

자료 구조와 알고리즘

Term Project

2016112609 산업시스템공학과

김동현

- 먼저 저번 Term Project와 제출했을 때의 다른 점이 존재합니다. 일단 기본적인 뼈대는 똑같지만 저는 구간을 받은 그 즉시 정렬을 했고 이번 Term Project는 구간을 입력 받는 것이 였기에 먼저 sort(저번 과제 정렬과 똑 같은 알고리즘)를 설정해주면서 코딩의 줄을 조금이나마 줄였습니다.

```
int n = arr_chosen.length;
System.out.println("");
System.out.println("정렬된 데이터");
for (int i1=0; i1<n; i1++) {
    for(int i2=i1+1; i2<n; i2++) {
        if (arr_chosen[i1]>arr_chosen[i2]) {
            int k1 = arr_chosen[i2];
            arr_chosen[i2] = arr_chosen[i1];
            arr_chosen[i1] = k1;
        }
    }
    System.out.println(arr_chosen[i1]);
}
```

즉 원편의 코딩은 Term Project1에서 정렬

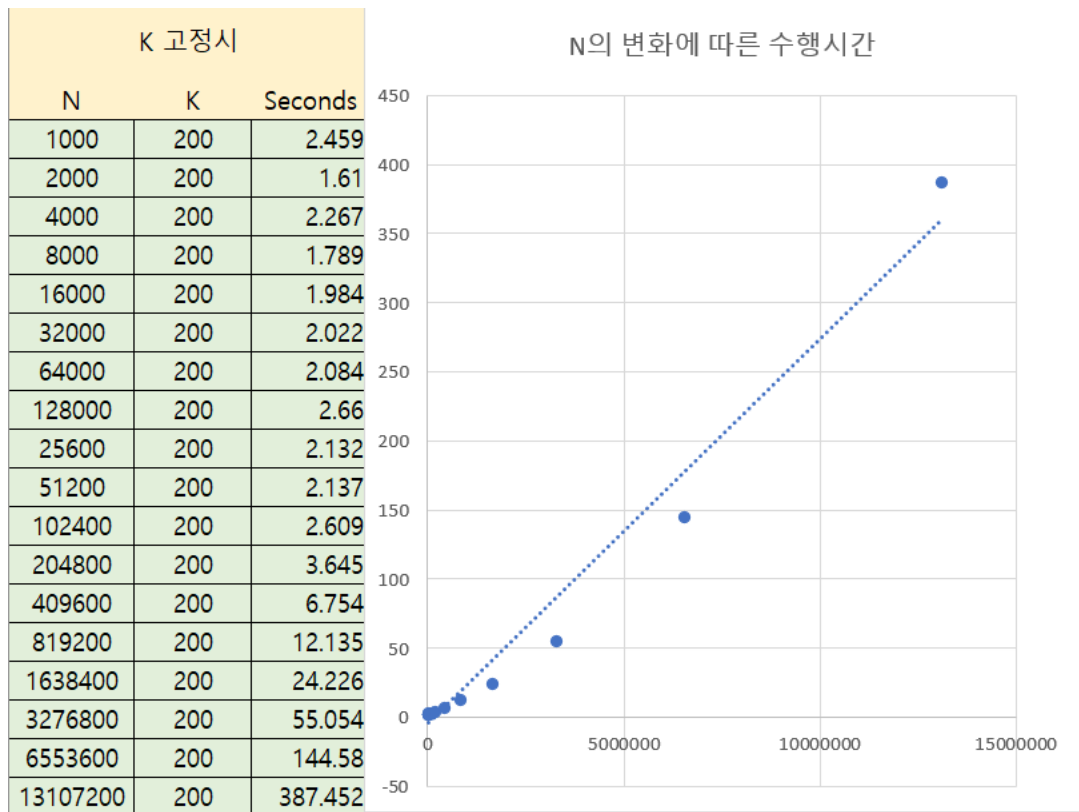
을 위해 사용했던 것인데 이를

```
static void sort(int [] v, int n) {
    // 배열 v, 크기 n인 정렬 함수를 작성한다.
    for(int i=0; i< n; i++) {
        for (int j=i+1; j<n; j++) {
            if(v[i] > v[j]) {
                int k = v[j];
                v[j] = v[i];
                v[i] = k;
            }
        }
    }
}
```

