

자료구조응용

21. Bubble sort, radix sort

1. 다음과 같이 사용자로부터 데이터를 입력받아서 버블정렬(bubble sort)을 수행한 결과를 출력하라. 각 레코드는 (key, name, grade)의 필드로 구성된다. 키값에 대하여 내림차순으로 버블정렬을 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 버블정렬을 실행한다.
- ③ 단계별로 정렬된 순서대로 (key, name, grade)를 화면에 출력한다.
- ④ 정렬결과를 파일(output.txt)에 저장한다.
- ⑤ 아래의 코드는 참조만 하세요..

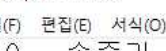
```
#define SWAP(x, y, t) ( (t)=(x), (x)=(y), (y)=(t) )
void bubble_sort(int list[], int n)
{   int i, j, temp;
    for(i=n-1; i>0; i--) {
        for(j=0; j<i; j++) // compare and change
            if(list[j]>list[j+1])
                SWAP(list[j], list[j+1], temp);
    }
}
```

(3) 실행 예



A screenshot of a Notepad window titled "input - 메모장". The window contains a list of names and scores, each on a new line. The text is as follows:

7			
10	송	중	95
35	조	인	89
25	김	수	59
50	홍	길	33
15	아	이	65
11	박	용	78
33	장	윤	67



```

( 10, 송기, 95 )
( 11, 박우, 78 )
( 15, 아이유, 65 )
( 25, 김수미, 59 )
( 33, 장윤정, 67 )
( 35, 조인성, 89 )
( 50, 홍길동, 33 )

```

```
C:\WINDOWS\system32\cmd.exe
<<<<<<<<<<< Input List >>>>>>>>>>
( 10, 송기민, 95 )
( 35, 조인성, 89 )
( 25, 김수미, 59 )
( 50, 이이유, 65 )
( 15, 이이유, 65 )
( 11, 박우철, 78 )
( 33, 이재정, 67 )
<<<<<<<<<< Step 1 : Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 25, 김수미, 59 )
( 35, 조인성, 89 )
( 15, 이이유, 65 )
( 11, 박우철, 78 )
( 33, 이재정, 67 )
( 50, 이이유, 33 )
<<<<<<<<<< Step 2 : Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 25, 김수미, 59 )
( 15, 이이유, 65 )
( 11, 박우철, 78 )
( 33, 이재정, 67 )
( 35, 조인성, 89 )
( 50, 이이유, 33 )
<<<<<<<<<< Step 3 : Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 15, 이이유, 65 )
( 11, 박우철, 78 )
( 25, 김수미, 59 )
( 33, 이재정, 67 )
( 35, 조인성, 89 )
( 50, 이이유, 33 )
<<<<<<<<<< Step 4 : Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 11, 이이유, 78 )
( 15, 이이유, 65 )
( 25, 김수미, 59 )
( 33, 이재정, 67 )
( 35, 조인성, 89 )
( 50, 이이유, 33 )
<<<<<<<<<< Step 5 : Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 11, 이이유, 78 )
( 15, 이이유, 65 )
( 25, 김수미, 59 )
( 33, 이재정, 67 )
( 35, 조인성, 89 )
( 50, 이이유, 33 )
<<<<<<<<<< Step 6 : Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 11, 이이유, 78 )
( 15, 이이유, 65 )
( 25, 김수미, 59 )
( 33, 이재정, 67 )
( 35, 조인성, 89 )
( 50, 이이유, 33 )
<<<<<<<<<< Result Sorted List >>>>>>>>>>
( 10, 송기민, 95 )
( 11, 이이유, 78 )
( 15, 이이유, 65 )
( 25, 김수미, 59 )
( 33, 이재정, 67 )
( 35, 조인성, 89 )
( 50, 이이유, 33 )
계속하려면 아무 키나 누르십시오 . . .
```

2. 다음 입력 리스트에 대해 기수정렬(radix sort)을 수행하고자 한다.

입력 리스트 (179 208 306 93 859 984 55 9 271 33)

<실행순서>

① 입력파일(input.txt)로부터 데이터를 읽어 들여 구조체 배열 a에 저장한다.

input.txt	
3 10	첫째 줄: d, n
179 208 306 93 859 984 55 9 271 33	둘째 줄: Key 리스트

※ element 타입은 key 필드만으로 구성된 구조체를 재정의한 것으로 가정한다.

② 각 레코드의 key에 대해 기수정렬을 실행한다.

※ (a)(b)(c)(d) 각 단계가 끝난 후의 체인의 Key값을 link 순서대로 화면에 출력하라.

③ 정렬 결과를 파일(output.txt)에 저장한다.

