

Week2_과제

[통계](#) [수정](#) [삭제](#)

yuyu7123 · 약 5시간 전 · 비공개

 0

2696 우선순위 큐

Smarcle_23Algorithm



▼ 목록 보기

3/3



Baekjoon_2696 : 중앙값 구하기

우선순위 큐를 이용하여 수열을 삽입하고 홀수번째마다 정렬 한 후 중앙값 출력

풀이 과정

1. 큐 초기화
2. 큐에 수열 원소 삽입
3. 정렬 후, 반복문을 통해 홀수값마다 중앙값 출력
4. 위의 과정 n번 반복

문제 해결

- 두 번째 수열부터 값이 제대로 출력이 안됨
선택정렬함수 내부에 출력문을 main함수로 옮김으로 해결
- 시간 초과
출력을 따로 for문으로 구현하였는데 입력하는 부분으로 옮겨 for문을 하나 제거 ->
시간초과 조금 나아졌지만 해결X
선택정렬에서 퀵정렬로 변경 -> 시간초과 조금 나아졌지만 해결X
선택정렬에서 삽입정렬로 변경 -> 해결

코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#define MAX_QUEUE_SIZE 10000

// 구조체 선언
typedef int element;
typedef struct {
    int back;
    int count;
    element data[MAX_QUEUE_SIZE];
} QueueType;

// 큐 초기화
void init_queue(QueueType* q) {
    q->back = -1;
    q->count = 0;
}

// 정수 item을 큐에 넣는 연산
void push(QueueType* q, int item) {
    q->back = (q->back + 1) % MAX_QUEUE_SIZE;
    q->data[q->back] = item;
    q->count++;
}

// 삽입 정렬
void insertionSort(QueueType* q, int n) {
    int i, j, key;
    for (i = 1; i < n; i++) {
        key = q->data[i];
        j = i - 1;

        while (j >= 0 && q->data[j] > key) {
            q->data[j + 1] = q->data[j];
            j = j - 1;
        }
        q->data[j + 1] = key;
    }
}

int main() {

    int n, m, l, cnt = 0;
    QueueType q;

    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        init_queue(&q);
        scanf("%d", &m);
        cnt = 0;

        for (int j = 0; j < m; j++) {
            scanf("%d", &l);
            push(&q, l);
        }
    }
}
```

```
    }
    if (m % 2 == 0) printf("%d\n", m/2);
    else printf("%d\n", m / 2 + 1);

    for (int j = 0; j < m; j++) {
        if (j % 2 == 0) {
            insertionSort(&q, j + 1);
            printf("%d ", q.data[j / 2]);
            cnt++;
            if (cnt % 10 == 0) printf("\n");
        }
    }
    printf("\n");
}

return 0;
}
```



유정



이전 포스트
Week1_스터디

0개의 댓글

댓글을 작성하세요

댓글 작성

