

<자료구조 2차 프로그래밍 보고서>

201720736 이균

1. 문제 이해

Circular queue를 활용하여 사용자로부터 입력을 받아 대기자를 큐에 집어넣고, 진료를 받으면 큐에서 빼게 하는 큐를 사용하는 문제이다.

입력은 맨 처음 서비스를 고르는 입력으로 1,2,3 세 개의 케이스가 존재하고, 서비스 입력에서 1을 입력하면 또다시 대기자등록(1) 또는 취소(2)를 고를 수 있다.

대기자 등록 시에는 대기자를 큐에 집어넣으면 된다.

또한 서비스 입력에서 2를 입력 받으면 진료를 완료하고 다음번에 대기하고 있는 사람을 출력한다.

마지막으로 서비스 입력으로 3이 입력되면 프로그램을 종료하며 프로그램은 3이 입력되기 전까지 반복된다.

2. 함수 구현

```
void addq(element item)
{
    rear = (rear + 1) % Max_Queue_Size;
    if (front == rear)
        queueFull();
    queue[rear] = item;
    qcnt++;
    if (queueclear)
        queueclear = FALSE;
}

element deleteq()
{
    if (front == rear)
    {
        queueclear = FALSE;
        return queueEmpty();
    }
    front = (front + 1) % Max_Queue_Size;
    qcnt--;
    if (qcnt == 0)
        queueclear = TRUE;
    return queue[front];
}
```

우선 큐를 사용하기 때문에 교재에 있는 addq함수와 deletq함수를 이용하였다.

출력으로 대기자가 몇 명인지 알기 위해서 qcnt변수를 선언하여 addq호출시 증가, deletq호출 시 감소되게 하여 대기자 명수를 체크하도록 하였다.

또한 queueclear변수는 대기자가 존재하다가 진료 완료되었을 경우에 스택이 비워지는 경우와 맨 처음 스택이 비워진 상태에서 사용자가 2번을 입력했을 때의 경우를 부군하기 위해서 사용하였다.

따라서 맨 처음 입력으로 2번이 입력되면 front == rear이기 때문에 바로 queueEmpty가 실행되고,

대기자가 들어왔다가 대기자가 진료를 받아 큐가 비워진 경우에는 queueclear가 TRUE가 되어서

추후의 출력에서 케이스를 구분해주었다.

```
void queueFull()
{
    if (rear != 0)
    {
        rear -= 1;
    }
    else
        rear = Max_Queue_Size - 1;

    printf("큐가 꽉 찼습니다.\n");
}

element queueEmpty()
{
    printf("대기중인 환자가 없습니다.\n");
    printf("감사합니다.\n");
    printf("-----\n");
    return queue[rear];
}
```

다음은 queue가 꽉 차거나 비워졌을 때 실행되는 함수이다.

