

알고리즘설계와분석 (CSE3081-03) 2차 과제 보고서

20181255 김기철

1. Experiment Environment

OS type : Windows 10

Memory size : 8GB

CPU speed : 2.4GHz

2. Experiment setup

Algorithm : 4개의 정렬 알고리즘을 구현하였다. 1번으로 insertion sort, 2번으로 quick sort, 3번으로 merge sort를 구현하였고, 4번으로 개선된 quick sort를 구현하였다.

Metric : int 범위 사용 ($-2,147,483,648 \sim 2,147,483,647$)

Range of input size : 10부터 100만까지 random 함수를 통해 무작위 배열로 함수를 실행해보았고, 최악의 상황에서의 차이를 보기 위해 non-increasing 하는 배열로도 함수를 실행해보았다.

3. Comments on the experience

다양한 정렬 함수들을 직접 구현해보고, 다양한 input size에 대하여 정렬 함수들을 비교해보는 시간을 가졌다. Insertion sort, quick sort, merge sort 함수의 running time을 비교해보았고, 직접 구현한 my sort 함수와도 비교하여 효율적으로 구현했는지 확인했다.

3-1) How you designed algorithm 4

quick sort 개선하는 방법의 하나로 알려진 Median of Three를 택했다. 기존 quick sort의 partition 함수를 호출하기 전, 배열의 left, mid, right 인덱스에 해당하는 3개의 값을 정렬해주었다. 정렬 후, mid 인덱스에 해당하는 값을 pivot으로 택하여 partition을 진행하였다. 또한, 배열의 크기가 100 이하라면 insertion sort를 호출하여 조금 더 속도를 높였다.

3-2) Describe how the running time of your algorithms grow as the input size grows

① input size : 10

(1) random list

sort function	running time (sec)
insertion sort	0.000032
quick sort	0.000042
merge sort	0.000043
my sort	0.000033

(2) non-increasing list

sort function	running time (sec)
insertion sort	0.000036
quick sort	0.000034
merge sort	0.000040
my sort	0.000037

② input size : 100

(1) random list

sort function	running time (sec)
insertion sort	0.000108
quick sort	0.000085
merge sort	0.000135
my sort	0.000103

(2) non-increasing list

sort function	running time (sec)
insertion sort	0.000149
quick sort	0.000163
merge sort	0.000159
my sort	0.000154

④ input size : 10,000

(1) random list

sort function	running time (sec)
insertion sort	0.211670
quick sort	0.008421
merge sort	0.013463
my sort	0.008591

(2) non-increasing list

sort function	running time (sec)
insertion sort	0.490734
quick sort	0.584477
merge sort	0.012758
my sort	0.007700

⑤ input size : 100,000

(1) random list

sort function	running time (sec)
insertion sort	42.969183
quick sort	0.062993
merge sort	0.114553
my sort	0.000154

(2) non-increasing list

sort function	running time (sec)
insertion sort	48.338852
quick sort	54.849602
merge sort	0.094571
my sort	0.154603

⑥ input size : 1,000,000

(1) random list

sort function	running time (sec)
insertion sort	-
quick sort	0.529195
merge sort	0.673076
my sort	0.525567

(2) non-increasing list

sort function	running time (sec)
insertion sort	-
quick sort	-
merge sort	0.824719
my sort	0.771629

시간이 매우 올리는 case는 -로 표시하였다. 표를 보면 알 수 있듯이 input size가 커짐에 따라 worst case에서 quick sort는 $O(n^2)$ 으로 매우 오래 걸리지만, 직접 구현한 개선된 quick sort에서는 효율적으로 정렬이 되는 것을 볼 수 있다.