

자료구조응용

19. Sorting: insertion sort, quick sort (10점)

2020/5/20 (수)

1. 다음 입력 리스트에 대해 insertionSort(Program 7.5)의 for문에서 insert() 함수 실행 이후의 배열상태를 순서대로 기술하라.(2점)

입력 리스트 (12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18)

```
void insertionSort(element a[], int n)
{ /* sort a[1:n] into nondecreasing order */
    int j;
    for (j = 2; j <= n; j++) {
        element temp = a[j];
        insert(temp, a, j-1);
    }
}
```

Program 7.5: Insertion sort

- ※ 입력 리스트는 레코드의 키만 표현한 것이므로 배열 상태를 키에 대해서만 표현할 것
- ※ 연습장에 적은 후 사진을 찍어도 되며 그 결과를 보고서에 넣을 것

2. 다음과 같이 사용자로부터 데이터를 입력받아서 삽입정렬(insertion sort)을 수행한 결과를 출력하라. 각 레코드는 (key, name, grade)의 필드로 구성된다. (3점)

(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 삽입정렬을 실행한다.
- ③ 정렬된 순서대로 (key, name, grade)를 화면에 출력한다.
- ④ 정렬결과를 파일(output.txt)에 저장한다.

Program 7.4: Insertion into a sorted list

Program 7.5: Insertion sort

output - 메모장

파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
10, 승	(승기)	95		
11, 승	(승우)	78		
15, 마	(마유)	65		
25, 마	(마미)	59		
33, 최	(최정)	67		
35, 최	(최정)	89		
36, 최	(최정)	33		
50, 최	(최정)			

※ 연습장에 적은 후 사진을 찍어도 되며 그 결과를 보고서에 넣을 것

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

(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.

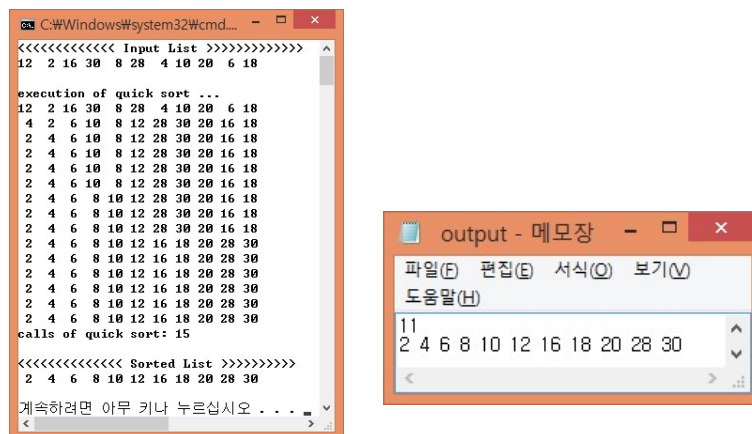
```

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



■ 제출 형식

- 솔루션/프로젝트 이름 : DS 19
- 소스파일 이름 : 2.c, 4.c
- 각 소스파일에 주석처리
“학번 이름”
“본인은 이 소스파일을 복사 없이 직접 작성하였습니다.”
- 솔루션 정리 후 솔루션 폴더를 “학번.zip”으로 압축하여 과제 게시판에 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 1차 마감 : 5월 21일(목) 자정
- 2차 마감 : 5월 22일(금) 자정(만점의 80%)