

REPORT



Report 작성 서약서

1. 나는 타학생의 보고서를 복사하지 않았습니다.
2. 나는 타학생에게 보고서를 보여주지 않았습니다.
3. 나는 보고서 제출 기한 경과 시점부터 25%씩 감점을 받으며 24시간 마다 추가로 25%의 감점을 받겠습니다.

과목/분반	자료구조 3분반
과제명	정렬알고리즘 성능 비교
담당교수	오 훈
학과/학년	IT융합학부/2학년
학번	20202086
이름	권다운
제출일 (년/월/일)	2021/6/2

(1) 프로그램 코드를 제출하시오.

보고서와 같이 제출하였습니다.

(2) 다음 형식의 비교 분석표를 작성하시오.

● 오름차순 입력 데이터에 대한 알고리즘 성능 비교표

	실행시간 (단위 us(마이크로 세컨드))			
	10	100	1000	10000
Selection Sort	100	1000	10000	15000
Shell Sort	0	0	0	1000
Quick Sort	0	0	1000 or 2000	측정 불가
Heap Sort	0	0	0	3000

■ Selection Sort는 주어진 예시를 활용하였으며, ms단위에서 us단위로 바꿔주었습니다

● 내림차순 입력 데이터에 대한 알고리즘 성능 비교표

	실행시간 (단위 us(마이크로 세컨드))			
	10	100	1000	10000
Selection Sort	100	1000	10000	15000
Shell Sort	0	0	0	1000
Quick Sort	0	0	1000 or 2000	측정 불가
Heap Sort	0	0	0	3000

■ Selection Sort는 주어진 예시를 활용하였으며, ms단위에서 us단위로 바꿔주었습니다

● 랜덤 입력 데이터에 대한 알고리즘 성능 비교표

	실행시간 (단위 us(마이크로 세컨드))			
	10	100	1000	10000
Selection Sort	100	1000	10000	15000
Shell Sort	0	0	0	3000
Quick Sort	0	0	0 or 1000	측정 불가
Heap Sort	0	0	0	3000

■ Selection Sort는 주어진 예시를 활용하였으며, ms단위에서 us단위로 바꿔주었습니다

(3) 위 입력 데이터 보고서에 수집된 데이터에 기초하여 다음 방법들 간의 이점과 단점 평가 하시오.

1. Selection Sort 와 Shell Sort

입력데이터가 10000일 때, Selection Sort는 전부 15000 us가 나오고, Shell sort는 1000 us 혹은 3000 us의 값이 나왔습니다. Shell Sort는 작은 리스트들로 나누어서 정렬해가기 때문에 오름차순, 내림차순, 랜덤 모든 부분에서 실행시간이 더 빠르다는 이점을 갖고 있습니다.

2. Selection Sort 와 Quick Sort

입력데이터가 1000일 때를 기준으로 비교해보았을 때, Quick Sort의 실행시간이 더 빠르다는 결론을 내릴 수 있었습니다. Quick Sort의 실행시간은 평균적으로 $O(n \log n)$ 으로 Selection Sort보다 빠르다는 점을 보여주고, Quick Sort가 더 효율적인 이점을 가지고 있음을 알게 되었습니다.

3. Quick Sort 와 Heap Sort

Heap Sort의 전반적인 실행시간이 Quick Sort보다 빠르다는 것을 알 수 있었습니다. 둘의 시간 복잡도는 $O(n \log n)$ 으로 같지만 Quick Sort가 Worst Case일 경우에는 $O(n^2)$ 이라는 점이 둘의 실행시간 차이를 보여준 거 같습니다.

◆ 과제 수행후기

1. 과제 수행에 걸린 시간 및 적절성

수행하는 데에는 총 7일이 걸렸습니다. 처음 2일 동안은 정렬에 대한 개념과 각 정렬 종류의 개념을 익히고, Implementation을 통해 알고리즘을 공부했습니다. 다음 이틀은 배운 정렬들을 구현해보았습니다. 다시 코딩을 통해 구현하면서 어떻게 작동하는지 익힐 수 있었습니다. 그리고, 입력데이터를 넣어주기 위해 main함수를 구현할 때 이틀이 걸리고, 마지막으로 보고서를 작성하며 총 7일이 걸렸습니다.

2. 과제 수행상의 문제점 및 요구사항

프로젝트를 하면서 Selection Sort는 처음 보았었습니다. Insertion Sort와는 다르지만 유사하다는 것을 알게 되었고, 공지사항의 Insertion Sort의 예시에 관한 얘기가 나와서 Insertion Sort라고 생각하고 보고서를 작성하였습니다. 이 부분에서 Selection Sort가 Insertion Sort로 수정되었으면 좋겠습니다.

3. 과제 수행에 대한 소감

실행시간을 구할 때, 각 sort만 측정할 수 있게 코딩 하였음에도 0 us가 나오는 경우와 입력데이터가 커질 때 측정 불가능한 경우가 있었습니다. 측정 불가능한 경우는 오류코드 C6262로 스택 사이즈를 초과했음을 알리는 오류였습니다. 다만, 이해를 하지 못 하여 측정 할 수 없었습니다. 그리고, 데이터를 출력하기 전까지는 내림차순의 입력 데이터들이 제일 오래 걸릴 줄 알았습니다. 허나, 각 정렬의 작동 방식으로 생각과는 다른 결과를 보여주었습니다. 이번 프로젝트를 통해 알고리즘을 더 효율적으로 만드는 방법을 배우고, 코딩에서 부족한 점을 보완할 수 있었습니다.