

4주차 2차시 삽입, 쉘 정렬

【학습목표】

1. 삽입 정렬을 설명할 수 있다.
2. 쉘 정렬을 설명할 수 있다.

학습내용1 : 삽입 정렬

1. 삽입정렬이란?

삽입정렬(Insert Sort)은 매우 간단한 방법으로 적은 양의 데이터를 처리하는데 매우 유용하다.

1) 삽입 정렬의 과정

가장 먼저 첫 번째 데이터와 두 번째 데이터를 비교하여 오름차순으로 정렬한다. 이렇게 정렬된 2개의 데이터를 하나의 subfile 로 간주하고 이것과 세 번째 데이터를 비교하여 오름차순 정렬이 되도록 세 번째 데이터를 알맞은 위치에 삽입시킨다. 그 다음에는 이렇게 정렬된 3개의 데이터를 다시 subfile로 간주하고 이것과 네 번째 데이터를 비교하여 오름차순 정렬이 되도록 네 번째 데이터를 알맞은 위치에 삽입시킨다. 이러한 방법으로 마지막 데이터까지 반복하면서 알맞은 위치에 데이터를 삽입하면 오름차순으로 정렬된 데이터를 얻을 수 있다.

이를 그림으로 표현하면 다음과 같다.

2) 삽입 정렬의 예

<원본 데이터>

3	1	4	8	6	10	7	11	9	2	5
20	0	30	70	50	90	60	100	80	10	40

<1-1단계>

먼저 맨 앞의 두 개의 데이터를 오름차순으로 정렬한다.

1	3	4	8	6	10	7	11	9	2	5
0	20	30	70	50	90	60	100	80	10	40

<1-2단계>

맨 앞의 두 개의 데이터와 세 번째 데이터를 비교하여 오름차순이 될 수 있도록 세 번째 데이터를 적절한 위치에 끼워 넣는다.

1	3	4	8	6	10	7	11	9	2	5
0	20	30	70	50	90	60	100	80	10	40

<2단계>

세 개의 데이터를 하나의 subfile로 간주하고 네 번째 데이터를 적절한 위치에 삽입시킨다.

1	3	4	8	6	10	7	11	9	2	5
0	20	30	70	50	90	60	100	80	10	40

<3단계>

네 개의 데이터를 하나의 subfile로 간주하고 다섯 번째 데이터를 적절한 위치에 삽입시킨다.

1	3	4	6	8	10	7	11	9	2	5
0	20	30	50	70	90	60	100	80	10	40

<4단계>

다섯 개의 데이터를 하나의 subfile로 간주하고 여섯 번째 데이터를 적절한 위치에 삽입시킨다.

1	3	4	6	8	10	7	11	9	2	5
0	20	30	50	70	90	60	100	80	10	40

<5단계>

여섯 개의 데이터를 하나의 subfile로 간주하고 일곱 번째 데이터를 적절한 위치에 삽입시킨다.

1	3	4	6	7	8	10	11	9	2	5
0	20	30	50	60	70	90	100	80	10	40

<6단계>

일곱 개의 데이터를 하나의 subfile로 간주하고 여덟 번째 데이터를 적절한 위치에 삽입시킨다.

1	3	4	6	7	8	10	11	9	2	5
0	20	30	50	60	70	90	100	80	10	40

<7단계>

여덟 개의 데이터를 하나의 subfile로 간주하고 아홉 번째 데이터를 적절한 위치에 삽입시킨다.

1	3	4	6	7	8	9	10	11	2	5
0	20	30	50	60	70	80	90	100	10	40

<8단계>

아홉 개의 데이터를 하나의 subfile로 간주하고 열 번째 데이터를 적절한 위치에 삽입시킨다.

1	2	3	4	6	7	8	9	10	11	5
0	10	20	30	50	60	70	80	90	100	40

<9단계>

열 개의 데이터를 하나의 subfile로 간주하고 열한 번째 데이터를 적절한 위치에 삽입시킨다.

1	2	3	4	5	6	7	8	9	10	11
0	10	20	30	40	50	60	70	80	90	100

모든 단계를 거치면 다음과 같이 오름차순으로 정렬된 데이터를 얻을 수 있다.

