자료구조 및 알고리즘1

산업시스템공학과
2016112609 김동현

Term Project 1

- N개의 데이터가 있습니다.
- 이중 연속된 구간에서의 최소값, 최대값, 합계를 구하는 프로그램을 작성해야 합니다.
- 입력

1째줄 : N
2째줄-N+1째줄 : 데이터값
N+2째줄 : K
N+3째줄-N+K+2째줄 : <구간 시작값> <구간 종료값>

입력예)
5
1
3
2
4
5
2
1 5
2 3

출력 예)
1 5 15
2 3 5

- 출력

1째줄-K째줄 : <구간별 최소값> <구간별 최대값> <구간별 합계값>

```
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]);
}
```

Term Project 2

- 제출기한 : 11월 27일
- N과 K를 입력받습니다.
  - N개의 데이터를 랜덤으로 생성합니다.
  - K개의 구간을 랜덤으로 생성합니다.
  - 각 구간별로 최소값, 최대값, 합계를 기존의 방식을 이용해서 구합니다.
- K를 고정하고 N을 변화시키면서 그래프를 그립니다.
- N을 고정하고 K를 변화시키면서 그래프를 그립니다.
- 현재 N, K에 대한 이 알고리즘의 성능을 평가합니다.
- 제출할 내용
  - 소스
  - 표지 1장 포함하여, N, K 변화한 것에 대한 그래프 결과
  - 알고리즘 성능 평가와 그 이유

```
static void sort(int [] v, int n) {
    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;
            }
        }
```

Term Project 1

- N개의 데이터가 있습니다.
- 이중 연속된 구간에서의 최소값, 최대값, 합계를 구하는 프로그램을 작성해야 합니다.
- 입력

1째줄 : N
2째줄-N+1째줄 : 데이터값
N+2째줄 : K
N+3째줄-N+K+2째줄 : <구간 시작값> <구간 종료값>

입력예)
5
1
3
2
4
5
2
1 5
2 3

출력 예)
1 5 15
2 3 5

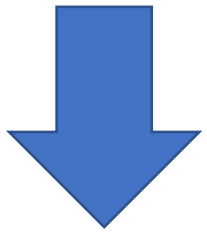
- 출력

1째줄-K째줄 : <구간별 최소값> <구간별 최대값> <구간별 합계값>

```
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]);
}
```

Term Project 1&2
후의 계획

버블 정렬의 시간 복잡도

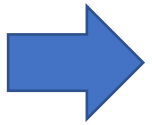


분할정복-합병 정렬의 시간 복잡도

Term Project 3 (개선 사항)

N 고정시

N	K	Seconds
1000000	25	24.198
1000000	50	18.897
1000000	100	18.025
1000000	200	16.025
1000000	400	14.198
1000000	800	13.42
1000000	1600	12.088
1000000	3200	12.042
1000000	6400	11.48
1000000	12800	11.988
1000000	25600	13.001
1000000	51200	17.655
1000000	102400	34.265
1000000	204800	99.079
1000000	409600	373.481
1000000	819200	1971.914



N 고정시

N	K	Seconds
100000000	10	1.12
100000000	20	1.251
100000000	30	1.34
100000000	40	1.721
100000000	50	1.862
100000000	60	1.845
100000000	120	2.703
100000000	240	5.114
100000000	480	11.337
100000000	960	17.172
100000000	1080	20.005
100000000	2160	36.053
100000000	5120	105.378
100000000	10240	228.672

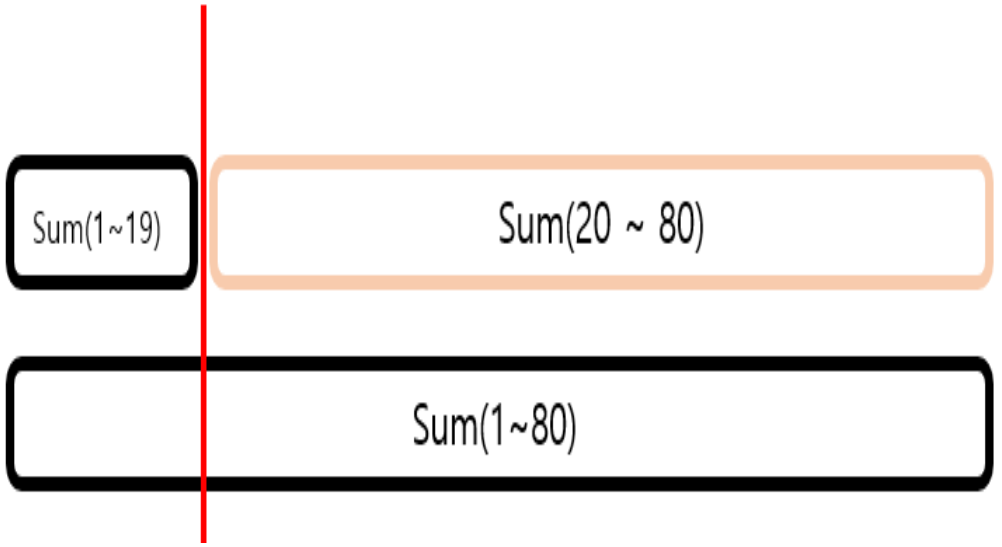
```
for (int i=0; i<k1; i++) {
    int b = random.nextInt(N)+1;
    //System.out.println(b);
    int a = random.nextInt(N)+1;
    //System.out.println(a);
    if(a>b) {int t1=a; a=b; b=t1;}
    int min = arr[a-1];
    int max = arr[a-1];

    long sum = arr[a-1];
    long sum1 = arr[a-1];
    long sum2 = arr[a-1];
    for (int i1=a; i1<b; i1++) {
        if(min > arr[i1]) min = arr[i1];
        if(max < arr[i1]) max = arr[i1];
        //sum1 += arr[i1];
    }
```

