

## <알고리즘분석 hw2>

B911094 컴퓨터공학과 2분반 소예원

```
Last login: Thu Apr  8 16:10:00 2021 from 183.100.42.229
[B911094@localhost ~]$ cd hw2
[B911094@localhost hw2]$ gcc hw2.c
[B911094@localhost hw2]$ ./a.out
sorted          N=50000          N=100000          N=150000
Mergesort       0.010000          0.020000          0.020000
Quicksort       3.700000          13.150000         29.629999

random
Mergesort       data1      N=1000000      N=2000000      N=4000000
                data2      0.240000      0.510000      1.070000
                data3      0.240000      0.510000      1.080000
                average    0.243333      0.510000      1.076667

Quicksort       data1      0.200000      0.430000      0.890000
                data2      0.210000      0.430000      0.890000
                data3      0.200000      0.430000      0.890000
                average    0.203333      0.430000      0.890000
[B911094@localhost hw2]$
```

정렬된 데이터 값을 각각 N(데이터의 개수)이 5만일 때, 10만일 때, 15만일 때로 나누고, 랜덤 데이터 값을 각각 N이 100만일 때, 200만일 때, 400만일 때로 나누어 각각의 경우에서 mergesort와 quicksort구현 시 걸리는 시간을 알아보았다.

Mergesort는 부분 배열로 나누어지는 단계에서 비교 연산, 이동 연산이 수행되지 않으므로 데이터의 정렬 정도에 크게 영향을 받지 않는다. 즉, 데이터가 이미 정렬이 되어 있든, 되어 있지 않든 그 여부에 크게 영향을 받지 않고, (worst case에서 시간 복잡도가 크게 변하지 않는다) 데이터의 크기에만 영향을 받게 된다. 따라서 Mergesort의 이론적 시간복잡도는 언제나  $O(N\log N)$ 이다.

Mergesort에서 sorted데이터가 들어갈 때, Mergesort의 경우 이론적으로  $O(N\log N)$ 의 시간 복잡도를 가지므로, N값이 2배 증가할 때마다 약 2배 이상 가량의 시간이 더 걸리게 된다. 데이터의 개수 N의 비가 1 : 2 : 3일 때, 측정된 시간은 0.010000s : 0.020000s : 0.020000s = 약 1 : 2 : 2의 비를 갖는데, 이는 clock의 오차를 고려하면 이론적 시간 복잡도의 비와 비슷한 값을 갖는다. random데이터의 경우도 마찬가지로  $O(N\log N)$ 의 시간복잡도를 가지므로, N값이 2배 증가할 때마다 약 2배 이상 가량의 시간이 더 걸리게 된다. 데이터의 개수 N의 비가 1 : 2 : 4일 때, 측정된 시간은 0.243333s : 0.510000s : 1.076667s = 약 1 : 2.1 : 4.4의 비를 갖는데, 이는 clock의 오차를 고려하면 이론적 시간 복잡도의 비와 비슷한 값을 갖는다.

반면 Quicksort는 pivot값을 잡아 하나 하나씩 모두 다 비교하므로 데이터의 정렬 정도에 영향을 받게 되는데, 이미 정렬된 상태일 때는 worst case로  $O(n^2)$ 의 이론적 시간복잡도를 가지게 되고, 보통의 경우에는 이론적으로  $O(N\log N)$ 의 시간복잡도를 가지게 된다.

Quicksort에서 sorted데이터가 들어갈 때 최악의 경우  $O(N^2)$ 의 시간복잡도를 가지므로, 이론적으로 실행시간이 약 1 : 4 : 9의 비를 가져야 하는데, 각각의 경우에 걸린 실행시간을 보면, 3.700000s : 13.150000s : 29.629999s = 1 : 3.6 : 8.0 으로 clock의 오차를 고려하면 이론적 시간 복잡도의 비와 비슷한 값을 갖는다. random데이터의 경우에는 평균적으로  $O(N\log N)$ 의 시간복잡도를 가지므로, N값이 2배 증가할 때마다 약 2배 이상 가량의 시간이 더 걸리게 된다. 데이터의 개수 N의 비가 1 : 2 : 4일 때, 측정된 시간은 0.203333s :

0.430000s : 0.890000s = 약 1 : 2.1 : 4.4의 비를 갖는데, 2배보다 조금 더 늘어났지만, 이는 clock의 오차를 고려하면 이론적 시간 복잡도의 비와 비슷한 값을 갖는다.

따라서 모든 데이터를 종합해 보았을 때, quicksort와 mergesort는 sorted와 random데이터에 대해 이론적 시간복잡도와 비슷한 시간이 걸림을 알 수 있다.