1	2	3	4	5	6	7	8	9	10	11
0	10	20	30	40	50	60	70	80	90	100

삽입 정렬의 성능은 최적일 경우 입력자료가 정렬하고자 하는 순서대로 배열되어 있는 경우로 최소 비교 횟수는 $n-1$ 이 되며, 최악의 상황인 입력 자료가 역순으로 정렬되어 있는 경우의 최대 비교 횟수는 $(n(n-1))/2$ 이 된다. 또한 각 패스별로 중간 정도에서 부분적인 정렬이 완료되었을 경우 평균 비교 횟수는 $(n(n-1))/4$ 이며, n 개의 레코드를 정렬할 때 $n-1$ 번 삽입이 발생되어 전체 수행시간은 $O(n^2)$ 이 된다.

삽입 정렬은 메모리 사용 공간이 별도의 기억 공간을 필요로 하지 않기 때문에, 리스트의 크기만큼의 기억 공간이 있으면 된다. 삽입 정렬은 간단하며, 안정적이며, 적은 비교와 많은 교환이 필요한 작은 레코드 배열에 가장 유리하다. 또한 순서가 틀린 레코드의 순서가 전체 레코드 수에 비해 현저하게 적은 경우 효율적이므로 원시 리스트가 부분적으로 정렬되어 있는 경우에 효과적이다. 그러나 매회 서브파일의 크기가 증가하는 단점이 있다.

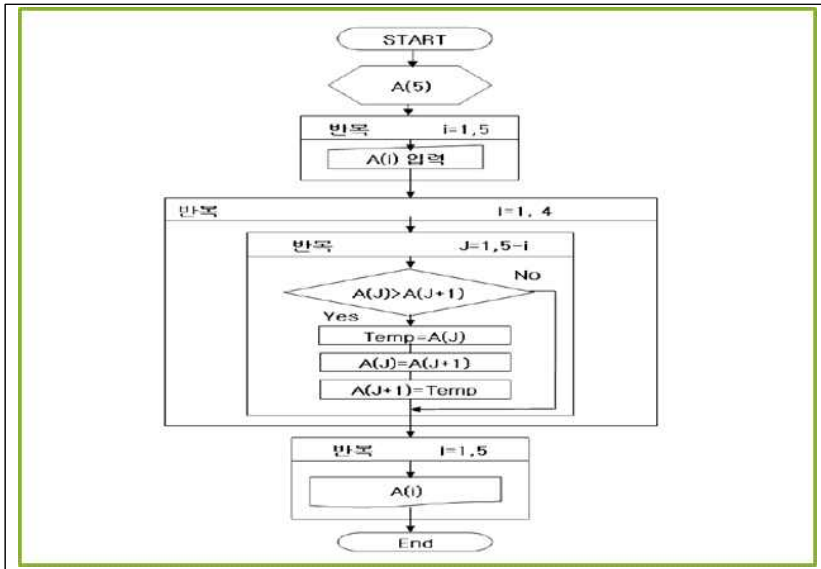
2. 삽입정렬(insertion sort)의 요약

좌측으로부터 한 원소씩 제 자리에 삽입하는 방법으로서 삽입할 원소보다 큰 것은 우측으로 하나씩 이동하고 빈자리에 삽입원소를 집어넣는다.

- ① 별도의 배열이 필요 없는 제자리 정렬이다.
- ② 정렬 후에도 같은 키들의 상대 위치가 그대로 유지되는 안정적 정렬 알고리즘이다.
- ③ 삽입정렬에서 최악의 경우는 역순으로 정렬된 것이고 최선의 경우는 이미 제 순서대로 정렬 된 것일 때 이다.

* 삽입정렬의 실행 예

최초 자료 : 64, 28, 33, 76, 55, 12, 43
 pass 1 : 28, 64, 33, 76, 55, 12, 43
 pass 2 : 28, 33, 64, 76, 55, 12, 43
 pass 3 : 28, 33, 64, 76, 55, 12, 43
 pass 4 : 28, 33, 55, 64, 76, 12, 43
 pass 5 : 12, 28, 33, 55, 64, 76, 43
 pass 6 : 12, 28, 33, 43, 55, 64, 76



학습내용2 : 셸 정렬

1. 셸 정렬이란?

셸 정렬(Shell Sort)은 삽입정렬의 확장 개념으로 생각하면 쉽다.