Term Project 3 (개선 사항)

```java
long prefixsum[] = new long [arr.length];
prefixsum[0] = arr[0];
for (int i=1; i < arr.length; i++) {
    prefixsum[i] = prefixsum[i-1] + arr[i];
    //System.out.println(prefixsum[i]);
}
```

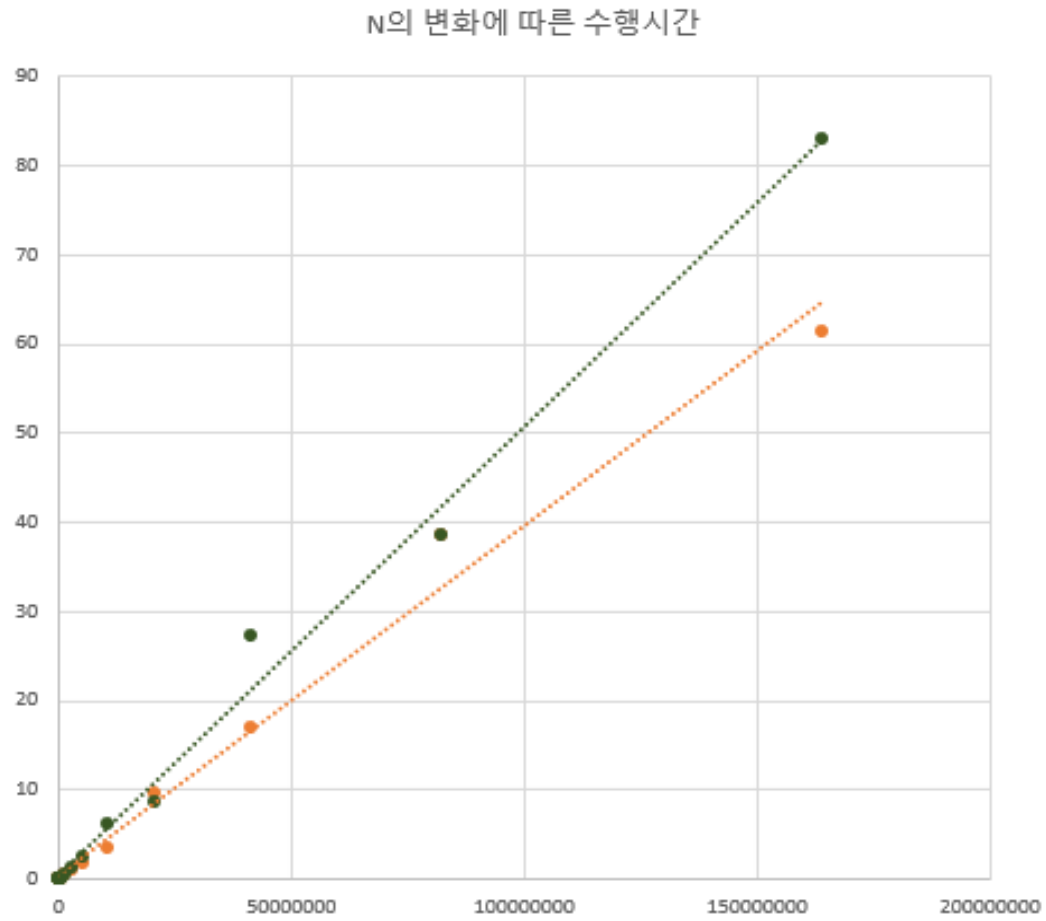
매 반복 시 sum을 구하는 것이 아닌 미리 생성된 구간 별 합을 통해서 구간 합을 갖고 옴



K 고정 시 N에 따른 수행시간 변화

K 고정시		
N	K	Seconds
10000	2000	0.035
20000	2000	0.039
40000	2000	0.063
80000	2000	0.105
160000	2000	0.16
320000	2000	0.241
640000	2000	0.397
1280000	2000	0.685
2560000	2000	1.447
5120000	2000	2.684
10240000	2000	6.217
20480000	2000	8.77
40960000	2000	27.285
81920000	2000	38.561
163840000	2000	83.092

K 고정시		
N	K	Seconds
10000	2000	0.032
20000	2000	0.034
40000	2000	0.043
80000	2000	0.071
160000	2000	0.118
320000	2000	0.197
640000	2000	0.344
1280000	2000	0.574
2560000	2000	1.087
5120000	2000	1.932
10240000	2000	3.548
20480000	2000	9.77
40960000	2000	17.012
81920000	2000	38.561
163840000	2000	61.411



