

주제: 여러 가지 정렬 알고리즘들의 구현 및 성능 비교

가. 다음 정렬 알고리즘들을 구현하고 실행 속도를 비교 확인하시오.

1. Insertion Sort
2. Shell Sort
3. Quick Sort (Pivot은 각 세그먼트의 선두에 있는 숫자로 한다)
4. Heap Sort

입력 데이터의 크기는 모니터에서 입력을 받도록 하시오. 그리고 각 입력 데이터는 다음과 같이 세 가지 형태로 만들어 집니다.

1. 오름차순으로 된 양의 정수 어레이
2. 내림차순으로 된 양의 정수 어레이
3. 랜덤순서 (Random Order)로 된 양의 정수 어레이

[입력 만드는 법]

오름차순: 10000을 입력받으면 1에서 10000까지의 숫자를 어레이에 차례로 삽입한다.

내림차순: 10000을 입력받으면 10000에서 1까지의 숫자를 어레이에 차례로 삽입한다.

랜덤순서: 오름차순 어레이에서 [0, 9999] 속하는 두 개의 랜덤번호를 생성하고 해당 위치에 속한 두 번호의 위치 값을 스와핑한다. 이를 5000번 반복하면 어느 정도 숫자들의 shuffling이 만들어 지고 이를 사용하여 정렬을 수행한다 (파이선에서 suffling하는 함수가 있으니 이를 사용하는 것도 가능).

입력: N = 10일 때의 입력 예

1. (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2. (10, 9, 8, 7, 6, 5, 4, 3, 2, 1)
3. (6, 1, 8, 5, 10, 3, 9, 2, 4, 7)

출력:

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

(1) 프로그램 코드를 제출하시오

(2) 다음 형식의 비교 분석표를 작성하시오

■ 오름차순 입력 데이터에 대한 알고리즘 성능 비교표

	실행시간 (값의 크기를 고려하여 단위 us 혹은 ms를 사용)			
	10	100	1000	10000
Insertion Sort	0.1	1	10	15
Shell Sort				
Quick Sort				
Heap Sort				

■ 내림차순 입력 데이터에 대한 알고리즘 성능 비교표

■ 랜덤 입력 데이터에 대한 알고리즘 성능 비교표

(3) 위 입력 데이터 보고서에 수집된 데이터에 기초하여 다음 방법들 간의 이점과 단점을 평가하시오 (각 평가는 3줄 이내):

1. Insertion Sort 와 Shell Sort
2. Insertion Sort 와 Quick Sort
3. Quick Sort 와 Heap Sort

나. 과제 수행후기 (각 항목에 대하여 5줄 이내로 작성)

1. 과제 수행에 걸린 시간 및 적절성

2. 과제 수행상의 문제점 및 요구사항

3. 과제수행에 대한 소감