1) 셸 정렬의 과정

먼저 입력 파일 안에 있는 데이터를 적당한 매개변수를 기준으로 그 값만큼 떨어진 데이터들을 하나의 subfile로 간주한다. 그렇게 해서 만들어진 각각의 subfile에서 삽입정렬의 방법을 이용하여 데이터를 정렬하게 된다. 이후 매개변수의 값을 적당하게 줄여나가면서 위의 방법을 반복하다가 매개변수의 값이 1이 될 때까지 반복하면 비로소 모든 데이터가 정렬되게 된다.

2) 셸 정렬의 예

<원본데이터>

3	1	9	5	4	6	8	10	2	7
15	4	42	28	19	30	38	87	7	35

<1-1단계>

매개변수를 5로 하여 인접한 데이터들끼리 비교한다.(배경 무늬가 같은 데이터끼리 묶어서 하나의 subfile로 간주하고 비교한다.)

3	1	9	5	4	6	8	10	2	7
15	4	42	28	19	30	38	87	7	35

<1-2단계>

이중 4번째 데이터가 9번째 데이터보다 크므로 둘의 위치를 바꾼다.

15	1	5	2	4	9	8	15	5	7
7	4	19	7	19	38	87	28	35	

<2단계>

이번에는 매개변수를 3으로 하여 인접한 데이터들끼리 비교한다.

15	1	9	5	4	6	10	5		
7	4	42	19	19	30	87	28	35	

<3단계>

이중 첫 번째 데이터 15와 네 번째 데이터 7 을 서로 바꾸고, 7번째 데이터 38과 10번째 데이터 35를 바꾸고, 세 번째 데이터 42 와 아홉 번째 데이터 28을 바꾼다.

2	1	5	3	4	6	7	10	9	8
7	4	28	15	19	30	35	87	42	38

<4단계>

이번에는 매개변수를 2로 하여 인접한 데이터를 비교한다.

2	1	5	3	4	6	7	10	9	8
7	4	28	15	19	30	35	87	42	38

<4-1단계>

세 번째 데이터 28과 다섯 번째 데이터 19를 서로 바꾸고, 여덟 번째 데이터 87과 열 번째 데이터 38을 서로 바꾼다.

2	1	4	3	5	6	7	8	9	10
7	4	19	15	28	30	35	38	42	87

<5단계>

나머지 데이터에 대하여 삽입정렬을 실행한다. 첫 번째와 두 번째의 데이터를 오름 차순 정렬하고 세 번째 데이터를 삽입정렬한 후 다시 세 번째 데이터까지를 subfile 로 놓고 네 번째 데이터를 삽입정렬하면 모든 데이터가 오름차순으로 정렬된다.

1	2	3	4	5	6	7	8	9	10
4	7	15	19	28	30	35	38	42	87

【학습정리】

1. 삽입정렬(insertion sort)

좌측으로부터 한 원소씩 제 자리에 삽입하는 방법으로서 삽입할 원소보다 큰 것은 우측으로 하나씩 이동하고 빈자리에 삽입원소를 집어넣는다.

- ① 별도의 배열이 필요 없는 제자리 정렬이다.
- ② 정렬 후에도 같은 키들의 상대 위치가 그대로 유지되는 안정적 정렬 알고리즘이다.
- ③ 삽입정렬에서 최악의 경우는 역순으로 정렬된 것이고 최선의 경우는 이미 제 순서대로 정렬 된 것일 때 이다.

* 삽입정렬의 실행 예

최초 자료 : 64, 28, 33, 76, 55, 12, 43

pass 1 : 28, 64, 33, 76, 55, 12, 43

pass 2 : 28, 33, 64, 76, 55, 12, 43

pass 3 : 28, 33, 64, 76, 55, 12, 43

pass 4 : 28, 33, 55, 64, 76, 12, 43

pass 5 : 12, 28, 33, 55, 64, 76, 43

pass 6 : 12, 28, 33, 43, 55, 64, 76

ㄹ

2. 셸 정렬(shell sort)

- 제자리에서 멀리 떨어진 원소가 제자리를 빨리 잡을 수 있도록 하기 위해 멀리 떨어진 원소들에 대하여 삽입정렬을 행한다는 것 셸 정렬은 삽입정렬을 확장한 것으로 멀리 떨어진 원소를 교환하여 속도를 빠르게 하였다. 셸 정렬 알고리즘의 시간복잡도는 최악 $O(n^2)$ 이 된다.