

課題 4

アルゴリズムとデータ構造B

第10回

本日の予定

課題 4

- レポート提出期限：**2024年11月13日（水）17:00**

キューのプログラミング（基本）

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

```
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す

Deque のアルゴリズムの確認

```
int Deque() {
    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す

キューを構造体で

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  
}
```

Display (参考)

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

問題 3

- 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 `Dequeue` 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 `Dequeue` を続けると良い

課題 4

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