이렇게 간추려서 정렬이 필요할 때 저 긴

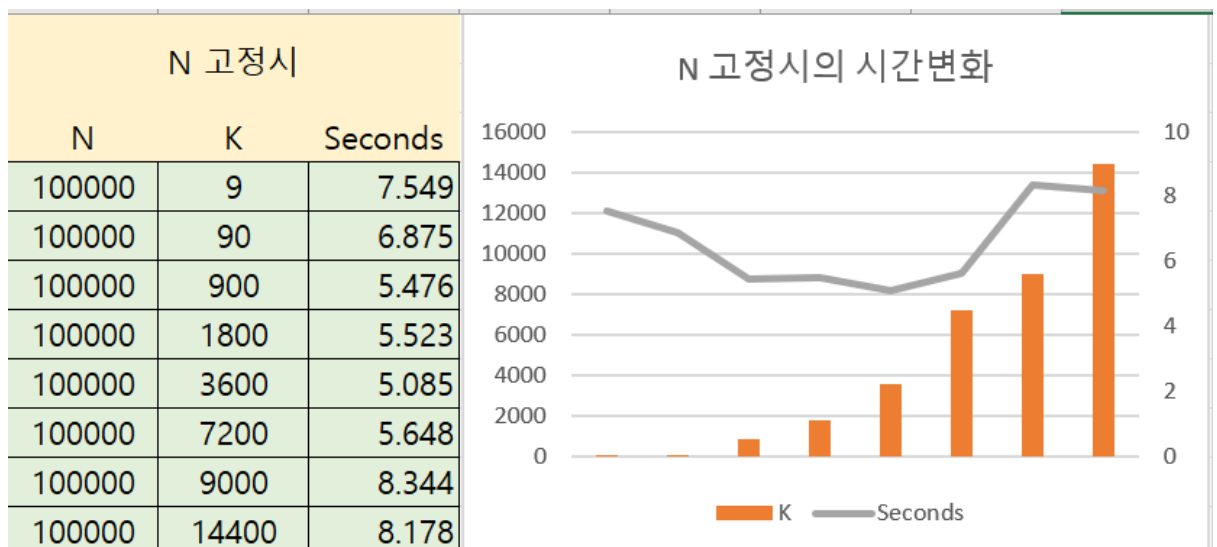
코딩을 매번 쓰는 것이 아닌 sort(배열, 배열의 크기)를 통해 코딩을 간략하게 만들었습니다.

