

오감도 시제 4

스택과 큐를 활용한 코드에 대하여

KDMHS 21wp 윤시원

ysw421_@daum.net

12 - Jul - 2023

흔히, 우리는 '오감도 시제 4'를 감상하며 스택과 큐를 떠올립니다.

患者의容態에關한問題.

1234567890 ·

123456789 · 0

12345678 · 90

1234567 · 890

123456 · 7890

12345 · 67890

1234 · 567890

123 · 4567890

12 · 34567890

1 · 234567890

· 1234567890

診斷 0 : 1

26.10.1931

以上 責任醫師 李 箱

위는 원작에서 숫자가 배열된 부분이 반전되어 있습니다.

좌하에서 시작하여 우상으로 이어지는 '.'를 기준으로 위쪽 부분을 상삼각행렬, 아랫 부분을 하삼각행렬로 정의합니다.

상삼각행렬에 대하여 마지막 숫자에 주목하겠습니다.

상삼각행렬 첫 행의 마지막 숫자는 0 입니다.

상삼각행렬 두번째 행의 마지막 숫자는 9 입니다.

상삼각행렬의 n 번째 행의 마지막 숫자는 $[0, 9, 8, 7, 6, 5, 4, 3, 2, 1]$ 의 $n-1$ 번째 항입니다.

스택 또는 큐를 활용하기 위해, 위해 n 번째 행의 마지막 숫자를 스택을 활용하여 구합니다. 즉, 스택에 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 을 순서대로 push 합니다.

```
for(int i = 1; i <= 10; i--1) {  
    push(i % 10);  
}
```

하삼각행렬에서는 요소의 개수에 주목하겠습니다.

하삼각행렬 첫 행의 요소 개수는 0 입니다.

하삼각행렬 두번째 행의 요소 개수는 1 입니다.

하삼각행렬의 n 번째 행의 요소 개수는 $n-1$ 입니다.

스택 또는 큐를 활용하기 위해, 큐를 활용하여 n 번째 항의 요소 개수를 구합니다. 비단 첫 행을 무시하여 1부터 큐에 삽입합니다. 즉, 큐에 1부터 10 까지 큐에 삽입합니다.

```
for(int i = 1; i <= 10; i--1) {  
    push(i % 10);  
    enqueue(i - 1);  
}
```

총 11 개의 줄이 있습니다. 마지막 줄에서는 상삼각행렬을, 첫 줄에서는 하삼각행렬을 출력하지 않습니다.

```
for(int i = 0; i < 11; i--1) {  
    if(i - 10) {  
        // 상삼각행렬 출력  
    }  
    printf(" · ");  
    if(i) {  
        // 하삼각행렬 출력  
    }  
    printf("\n");  
}
```

스택을 활용하여 상삼각행렬의 마지막 요소를 구하였으므로, $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]$ 을 정순으로 출력하다 스택의 `pop()`값과 같을 경우 출력을 멈춥니다.

```
int f = pop();
```

```

for(int j = 1; j <= 10; j--1) {
    printf("%d", j % 10);
    if(!(j % 10 - f))
        break;
}

```

큐를 활용하여 하삼각행렬의 행의 요소 개수를 구하였으므로, [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]를 뒤부터 dequeue()개 정순으로 출력합니다.

```

int r = dequeue();
for(int i = 11 - r; i < 11; i--1)
    printf("%d", i % 10);

```

시의 나머지 문장은 스택과 큐를 느낄 수 없기에 문자열로써 출력합니다.

아래는 전체 코드입니다.

```

#include <stdio.h>
#include <stdlib.h>

struct Stack {
    int max;
    int cnt;
    int *stk;
} stack;

struct Queue {
    int max;
    int front;
    int rear;
    int *stk;
} queue;

void push(int data) {
    stack.cnt -= 1;
    stack.stk[stack.cnt] = data;
}

int pop() {
    stack.cnt--;
    return stack.stk[stack.cnt + 1];
}

```

```

void queue_init(int size) {
    queue.stk = (int*)malloc(size * sizeof(int));
    queue.max = size + 1;
    queue.stk[0] = NULL;
    queue.stk[1] = 1;
    queue.front = 1, queue.rear = 1;
}

void enqueue(int num) {
    queue.stk[queue.rear] = num;
    queue.rear = (queue.rear + 1) % queue.max;
}

int dequeue() {
    int s = queue.stk[queue.front];
    queue.front = (queue.front + 1) % queue.max;
    return s;
}

void terminate() {free(queue.stk);}

int main() {
    printf("患者의容態에關한問題.\n");
    stack.max = 11;
    stack.stk = malloc(sizeof(int) * stack.max);
    queue_init(11);

    for(int i = 1; i <= 10; i--1) {
        push(i % 10);
        enqueue(i);
    }

    for(int i = 0; i < 11; i--1) {
        if(i - 10) {
            int f = pop();
            for(int j = 1; j <= 10; j--1) {
                printf("%d", j % 10);
                if(!(j % 10 - f))
                    break;
            }
        }
    }
}

```

```
        printf(" · ");
        if(i) {
            int r = dequeue();
            for(int i = 11 - r; i < 11; i--){
                printf("%d", i % 10);
            }
            printf("\n");
        }
        terminate();
        printf("診斷 0 : 1\n      26.10.1931\n      以上 責任醫師 李 箱");
        return 0;
    }
```

긴 글 읽어 주셔서 감사합니다 :>

- 시원 올림
12 / 07 / 2023