[백준] 11650번: 좌표 정렬하기

yoool · 방금 전 · 비공개

> 11650번: 좌표 정렬하기
> 2차원 평면 위의 점 N개가 주어진다. 좌표를 x좌표가 증가하는 순으로, x좌표가 같으면 y좌표가 증가하는 순서로 정렬한 다음 출력하는 프로그램을 작성하시오.

분할통치법

- 분할(divide), 재귀(recur), 통치(conquer)를 사용
- 분할통치법을 사용하는 정렬 방법 : 합병 정렬, 퀵 정렬

합병 정렬(merge-sort)

1. 무순리스트 L을 n/2개의 원소를 가진 두 개의 부리스트 L_1과 L_2로 분할
2. L_1, L_2를 재귀적으로 정렬
3. L_1, L_2를 단일 순서리스트로 합병

퀵 정렬(quick-sort)

1. 기준원소 pivot 설정 후 리스트 L을 LT(pivot보다 작은 원소들), EQ(pivot과 같은 원소들), GT(pivot보다 큰 원소들)의 세 부분으로 분할
2. LT와 GT를 재귀적으로 정렬
3. LT, EQ, GT를 결합

합병 정렬과 퀵 정렬 비교

	합병 정렬	퀵 정렬
기법	분할통치법	분할통치법
실행시간	$O(n \log n)$ 최악실행시간	$O(n^2)$ 최악실행시간 $O(n \log n)$ 기대실행시간
분할 vs. 결합	분할은 쉽고, 합병은 어렵다	분할은 어렵고, 합병은 쉽다
제자리 구현	제자리 합병이 어렵다	제자리 분할이 쉽다
실제 작업 순서	작은 것에서 점점 큰 부문제로 진행	큰 것에서 점점 작은 부문제로 진행

중앙값 구하기를 할 때 퀵 정렬을 사용해 본 적이 있기 때문에 이번에는 합병 정렬을 이용해 문제를 풀었다. 알고리즘 강의를 들으며 전반적인 개념에 대해 이해를 하고 의사 코드를 보며 코드를 작성했다. 잘 작성했다고 생각했는데 계속해서 오류가 나 완전히 다시 코드를 작성했는데 알고 보니 m+1을 m+l로 작성해서 나타난 오류였다.... 두 개의 함수 중에서 하나만 오타가 났던 거라 수정을 하니 바로 실행이 됐다!

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

typedef struct {
    int x;
    int y;
} coor;

coor B[100001];

void rMergeSort(coor* A, int l, int r);
void merge(coor* A, int l, int m, int r);

int main() {

    int N, i;

    scanf("%d", &N);

    coor* A = (coor*)malloc((N + 1) * sizeof(coor));

    for (i = 0; i < N; i++)
        scanf("%d %d", &A[i].x, &A[i].y);

    rMergeSort(A, 0, N - 1);
    //mergesort(A, 0, N - 1);

    for (i = 0; i < N; i++)
```

```
            printf("%d %d\n", A[i].x, A[i].y);

        return 0;
    }

    void rMergeSort(coor* A, int l, int r) {
        int m;
        if (l < r) {
            m = (l + r) / 2;
            rMergeSort(A, l, m);
            rMergeSort(A, m + 1, r);
            merge(A, l, m, r);
        }
    }

    void merge(coor* A, int l, int m, int r) {

        int i = l, k = l, j = m + 1;

        while (i <= m && j <= r) {
            if (A[i].x < A[j].x)
                B[k++] = A[i++];
            else if (A[i].x > A[j].x)
                B[k++] = A[j++];
            else if (A[i].x == A[j].x) {
                if (A[i].y < A[j].y)
                    B[k++] = A[i++];
                else if (A[i].y > A[j].y)
                    B[k++] = A[j++];
            }
        }

        while (i <= m)
            B[k++] = A[i++];

        while (j <= r)
            B[k++] = A[j++];

        for (k = l; k <= r; k++)
            A[k] = B[k];

    }
```

김지율

이전 포스트
[백준] 2696번: 중앙값 구하기