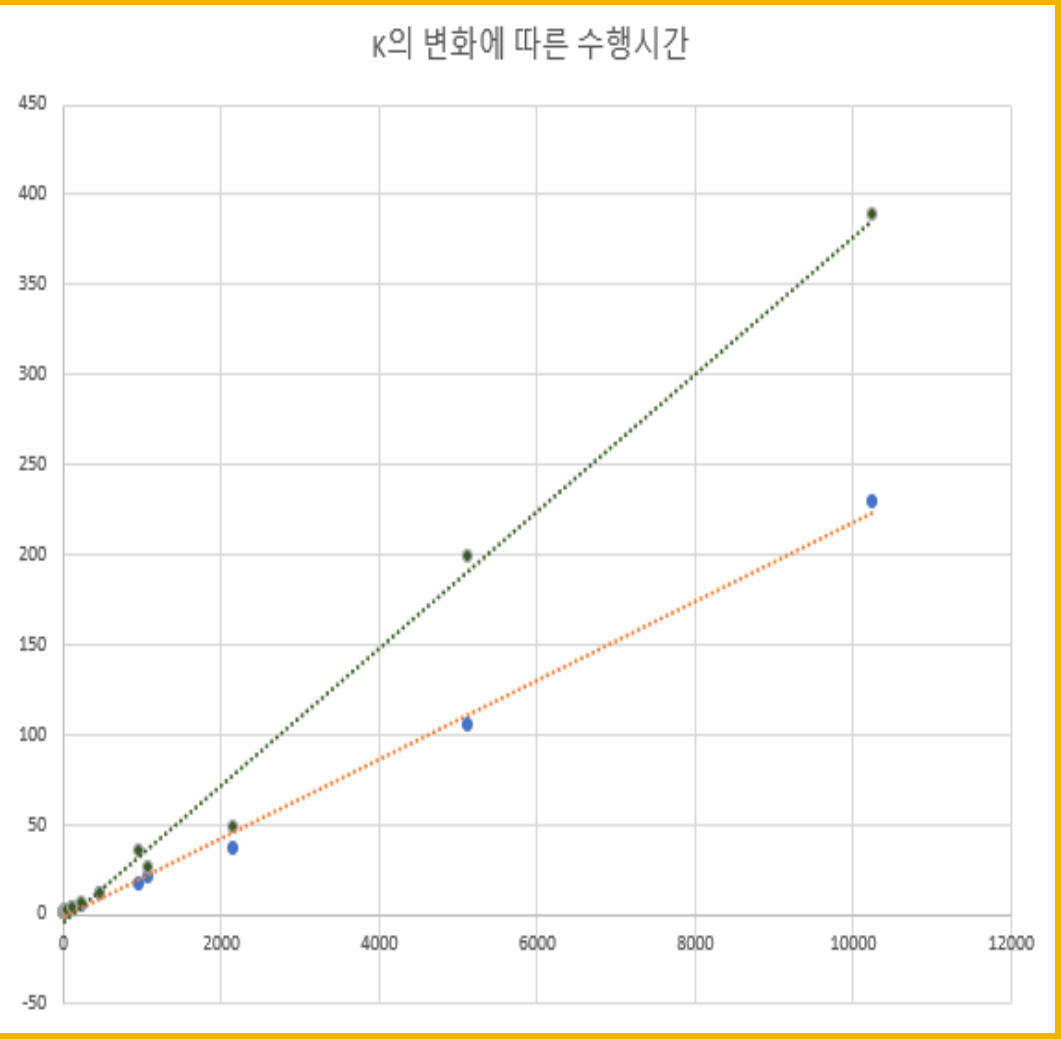
N 고정 시 K에 따른 수행시간 변화

N 고정시

N	K	Seconds
100000000	10	0.824
100000000	20	1.467
100000000	30	1.563
100000000	40	1.815
100000000	50	1.926
100000000	60	2.225
100000000	120	3.753
100000000	240	6.015
100000000	480	11.033
100000000	960	34.675
100000000	1080	25.589
100000000	2160	48.323
100000000	5120	198.668
100000000	10240	388.271

N 고정시

N	K	Seconds
100000000	10	1.12
100000000	20	1.251
100000000	30	1.34
100000000	40	1.721
100000000	50	1.862
100000000	60	1.845
100000000	120	2.703
100000000	240	5.114
100000000	480	11.337
100000000	960	17.172
100000000	1080	20.005
100000000	2160	36.053
100000000	5120	105.378
100000000	10240	228.672



Term Project 고찰

01

정렬로 인한 의미 없는 알고리즘의 시간 복잡도 증가 및 수행 시간 증가.

02

구간 합을 구하는 과정에서의 프로젝트 개선도는 성공적 이였으나 min / max 를 더 효율적으로 구하는 알고리즘 구현 실패

03

기존의 정렬 베이스를 통한 min/max 탐색 포기 및 profix sum을 통한 수행 시간 단축