

자료구조응용

19. Sorting: insertion sort, quick sort

1. 다음과 같이 사용자로부터 데이터를 입력받아서 삽입정렬(insertion sort)을 수행한 결과를 출력하라. 각 레코드는 (key, name, grade)의 필드로 구성된다. 키값에 대하여 올림차순으로 삽입정렬을 하시오.

(1) 입력파일(input.txt)

7
10 송중기 95
35 조인성 89
25 김수미 59
50 홍길동 33
15 아이유 65
11 박용우 78
33 장윤정 67

(2) 실행순서

- ① 입력파일로부터 데이터를 읽어 들여 구조체 배열에 저장한다.
- ② key에 대해 내림차순으로 삽입정렬을 실행한다.
- ③ 정렬된 순서대로 (key, name, grade)를 화면에 출력한다.
- ④ 정렬결과를 파일(output.txt)에 저장한다.

```
void insert(element e, element a[], int i)
{ /* insert e into the ordered list a[1:i] such that the
   resulting list a[1:i+1] is also ordered, the array a
   must have space allocated for at least i+2 elements */
  a[0] = e;
  while (e.key < a[i].key)
  {
    a[i+1] = a[i];
    i--;
  }
  a[i+1] = e;
}
```

Program 7.4: Insertion into a sorted list

Program 7.5: Insertion sort

```
C:\WINDOWS\system32#c... - □ ×
```

```
<<<<<<<<<< Input List >>>>>>>>>>>>
```

(10,	송지국,	기성민,	95)
(35,	김민준,	이수민,	89)
(25,	김민준,	이수민,	59)
(50,	홍길동,	이수민,	33)
(15,	아이유,	유리,	65)
(11,	박용우,	정철,	78)
(33,	장영재,	정철,	67)

```
<<<<<<<<< Sorted List >>>>>>>>>>>>
```

(10,	송지국,	기성민,	95)
(11,	박용우,	정철,	78)
(15,	아이유,	유리,	65)
(25,	김민준,	이수민,	59)
(33,	장영재,	정철,	67)
(35,	김민준,	이수민,	89)
(50,	홍길동,	이수민,	33)

계속하려면 아무 키나 누르십시오 . . .

파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
(10,	송중기,	95)		
(11,	박우우,	78)		
(15,	아이유,	65)		
(25,	김수미,	59)		
(33,	장유정,	67)		
(35,	조민성,	89)		
(50,	홍길동,	33)		

2 입력리스트의 데이터를 파일로 입력받아 셸정렬 수행결과를 단계적으로 실행의 예와 같이 출력 하시오.

(1) 입력파일(input.txt)

11
10 8 6 20 4 3 22 1 0 15 16

 ※ 첫 줄은 레코드의 정렬할 키의 개수

(2) 실행순서

- ① 입력파일(input.txt)로부터 데이터를 읽어 들여 배열 list에 저장한다.
- ② 각 원소에 대해 셸 정렬을 실행한다.
- ③ gap의 크기에 따라 정렬된 결과를 출력한다.

(3) 실행 예

```
C:\WINDOWS\system32\cmd.exe
list[] data
10 8 6 20 4 3 22 1 0 15 16
gap is 5 =====>
3 8 1 0 4 10 22 6 20 15 16
gap is 3 =====>
0 4 1 3 6 10 15 8 20 22 16
gap is 1 =====>
0 1 3 4 6 8 10 15 16 20 22
계속하려면 아무 키나 누르십시오 . . .
```

3. 입력리스트의 데이터를 파일로 입력받아 퀵정렬 수행결과 및 quickSort 함수호출 회수를 구하여 출력하라. 단, 각 레코드는 하나의 int형 key 필드로 구성되어 있다.

(1) 입력파일(input.txt)

11
12 2 16 30 8 28 4 10 20 6 18

 ※ 첫 줄은 레코드의 정렬할 키의 개수

(2) 실행순서

- ① 입력파일(input.txt)로부터 데이터를 읽어 들여 구조체 배열 a에 저장한다.
- ② 각 레코드의 key에 대해 퀵정렬을 실행한다.
- ③ 정렬된 key값 및 quickSort 함수호출 회수를 화면에 출력하라.
- ④ 정렬결과를 파일(output.txt)에 저장한다.

```
void quickSort(element a[], int left, int right)
{
    /* sort a[left:right] into nondecreasing order
       on the key field; a[left].key is arbitrarily
       chosen as the pivot key; it is assumed that
       a[left].key <= a[right+1].key */
    int pivot, i, j;
    element temp;
    if (left < right) {
        i = left; j = right + 1;
        pivot = a[left].key;
        do { /* search for keys from the left and right
              sublists, swapping out-of-order elements until
              the left and right boundaries cross or meet */
            do i++; while (a[i].key < pivot);
            do j--; while (a[j].key > pivot);
            if (i < j) SWAP(a[i], a[j], temp);
        } while (i < j);
        SWAP(a[left], a[j], temp);
        quickSort(a, left, j-1);
        quickSort(a, j+1, right);
    }
}
```

Program 7.6: Quick sort

(3) 실행 예