④ 아래의 코드는 참조만 하세요..

```
int radixSort(element a[], int link[], int d, int r, int n)
{
    /* sort a[1:n] using a d-digit radix-r sort, digit(a[i],j,r)
       returns the jth radix-r digit (from the left) of a[i]'s key
       each digit is in the range is [0,r); sorting within a digit
       is done using a bin sort */
    int front[r], rear[r]; /* queue front and rear pointers */
    int i, bin, current, first, last;
    /* create initial chain of records starting at first */
    first = 1;
    for (i = 1; i < n; i++) link[i] = i + 1;
    link[n] = 0;

    for (i = d-1; i >= 0; i--)
    {
        /* sort on digit i */
        /* initialize bins to empty queues */
        for (bin = 0; bin < r; bin++) front[bin] = 0;

        for (current = first; current; current = link[current])
        {
            /* put records into queues/bins */
            bin = digit(a[current], i, r);
            if (front[bin] == 0) front[bin] = current;
            else link[rear[bin]] = current;
            rear[bin] = current;
        }
        /* find first nonempty queue/bin */
        for (bin = 0; !front[bin]; bin++);
        first = front[bin]; last = rear[bin];

        /* concatenate remaining queues */
        for (bin++; bin < r; bin++)
            if (front[bin])
                {link[last] = front[bin]; last = rear[bin];}
        link[last] = 0;
    }
    return first;
}
```

Program 7.14: LSD radix sort

<실행결과>

```
input - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
3 10
179 208 306 93 859 984 55 9 271 33
```

```
output - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
9 33 | 55 93 179 208 271 306 859 984
```

```
C:\WINDOWS\system32\cmd.exe
***** initial chain *****
link: [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9] [10]
a: 179 208 306 93 859 984 55 9 271 33
first: 1
result: 179 208 306 93 859 984 55 9 271 33

***** pass 1 *****
link: [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9] [10]
a: 179 208 306 93 859 984 55 9 271 33
first: 9
result: 271 93 33 984 55 306 208 179 859 9

rear: [ 0] [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9]
front: 0 9 0 4 6 7 3 0 2 8

***** pass 2 *****
link: [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9] [10]
a: 179 208 306 93 859 984 55 9 271 33
first: 3
result: 306 208 9 33 55 859 271 179 984 93

rear: [ 0] [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9]
front: 8 0 0 10 0 5 0 1 6 4

***** pass 3 *****
link: [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9] [10]
a: 179 208 306 93 859 984 55 9 271 33
first: 8
result: 9 33 55 93 179 208 271 306 859 984

rear: [ 0] [ 1] [ 2] [ 3] [ 4] [ 5] [ 6] [ 7] [ 8] [ 9]
front: 4 1 9 3 0 0 0 0 5 6
8 1 2 3 0 0 0 0 5 6
계속하려면 아무 키나 누르십시오 . . .
```

3. 다음 입력 리스트에 대해 기수정렬(radix sort)을 수행하고자 한다.

입력 리스트 (179 208 306 93 859 984 55 9 271 33)

<실행순서>

① 입력파일(input.txt)로부터 데이터를 읽어 들여 구조체 배열 a에 저장한다.

input.txt	
3 10	첫째 줄: d, n
179 208 306 93 859 984 55 9 271 33	둘째 줄: Key 리스트

② 각 레코드의 key에 대해 기수정렬을 실행한다.

③ 정렬 결과에 대하여 실행의 예와 같이 단계적으로 큐의 내용을 함께 출력하시요

<실행결과>

```
C:\WINDOWS\system32\cmd.exe
input data ---
179 208 306 93 859 984 55 9 271 33
step 0 자리 ---
Queue의 내용 ---
0 1 2 3 4 5 6 7 8 9
9
33 859
271 93 984 55 306 208 179
271 93 33 984 55 306 208 179 859 9
step 1 자리 ---
Queue의 내용 ---
0 1 2 3 4 5 6 7 8 9
9
208 859 179
306 33 55 271 984 93
306 208 9 33 55 859 271 179 984 93
step 2 자리 ---
Queue의 내용 ---
0 1 2 3 4 5 6 7 8 9
93
55
33 271
9 179 208 306 859 984
9 33 55 93 179 208 271 306 859 984
result ---
9 33 55 93 179 208 271 306 859 984
계속하려면 아무 키나 누르십시오 . . .
```

■ 제출 형식

- 공학인증 시스템(ABEEK)에 과제를 올릴 때 제목:
- 1차 제출: 학번 이름 DS-21(1), 2차 제출: 학번 이름 DS-21(2)
- 솔루션 이름 : DS-21
- 프로젝트 이름 : 1, 2, 3
- 실행화면을 캡처하여 한글파일에 추가 후 솔루션 폴더에 포함.
- 한글 파일명 : 학번_이름_실습결과.hwp
- 솔루션 폴더를 압축하여 게시판에 제출할 것.
- 압축 파일 명: 학번_이름_DS-21.zip
- 제출은 2회걸쳐 가능(수정 시간 기준으로 처리)