

TERM PROJECT

:구간이 여러 개 주어졌을 때 문제

강의명	자료구조 및 알고리즘
담당 교수	최창락
학번	2016112723
이름	서상민
제출일	2020.11.24.

코드 요약

1. 구성

1차 텀프로젝트로 제출했던 코드 중 클래스로 구성했던 'termproject.java' 파일의 객체를 불러와 두 번째 텀프로젝트 과제를 구현했습니다. 따라서 'termproject', 'secondB' 두 가지 파일로 구성되어 있습니다.

2. term project 클래스

```
1 import java.util.Random;
2 import java.util.Scanner;
3 class termproject {
4     Scanner sc = new Scanner(System.in);
5     Random rd = new Random();
6     int min;
7     int max;
8     int sum;
9     int size;
10    int a1;
11    int a2;
12    int[] array;
13    termproject() {}
14    public int mini(int a1, int a2) {
15        min = array[a1];
16        for(int i=a1-1; i<a2; i++) {
17            if(array[i]<min) {
18                min = array[i];
19            }
20        }
21        return min;
22    }
23    public int maxi(int a1, int a2) {
24        max = array[a1];
25        for(int i=a1-1; i<a2; i++) {
26            if(array[i]>max) {
27                max = array[i];
28            }
29        }
30        return max;
31    }
32    public int sum(int a1, int a2) {
33        sum = 0;
34        for(int i=a1-1; i<a2; i++) {
35            sum += array[i];
36        }
37        return sum;
38    }
39 }
```

- 최솟값을 찾는 메소드 : mini(int a1, int a2)
배열 array의 [a1, a2] 구간에서 최솟값을 찾고 반환합니다.

- 최댓값을 찾는 메소드 : maxi(int a1, int a2)
배열 array의 [a1, a2] 구간에서 최댓값을 찾고 반환합니다.

- 합을 반환하는 메소드 : sum(int a1, a2)
배열 array의 [a1, a2] 구간의 합계를 구하고 반환합니다.

3. SecondB

```
1 import java.util.*;
2 public class secondB {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         Random rd = new Random();
6         termproject pr = new termproject();
7
8         System.out.println("N(데이터 갯수)와 K(구간의 갯수)를 차례대로 입력하세요.");
9         int N = sc.nextInt();
10        int K = sc.nextInt();
11
12        pr.array = new int[N];
13        for (int i = 0; i < N; i++) {
14            pr.array[i] = rd.nextInt();
15        }
16
17        for(int i=0; i<K; i++) {
18            int a1 = (rd.nextInt(N-1)+1);
19            int plus = rd.nextInt(N-a1);
20            int a2 = a1+plus;
21
22            int minimum = pr.mini(a1, a2);
23            int maximum = pr.maxi(a1, a2);
24            int sum = pr.sum(a1, a2);
25            System.out.println("구간 : "+a1+"-"+a2);
26            System.out.println(minimum+" "+maximum+" "+sum+"\n");
27        }
28        sc.close();
29    }
30 }
```

- 스캐너를 이용하여 데이터 갯수 N과 구간의 갯수K를 입력받습니다.
- termproject 객체를 생성한 뒤 객체 내의 배열에 랜덤으로 데이터를 생성합니다.
- 입력받은 구간의 갯수(K)를 기준으로 for문을 K번 실행시킵니다.
- for 문 : 랜덤으로 구간을 나눌 인덱스 값을 생성한 뒤 termproject 함수의 mini, maxi, sum 함수를 각각 사용해 값을 반환받아 출력합니다.

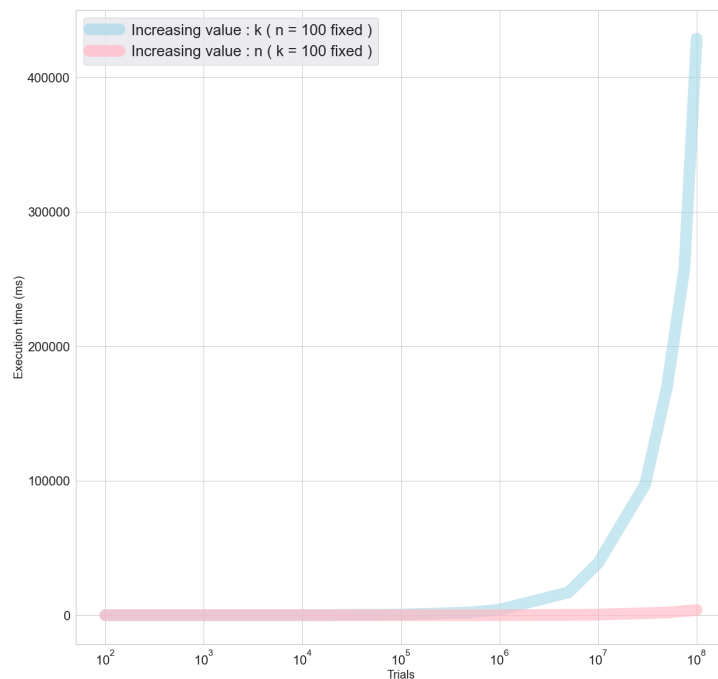
출력 예시)

```
N(데이터 갯수)와 K(구간의 갯수)를 차례대로 입력하세요.
100 1
구간 : 63-96
-1910763851 1989461264 1193160780
```

성능 평가

1. 수행시간 분석

데이터수(N)를 100으로 고정한 뒤 출력해야하는 데이터 구간(K)을 늘려가면서 측정한 수행시간은 파란색 선으로, 반대로 데이터 구간을 100으로 고정한 뒤 데이터수를 늘려가며 측정한 수행시간은 분홍색 선으로 표현하였습니다.



- 데이터 수를 늘릴 때

데이터 수를 1억개 까지 늘리며 측정한 결과, 수행 시간에는 크게 차이가 없는 것으로 보입니다. 배열의 크기가 커져서 탐색에 있어 시간이 더 걸린 것으로 보이지만 구간의 수는 100개로 고정되어 for문이 100번 시행되는 것은 모두 동일하기에 시간 상 큰 차이가 없었습니다. 따라서 N의 증가에 따른 수행 시간은 상수시간($O(1)$)입니다.

- 데이터 구간 수를 늘릴 때

데이터 구간의 증가는 수행 속도의 급격한 증가에 영향을 주었습니다. 엔트리포인트 코드를 살펴보면, for문이 K번 수행됨을 알 수 있습니다. 하지만 코드 안에 호출된 객체의 함수 min, maxi, sum 함수에도 각각 for문이 있습니다. 세 함수 모두 매 번 설정된 구간의 길이만큼 반복함을 알 수 있습니다. 즉 구간의 길이를 x라고 한다면, 수행횟수는 $K*(3x)$ 입니다. 이때 구간의 길이의 평균을 측정해본 결과, 데이터의 수가 100일 때 25, 1000일 때 240, 1만일 때 2585, 십만일 때 22754로 매 회마다 랜덤한 구간의 평균이 데이터 길이의 1/4만큼으로 생성되는 것을 알 수 있었습니다. 따라서 $N=100$ 이므로 $x=25$ 라고 생각할 수 있습니다. 이를 앞서 도출해낸 식에 대입하면, $N=100$ 일 때 수행 시간은 $O(75N)$ 입니다. 이를 좀 더 일반화시키면, N을 일반화하기 위해 K라고 한다면, 수행 시간은 $O(3/4*K*N)$ 라고 볼 수 있습니다.