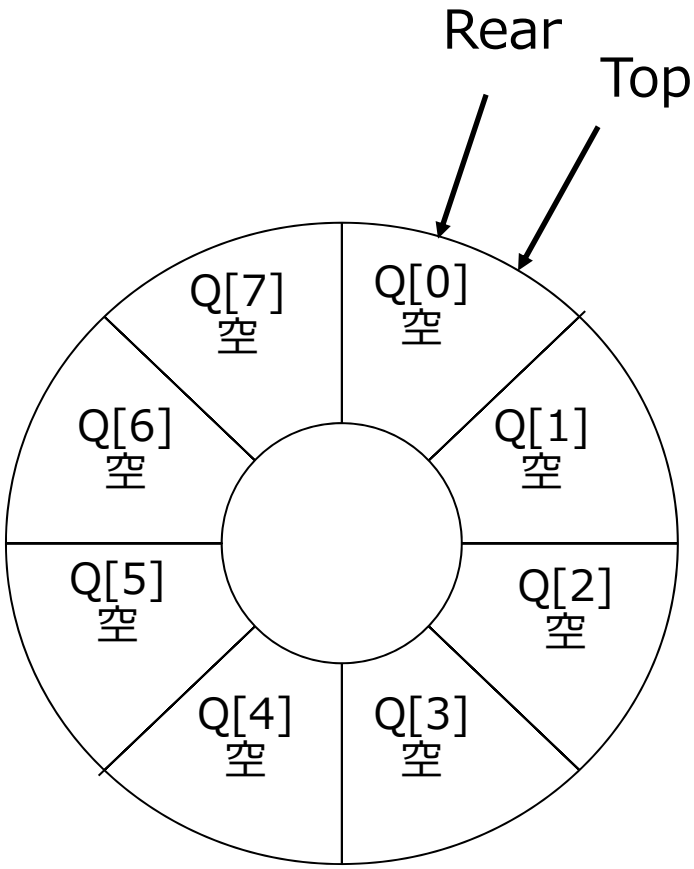
●配列に空きがあるのにenqueueできない問題の回避

- top, rear=3の次は0に戻す

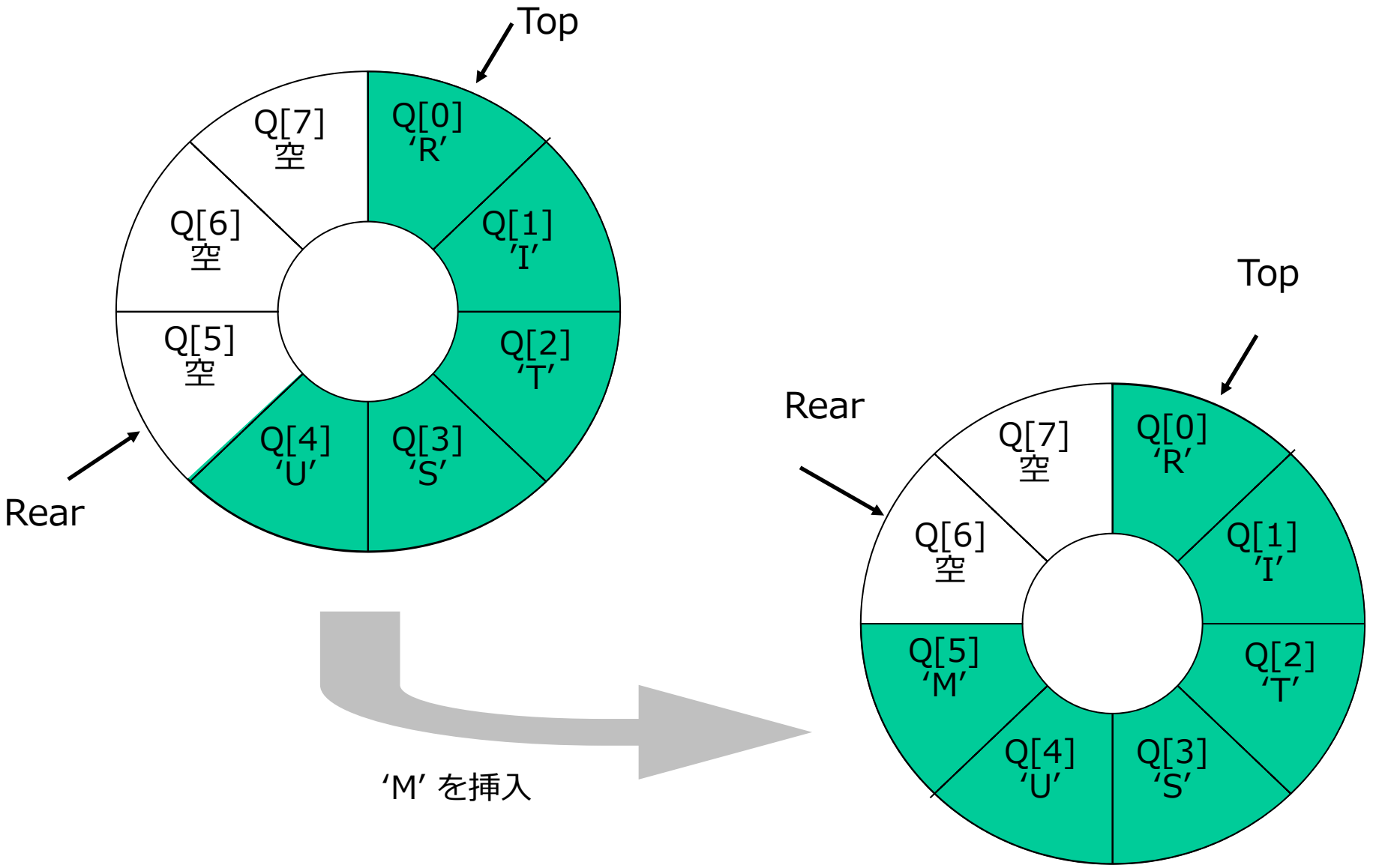


キューによるリングバッファの初期状態

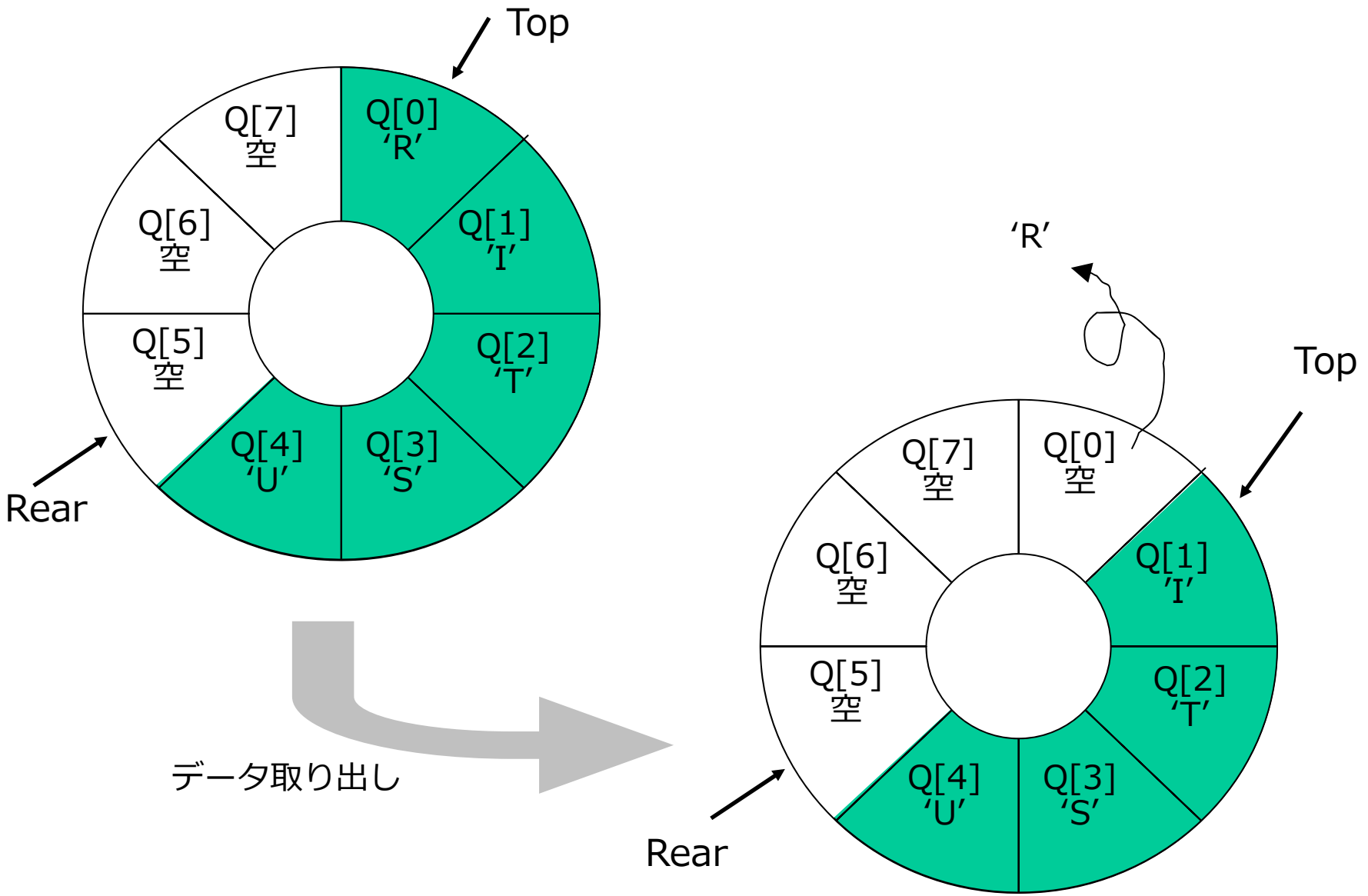
●キューの大きさMAX=8のときの例



データ挿入:Enqueue

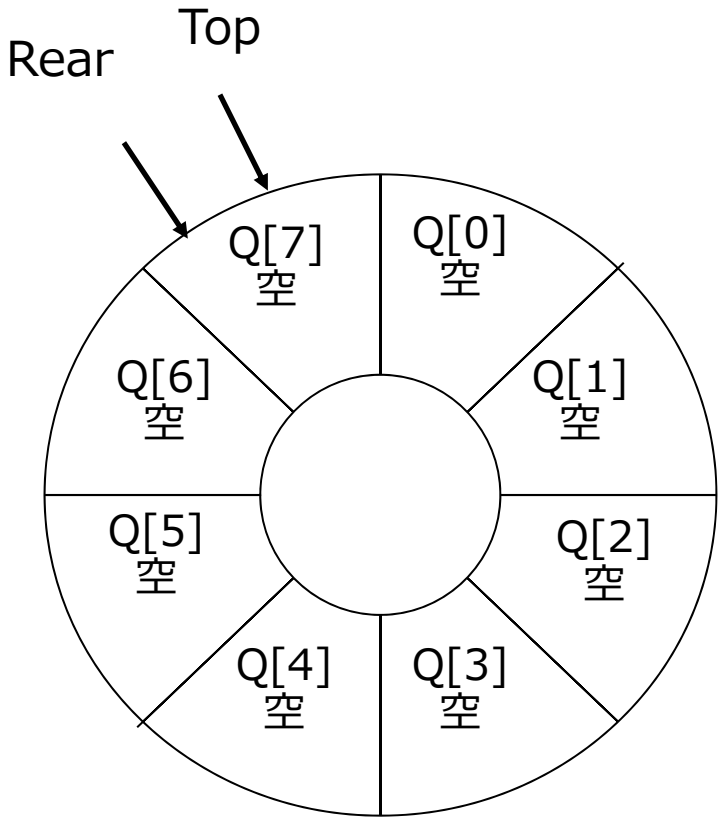


