CSE3081-3: Design and Analysis of Algorithms(Fall 2022)

Machine Problem 2: Master of Sorting

20181536 엄석훈

1. 각 알고리즘 별 입력에 따른 5회 평균 러닝타입 비교

1) 랜덤한 순서의 입력

입력 사이즈	알고리즘 1 Insertion sort	알고리즘 2 Quick sort	알고리즘 3 Merge sort	알고리즘 4 나의 알고리즘
10	**0.000003초**	**0.000003초**	**0.000006초**	**0.000003초**
100	**0.000027초**	**0.000019초**	**0.000042초**	**0.000020초**
1000	**0.002215초**	**0.000224초**	**0.000473초**	**0.000225초**
10000	**0.121986초**	**0.002989초**	**0.005545초**	**0.003631초**
100000	**8.576076초**	**0.019518초**	**0.036012초**	**0.019498초**
1000000	**측정X**	**0.184028초**	**0.305350초**	**0.174986초**

2) 증가하지 않는 순서로 정렬된 입력

입력 사이즈	알고리즘 1 Insertion sort	알고리즘 2 Quick sort	알고리즘 3 Merge sort	알고리즘 4 나의 알고리즘
10	**0.000003초**	**0.000004초**	**0.000006초**	**0.000003초**
100	**0.000048초**	**0.000049초**	**0.000033초**	**0.000023초**
1000	**0.004352초**	**0.004154초**	**0.000326초**	**0.000233초**
10000	**0.209607초**	**0.199559초**	**0.003878초**	**0.002977초**
100000	**17.436879초**	**16.420642초**	**0.025021초**	**0.189380초**
1000000	**측정X**	**측정X**	**0.206358초**	**0.163627초**

2. 알고리즘 4 설명

1) 실험 환경 : 모든 알고리즘의 러닝타임을 측정할 때는 학교에서 제공해준 cspro9 서버의 자원을 이용하였다.

2) 실험 설정 : 입력을 사이즈는 10부터 1000000까지 매우 작은 범위부터 큰 범위까지의 숫자를 정렬하도록 하였다. 또한 러닝타임을 측정하여 비교하였다.

3) 실험 코멘트 : 알고리즘4를 설계하기 전에 insertion sort, quick sort, merge sort의 러닝타임을 비교하였을 때 랜덤한 입력의 경우는 quick sort와 merge sort가 O(nlogn)으로 asymptotic하게는 속도가 동일하지만 실제 러닝타임은 quick sort가 가장 빨랐다. 반면에 증가하지 않는 순서로 정렬된 입력의 경우에는 quick sort는 O(n^2)의 속도를 가지

고 있어 insertion sort와 비슷한 시간이 걸렸고 merge sort는 훨씬 적은 시간이 걸렸다. 따라서 알고리즘 4를 설계할 때 랜덤한 입력의 경우는 quick sort를 사용하고 다른 경우에는 merge sort를 사용하도록 해주었다.
이를 위해서 argo4_test()함수를 만들어서 quick sort를 수행할 지 merge sort를 수행할 지 결정해주었다.
그리고 다음으로 merge sort가 수행될 때 입력이 매우 작은 경우에는 merge sort가 insertion sort와 같은 다른 정렬 방식에 매우 느리다는 것을 코드를 테스트 하는 과정에서 알게 되었다. 따라서 merge sort를 개량해서 큰 배열을 나누어서 배열의 사이즈가 32이하가 되면 insertion sort를 수행한 뒤 정렬된 작은 배열들을 merge하는 방식으로 merge sort를 insertion sort와 합쳐서 개량해주었다.
그 결과 랜덤한 입력이거나, 감소하지 않는 순서로 된 입력이거나 모두 이전의 알고리즘 1,2,3 보다 훨씬 훌륭한 모습을 보여주었다.