**N과 K에 따른 알고리즘 성능 평가**

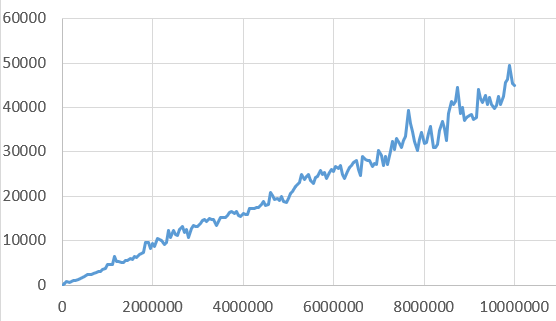


학과 : 산업시스템공학과

학번 : 2016112582

이름 : 김유탄

1. N을 고정 100으로 고정하고 K에 변화를 주는 경우

x 축 : K

y 축 : 시간 (단위 : ms)

N = 100

K : 0 ~ 10.000.000

1. N을 고정시키고 K의 값을 변화 시키면서 시간을 측정한 결과, 선형에 가까운 그래프가 형성되었다.
2. 시간 분석 그래프에서 시간이 단순증가 하지 못하는 이유는, 생성되는 구간의 크기가 랜덤이고, 이에 따라 구간의 합을 구하는 시간과 구간내의 값들을 정렬하는 시간도 랜덤이다.. 이러한 현상은, K의 값의 증가할수록 더욱 심해진다.

* 시간 복잡도 분석에 있어서 최악 경우, 최선 경우가 존재 하고, 수행 시간은 두 경우의 수행 시간 사이에 분포해 있다

1. 코드의 구조 –

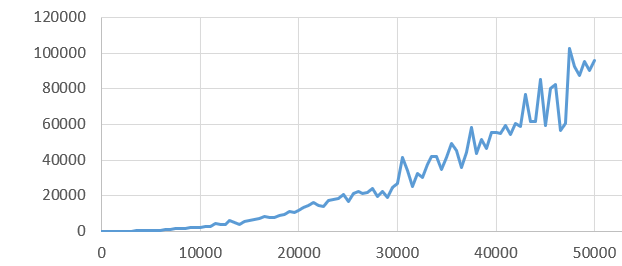
for( K개의 구간 형성){

최대, 최소, 합계 구하기

}

이와 같은 구조이기 때문에 K가 증가 하면, 이에 거의 비례하게 수행 시간이 증가한다.

1. 전체적인 모양은 선형으로 증가 -> 랜덤하게 범위를 형성하는 K에 대해서 알고리즘의 수행시간은 선형시간( O( K )) 이다.
2. K을 고정 100으로 고정하고 N에 변화를 주는 경우

x 축 : N

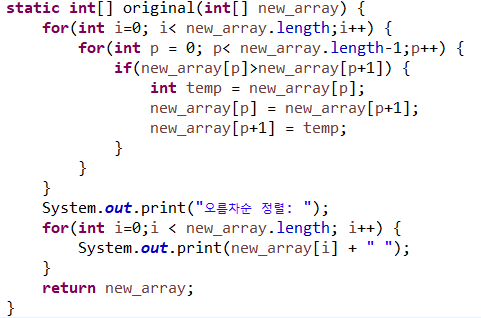
y 축 : 시간 (단위 : ms)

K = 100

N : 0 ~ 10.000

1. K을 고정시키고 N의 값을 변화 시키면서 시간을 측정한 결과, 이차 방정식의 곡선 모양과 유사하다.
2. 그래프의 모양이 진동하는 것처럼 보이는 이유는 구간의 크기가 랜덤이고, 이에 따라 구간의 합을 구하는 시간과 구간내의 값들을 정렬하는 시간도 랜덤이다. 이러한 현상은, N의 값의 증가할수록 더욱 심해진다.

* 시간 복잡도 분석에 있어서 최악 경우, 최선 경우가 존재 하고, 수행 시간은 두 경우의 수행 시간 사이에 분포해 있다.(그래프 내에서 급격하게 떨어지는 부분은 최악 및 최선에 가까운 경우라고 판단)

크기가 N인 배열의 최대 최소를 구하는 과정에서 정렬을 할 때, N^2번을 수행하게 된다.

* 최대 차수의 계수는 생략하고 O괄호 안에 표시 해서 나타내기

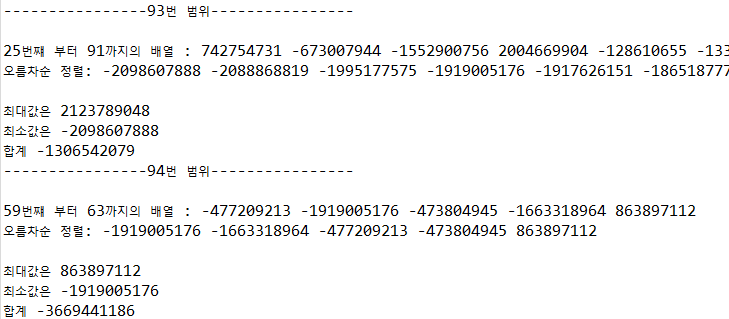
1. 전체적인 모양은 N^2에 비례하는 형태 -> 배열의 크기를 결정하는 N에 대해서 알고리즘의 수행시간은 제곱시간 ( O( N^2)) 이다.
2. 추가 분석

알고리즘 성능평가는 (1) 시간 복잡도 (2) 공간 복잡도 에 대해서 분석하는 것이 일반적이다.

하지만 이번 프로젝트의 알고리즘에 소모되는 컴퓨터의 메모리 사용량이 매우 작기 때문에 시간 복잡도에 대해서만 분석을 했다.

알고리즘의 시간 복잡도 평가는 위에 나온 횟수만큼 5번을 진행했다. 매 실험을 진행할 때 마다 그 결과값은 계속 변했다.

알고리즘 평가를 진행하는 과정에서, 컴퓨터를 다른 용도로 사용함에 따라, 알고리즘의 속도가 느려지는 경우도 발생해서, 평가에 대한 오차가 발생하기도 하였다.(컴파일 시간은 고정되지만 실행 시간은 시간 측정 당시 컴퓨터의 성능과도 연관되어 있기 때문)



위 사진은 알고리즘 N =100, K =100일떄 이다.

생성 되는 구간의 크기가 랜덤이고, 구간의 크기에 따라 수행 시간 또한 크게 변하기 때문에, 일정한 모양의 수행 시간 그래프를 형성하기 어렵다.

1. 평가 및 개선 가능 사항
2. 현재 최대 최소를 구하는 방식

* 구간별로 숫자를 오름 차순으로 정렬 하고, 정렬한 배열의 첫번째와 마지막에 해당하는 숫자를 최소값과 최대값으로 설정한다.

구간 내에서 최대, 최소를 구하기 위해서, 최대 최소에 해당하는 숫자 외에, 다른 숫자들에 대해서도 정렬 연산을 수행하고 있기 때문에 현재 최대 최소를 구하는 방식은 비효율적이라고 판단

1. 개선 가능 사항

* 정렬 방식을 변경
* 정렬 방식이 아닌, 최대 최소만을 뽑는 방식 채택
* 알고리즘 내에 불필요하게 수행되고 있는 코드 제거

또는 중간에 멈추게 하기