- K 고정 시 N에 따른 알고리즘의 작동 시간 변화



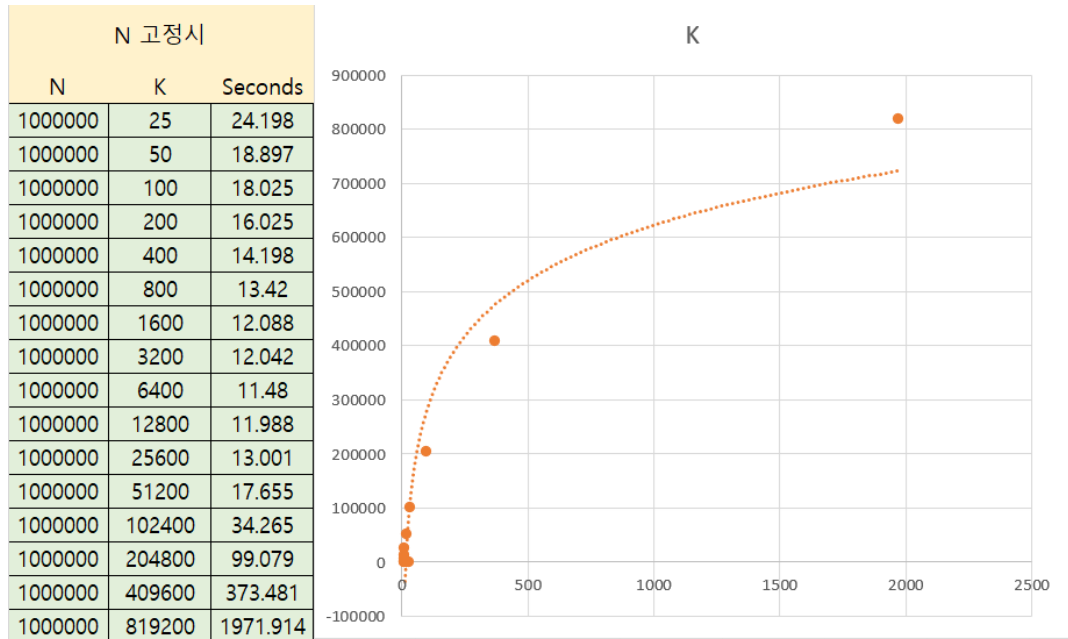
처음 알고리즘을 돌리기 전, K가 고정이 되어 있고 N, 즉 데이터의 크기가 점점 증가하면 그 작동시간 또한 증가할 것이라 예상했습니다. N의 변화는 단순 입력 값의 변화였기에  $O(n)$ 의 시간 복잡도를 지닐 것이라고 생각했습니다. 엑셀 테이블을 이용해서 대략의 추세선을 그어봤을 때 어느 정도의 선형관계를 띄고 있는 것을 볼 수 있었습니다.

- N 고정 시 K에 따른 알고리즘의 작동 시간 변화



데이터를 돌리기 전 저는 작동시간이 약간의 이차함수 모양을 할 것이라고 생각했습니다.

즉 데이터의 개수를 구간별로 알맞게 구분 후에 계산을 하면 최대, 최소, 총합계를 구하는 시간이 조금은 줄어 들 것이라고 생각 했습니다. 그래서 실행을 해본 결과 뭔가  $y=x^2$ 의 모양같이 나온 것 같으면서도 아닌 것 같아 k의 값을 좀 더 크고 항목들을 추가해서 진행을 해보았습니다.



N, K의 값 모두 더 크게 설정하여 추세를 파악할 때 더 용이하게 해보았습니다. 알고리즘을 돌리기 전에는  $O(n^2)$ 의 시간 복잡도를 지니고 있을 것이라 생각했지만 저의 예상과는 다르게 사진에서 볼 수 있듯이 약간  $O(\log n)$ 의 성향을 띄고 있었습니다. 또한 제가 추세 그래프가  $y=x^2$ 의 모양을 띄고 있을 것이라 예측한 것은 k가 50-51200까지 증가해 갈 때 수행시간이 한번 저점을 찍는 구간 때문 이였습니다. 하지만 엑셀 테이블의 결과를 보았을 때 그 저점은 바로 log 함수 그래프의 음수 부분을 x축 대칭 한 값이라고 판단했습니다. 수행시간은 음수가 나오지 않기에 그 값을 x축 대칭을 시키면 잠시 수행시간이 감소하다가 다시 증가하는 모습을 볼 수 있습니다.

알고리즘의 시간 복잡도의 분석을 떠나 엑셀의 테이블을 보면 아시겠지만 데이터의 크기가 커짐에 따라 수행시간이 기하급수적으로 증가했습니다. 지금 제가 작성한 알고리즘에서 정렬이 굉장히 많이 쓰이고 또 중요한 부분인데 저번 Term Project와 이번 알고리즘에서는 선택 정렬을 사용해서  $O(n^2)$ 의 시간 복잡도를 지니고 있습니다. 이 정렬 방법이 저의 수행시간 증가에 주범이라 생각을 하며 최종 프로젝트 발표에서는 다른 알고리즘 (12주차 실습 수업 때 배운 병합 정렬)을 사용해서 좀 더 효율적인 알고리즘을 만들어 볼 생각 입니다.