

**(문제1) (7-6) 링크드 최대힙**

연결리스트에 최대 힙을 표현하는 linked max heap을 구현하시오.

다음 예시의 입력값들을 하나씩 추가하여 최대힙을 구성하여 출력하고, 최대값(루트)를 하나씩 삭제하는 과정을 보이시오.

<예시>

- 7-6 예시 [10, 20, 30, 40, 56, 35, 60, 70, 85]

- 연습문제 7-1 입력값

- 193쪽, 예제 입력값

(조건: parent 필드는 부모 노드로 연결되는 링크를 의미한다)

<노드의 필드 구조>

llink	data	parent	rlink
-------	------	--------	-------

<헤드 노드의 구조>

헤드 노드의 llink는 루트 노드를, rlink는 last node를 가리킨다. (초기값은 None)

llink	data	None	rlink
-------	------	------	-------

<기능>

1. 노드 추가 : 마지막 노드 다음 위치에 추가한 후 최대힙을 재구성한다.

(중복 값이 없는 경우에만 추가)

2. 노드 삭제(루트 삭제) : 루트 삭제 후 마지막 노드가 대신한 후 힙 재구성한다.

3. 노드 추가 또는 삭제 연산 후 최종 최대힙의 모든 노드 값 postorder로 출력한다.

### (문제2) (9-9) 최소비용 신장트리 (Prim 방법)

프로그램 9.3(Kruskal 방법)을 수정하여 Prim의 방법으로 최소 비용 신장 트리를 탐색하는 프로그램을 작성하시오.

- 연습문제 9-1, 9-2, 9-3의 예시 그래프를 사용하여 작성한 프로그램을 테스트한다.
- 탐색 시작 노드를 사용자로부터 입력 받는다. (시작 노드가 변경되는 경우도 처리할 것)
- 최소 비용 신장 트리에 간선이 추가되는 순서대로 출력한다.

(\* 교재의 코드를 활용하지 않을 경우 0점)

### (문제3) Floyd-Warshall 알고리즘을 구현하시오. (30점)

구현 프로그램은 각 비용행렬과 각 정점 간 경로를 출력해야 한다. (아래의 조건을 만족해야 하며, 그림 10.2와 연습문제 10-2, 10-3의 그래프로 각각 테스트할 것)

(조건)

비용행렬(dist)과 동일한 크기의 p배열(prev)을 단계별로 출력한다.

최종 비용행렬에서 두 정점간 최단 경로(정점들의 시퀀스)를 재귀적으로 역추적한다.

\* 경로 (a, b)에 대한 p배열에 경유 정점 k1이 존재하면 경로는 (a, k1, b)이 된다.

출력 예시) 그림 10.2 그래프

dist: A -1

-----

0: 0 8 inf 1  
1: inf 0 1 inf  
2: 4 inf 0 inf  
3: inf 2 9 0

dist: A 0

-----

0: 0 8 inf 1  
1: inf 0 1 inf  
2: 4 12 0 5  
3: inf 2 9 0

.....

<최단 경로>

s~d: 0 1, path [0, 3, 1] len = 3  
s~d: 0 2, path [0, 3, 1, 2] len = 4  
s~d: 0 3, path [0, 3] len = 1  
s~d: 1 0, path [1, 2, 0] len = 5  
s~d: 1 2, path [1, 2] len = 1  
.....

#### (문제4) 위상 정렬 (20점)

방향 그래프의 간선 정보를 입력 받아 그래프의 인접 리스트를 생성하고 위상 정렬을 수행하시오.

(그림 10.3의 예시 그래프 및 추가 방향 그래프에 대해서 테스트를 수행한다)

(\* 프로그램 9.2를 활용하고 이에 기반하여 위상 정렬 코드를 작성할 것)

(조건)

- 초기 인접 리스트를 출력한다.
- 위상 정렬 수행 과정에서 큐의 상태 및 각 정점의 진입 차수 정보를 단계별로 갱신하여 출력한다.
- 위상 정렬 순서를 출력한다.
- 그래프에 사이클이 있는 예시와 없는 예시를 각각 테스트할 것

#### <문항별 개별평가표>

문제	제목	완성도 (0~100%) (A)	배점(B)	평가점수 (A*B)
1	링크드 최대 힙		25	
2	최소비용신장트리 (Prim)		25	
3	Floyd-Warshall 최단 경로 출력		30	
4	위상 정렬		20	
총점			100	