

자료구조응용

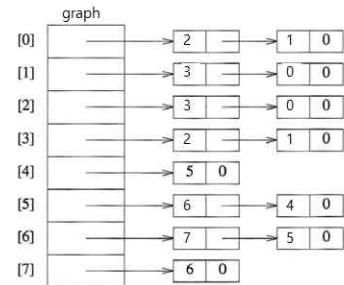
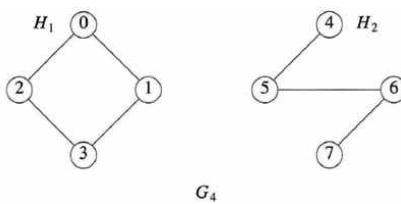
20.connected components, insertion sort (10점)

2025.11.19

1. 입력된 무방향그래프의 connected component를 출력하는 프로그램을 작성하라. (5점)

(1) 입력파일(input.txt) 및 자료구조

8 7
0 1
0 2
1 3
2 3
4 5
5 6
6 7



(2) 구현 세부사항

```
void connected(void)
/* determine the connected components of a graph */
int i;
for (i = 0; i < n; i++)
    if(!visited[i]) {
        dfs(i);
        printf("\n");
    }
}
```

Program 6.3: Connected components

(3) 실행 예

자료구조응용 19. 1번 그래프

Graph G4

```
C:\Windows\system32\cmd.exe
<<<<<<<< Adjacency List >>>>>>>>
graph[0] : 2 1
graph[1] : 3 0
graph[2] : 6 5 0
graph[3] : 7 1
graph[4] : 7 1
graph[5] : 7 2
graph[6] : 7 2
graph[7] : 6 5 4 3

<<<<<<<< Connected Components >>>>>>>
connected component 1 : 0 2 6 7 5 4 1 3
계속하려면 아무 키나 누르십시오 . . .

```

```
C:\Windows\system32\cmd.exe
<<<<<<<< Adjacency List >>>>>>>>
graph[0] : 2 1
graph[1] : 3 0
graph[2] : 3 0
graph[3] : 2 1
graph[4] : 5
graph[5] : 6 4
graph[6] : 7 5
graph[7] : 6

<<<<<<<< Connected Components >>>>>>>
connected component 1 : 0 2 3 1
connected component 2 : 4 5 6 7
계속하려면 아무 키나 누르십시오 . . .

```

2. 다음 입력 리스트에 대해 insertionSort(Program 7.5)의 for문에서 insert() 함수 실행 이 후의 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

※ 기술한 결과를 보고서에 넣을 것

3. 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 흥길동 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.

```

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

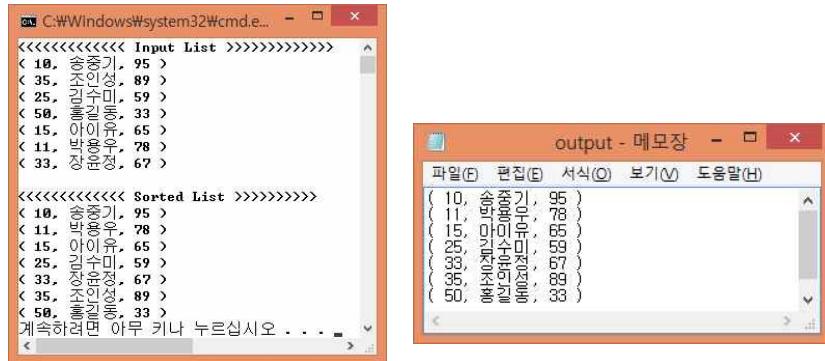
```

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

(3) 실행 예



■ 제출 형식

- 솔루션 이름 : DS 20
- 프로젝트 이름 : 1, 3
- 각 소스파일에 주석처리
“학번 이름”
“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”
- 제출 파일
 - ① 소스코드와 실행 결과가 보이도록 화면을 캡쳐한 보고서 파일(“학번.pdf”)
※ 한글 [파일 → pdf로 저장하기...] 메뉴 사용
 - ② C 소스 파일을 하나의 디렉터리에 모아 압축한 파일 (“학번.zip”)
※ “학번.pdf”와 “학번.zip”을 하나로 압축하지 말고 별도 파일로 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 1차 마감 : 수업일 자정
- 2차 마감 : 수업 익일 자정(만점의 60%, 반올림)
- 문항 별로 1차 2차 나눠서 제출할 수 없으며, 최종 제출 시간에 따라 1차, 2차로 구분함
- **4시 40분에 제출 상황 체크함**
- **완료된 과제를 제출하지 않고 일찍 퇴실한 학생은 0점 처리하겠음**