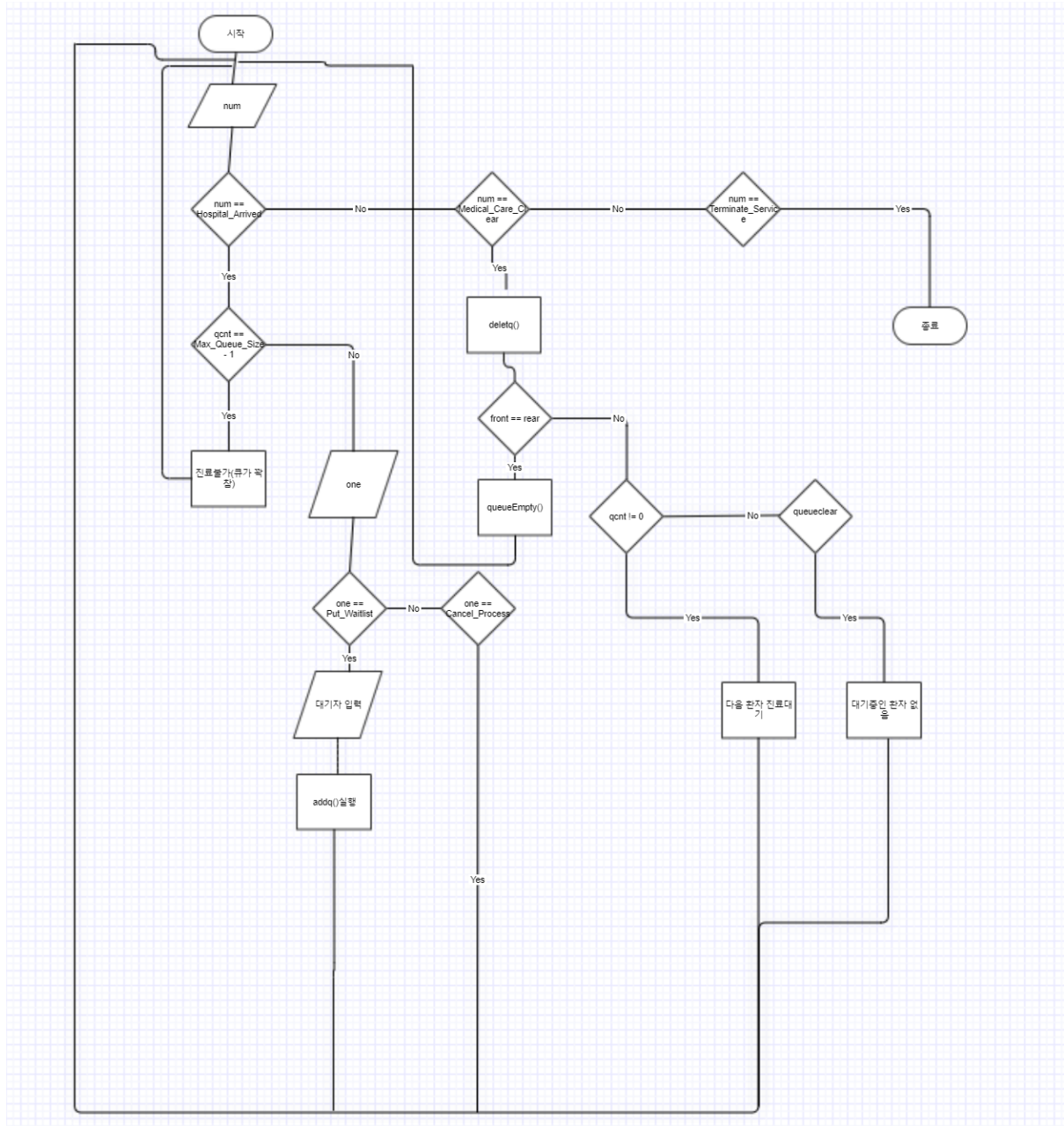
우선 queueFull은 큐가 꽉 찼기 때문에 addq에서 증가된 rear을 되돌리기 위해서 circular queue의 특성상 rear가 맨 마지막에서 1증가 시 0이 되기 때문에 케이스를 rear가 0이 아닐 때와 0일 때로 나누어서 다시 rear을 전의 값으로 초기화 시켜준다. 다만 이 부분은 과제 명세서에 5명이 되면 바로 새 진료가 불가능하다는 메시지를 출력하라고 했기 때문에 실제 코드에서는

qcnt == Max_Queue_Size - 1 조건을 이용하였고, queueFull은 학습목적으로 구현해보았다.

Circular queue이기 때문에 큐가 꽉 차 있다는 것은 큐에 비워진 한 개 빼고는 다 차있다는 의미이다. 따라서 qcnt변수가 Max_Queue_Size - 1이면 front와 rear의 비교를 생각하지 않아도 큐가 꽉 차있는

것이 된다. 다음으로 queueEmpty는 큐가 비워졌을 때 대기중인 환자가 없다고 출력하고 있다.

2. Flowchart



사용자가 입력으로 병원도착(1)을 입력한 경우에는 큐가 꽉 찼을 경우에는 “새 진료 접수가 불가능 합니다.” 라는 출력을 한 뒤에 다시 서비스 입력으로 돌아간다. 반면 큐가 비워져 있거나 남아있는 경우에 대기자 입력을 원할 시에 새로운 입력을 받고, 큐에 집어넣는다. 또한 큐가 비워져 있거나 남아있

는 경우에 취소를 원하면 취소 후 다시 서비스 입력으로 돌아간다.

서비스 입력에서 진료완료(2)를 입력 받으면 만약 맨 처음 입력이라면 queueEmpty가 실행이 되어 대기중인 환자가 없다고 출력이 되고, 만약 사용자를 입력 받아 qcnt가 증가된 상태여서 0이 아닌 경우 진료 완료를 입력한 경우에는 다음에 기다리고있는 대기자를 출력시킨다. 또한 큐에 맨 마지막 대기자가 있었는데 진료완료를 누르면 deleteq가 실행되어 qcnt가 0이 되고, 이때에는 queueclear가 1이 되기 때문에 또다시 queueEmpty가 실행된다.

마지막으로 사용자 입력에서 3이 입력되면 프로그램이 종료된다.