

Week4_과제

[통계](#) [수정](#) [삭제](#)

yuyu7123 · 방금 전 · 비공개

 0

11650 퀵정렬 합병정렬

Smarcle_23Algorithm



▼ 목록 보기

5/5



Baekjoon_11650 : 좌표 정렬하기

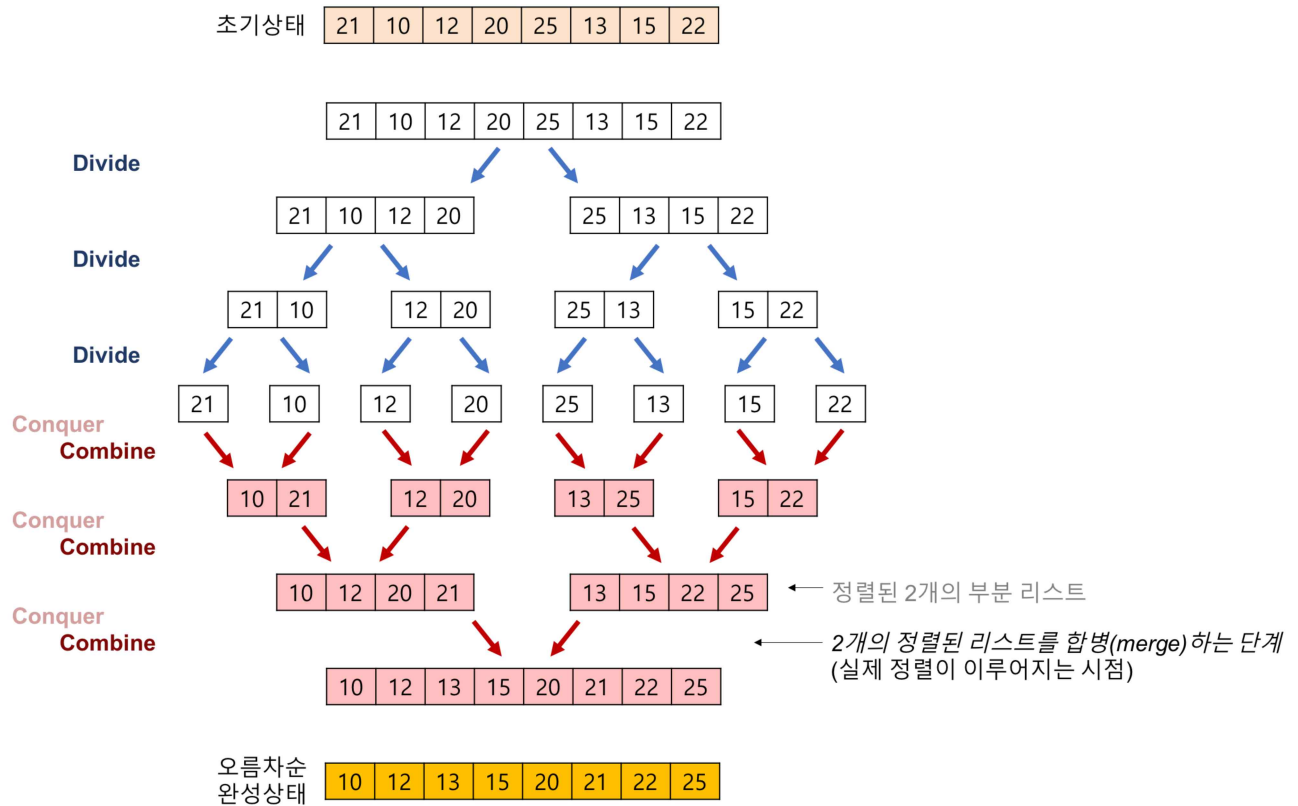
N이 100,000개까지 주어질 수 있으므로 값이 많아도 안정적인 시간을 가지는 **합병정렬**을 사용
합병정렬: 입력된 배열을 최대한 작은 단위로 나눈 후 하나씩 다시 정렬하는 방법

풀이 과정

1. 기본적인 합병정렬의 분할과 병합 코드를 작성
2. 구조체 인덱스 i에 i번째 x, y 좌표 입력받기
3. 재귀적으로 정렬 함수를 호출하여 최대한 작은 단위로 쪼개기
4. 합병함수로 left, mid, right에 대한 정보를 전달하여 좌우 배열의 값을 비교하여 작은 값을 배열에 저장

문제 해결

- 합병정렬의 이해



그림출처

- 런타임 에러가 발생

학교 알고리즘 수업의 강의자료에 있는 의사코드를 참고하였는데 보조배열 b의 동적할당 위치의 문제인 것 같아 merge -> mergeSort -> main 함수로 위치 변경, main 함수내에서 에러 없이 잘 작동

코드

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

struct Point {
    int x, y;
};

void merge(struct Point a[], struct Point b[], int left, int mid, int right) {
    int i = left, k = left;
    int j = mid + 1;

    while (i <= mid && j <= right) {
        if (a[i].x < a[j].x)
            b[k++] = a[i++];
        else if (a[i].x == a[j].x && a[i].y < a[j].y)
            b[k++] = a[i++];
        else
            b[k++] = a[j++];
    }
    while (i <= mid)
        b[k++] = a[i++];
    while (j <= right)
        b[k++] = a[j++];
    for (k = left; k <= right; k++)
```

```
        a[k] = b[k];
    }

void mergeSort(struct Point a[], struct Point b[], int left, int right) {
    if (left < right) {
        int mid = (right + left) / 2;
        mergeSort(a, b, left, mid);
        mergeSort(a, b, mid + 1, right);
        merge(a, b, left, mid, right);
    }
}

int main() {
    int N;
    scanf("%d", &N);
    struct Point* a = (struct Point*)malloc(N * sizeof(struct Point));
    struct Point* b = (struct Point*)malloc(N * sizeof(struct Point));

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

    mergeSort(a, b, 0, N-1);

    for (int i = 0; i < N; i++) {
        printf("%d %d\n", a[i].x, a[i].y);
    }

    free(a);
    free(b);

    return 0;
}
```



유정



이전 포스트
Week3_과제

0개의 댓글

댓글을 작성하세요

댓글 작성

