

課題 4

アルゴリズムとデータ構造B

第10回

課題 4

- レポート提出期限：**2024年11月13日（水） 17:00**

データ構造

- 配列 `int que[MAX];`
 - 先頭, 末尾 `int front, rear;`
 - 要素数 `int num;`
- } グローバル変数
(`front`, `rear`, `num` は 0 で初期化)

アルゴリズム

- データの入力 `int Enque(int x);`
 - ✓ 末尾の位置にデータを格納する
 - ✓ 末尾を 1 つずらす（進める）, 要素数を 1 つ **増やす**
 - ✓ `rear` が末尾まで行ったら先頭に戻す
- データの出力 `int Deque();`
 - ✓ 先頭の位置からデータを取り出す
 - ✓ 先頭を 1 つずらす（進める）, 要素数を 1 つ **減らす**
 - ✓ `front` が末尾まで行ったら先頭に戻す

Enque のアルゴリズムの確認

3

```
int Enque(int x) {  
    if (num < MAX) {  
        que[rear++] = x;  
        if (rear == MAX)  
            rear = 0;  
        num++;  
    } else  
        return -1;  
  
    return 0;  
}
```

方針：入れてからずらす
キューが full のときは -1 を返す
rear が末尾に到達したとき先頭に戻る

Deque のアルゴリズムの確認

4

```
int Dequeue() {  
    if (num > 0) {  
        int tmp = que[front++];  
        if (front == MAX)  
            front = 0;  
        num--;  
        return tmp;  
    } else  
        return -1;  
}
```

方針：出してからずらす

キューがemptyのときは -1 を返す

front が末尾に到達したとき先頭に戻る

前回の実装

- 配列
`int que[MAX];`
- 先頭, 末尾
`int front, rear;`
- 要素数
`int num;`



今回の実装

```
struct Que {  
    ...; // 配列  
    ...; // 先頭  
    ...; // 末尾  
    ...; // 要素数  
};
```

関数も構造体に対応させる

- 参考) 初期化

```
void Initialize(struct Que *q) {  
    ... = 0;  
    ... = 0;  
    ... = 0;  
    for (int i=0; i<MAX; i++)  
        q->que[i] = 0; // 0で初期化しとく  
}
```

```
void Display() {  
    int i;  
    for (i=0; i<MAX; i++) {  
        if (i == front)  
            printf(" front->");  
        else  
            printf("      ");  
        printf("%6d: %6d", i, que[i]);  
        if (i == rear)  
            printf(" <-rear¥n");  
        else  
            printf("      ¥n");  
    }  
}
```


- 課題 4 の重要な部分は問題 1 と問題 2 で，問題 3 は複数のキューを扱う例題
- `scanf` を繰り返し行うようにし，その回数をカウントする
- カウントに応じて `que1` に `Enque` するか `que2` に `Enque` するかを分岐させる
- キューから値を全て取り出すためには，
キューが空になる（`num` が 0 になる）まで，もしくは，
キューが空の場合に `Deque` すると `-1` が返ってくることを利用して
`-1` が返ってくるまで `Deque` を続けると良い

配布プリントの問題を解き（プログラムを作成し）、
期限までにレポートを Teams 上で提出すること

- レポート提出期限：2024年11月13日（水） 17:00