データ取り出し:Dequeue

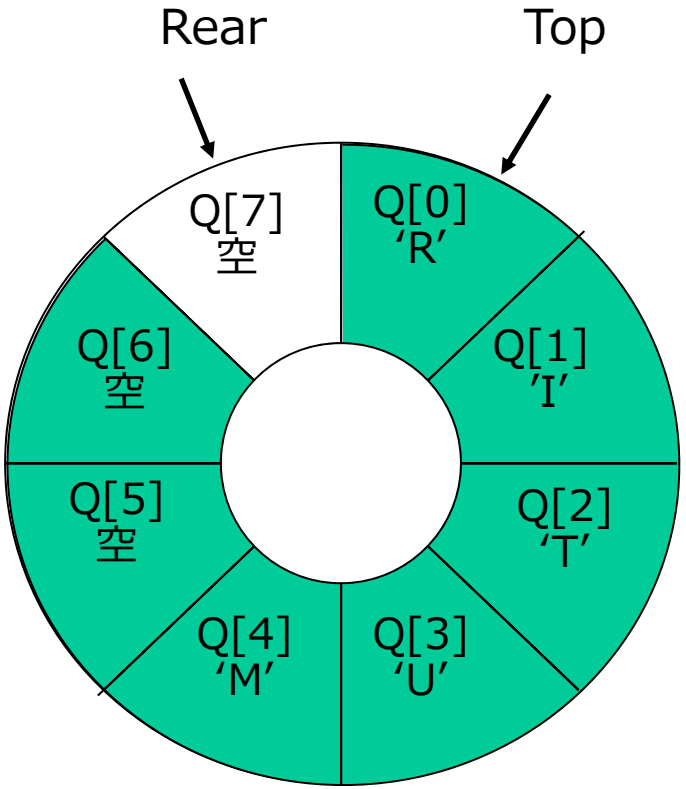


キューの空/満杯の判定

- Top, rearの値だけで空と満杯を区別するため，常に一つの要素は空けておく



空



満杯

参考プログラム

```
#define MAX 256 /* 配列の大きさ*/

int main(void){
   char  q [MAX] ; /* キュー配列*/
   int top = 0;
   int rear = 0;

   char data; /* キューから取り出した値*/

   enqueue (‘A’, q, &top, &rear);      /* enqueue操作*/
   data = dequeue(q, &top, &rear); /* dequeue操作*/
   print_queue_ary(q, top, rear);    /* キューの内容を出力*/
}
```

著者リスト

1. 安積 卓也（情報システム学科）
2. 泉　朋子　（情報コミュニケーション学科）
3. 原田 史子（情報システム学科）