

알고리즘 및 실습

프로그래밍 #2

제출일: 6 월 12 일 (일) 자정

파트 #1: “우선순위 큐”를 힙 자료구조를 사용하여 구현한다.

최소한 아래 기능을 수행할 수 있어야 한다.

- **Insert(Q, k)**
키 값 k 를 큐 Q 에 추가
- **Delete(Q, k)**
키 값 k 를 큐 Q 에서 제거
- **ExtractMin(Q)**
가장 키 값이 작은 원소의 키 값을 제거하고, 그 키 값을 리턴한다.
- **IncreaseKey(Q,k,v)**
키 값 k (혹은 인덱스 k)를 가진 원소의 키 값을 증가된 키 값 v 로 바꾼다.
- **DecreaseKey(Q,k,v)**
키 값 k (혹은 인덱스 k)를 가진 원소의 키 값을 감소된 키 값 v 로 바꾼다.

- 우선순위 큐를 Polymorphic Type 으로 구현할 수 있다면, 추가 점수 부여함.

- 그렇지 않다면, 최소값이 우선순위가 높은 우선순위 큐를 구현하라.

- 파트 #2 를 구현하지 않는다면 작동의 예를 보인다.

파트 #2: Prim's MST(Minimum Spanning Tree) 알고리즘을 구현한다.

Prim's MST algorithm --

```
Q ← V           // G=(V,E,W) ; Q is a priority queue maintaining V - S
key[v] ← ∞       for all v ∈ V
key[s] ← 0       for some s ∈ V
1  while Q ≠ ∅
2    do u ← EXTRACT-MIN(Q)
3      for each v ∈ Adj[u]      // Adjacency list
4        do if (v ∈ Q and w(u,v) > key[v] )
5          then key[v] ← w(u, v) // Decrease-Key operation
6            p[v] ← u
```

문제 A: 키값을 저장 관리하기 위해서,

- 1) 배열을 사용하는 경우
- 2) (파트 1 에서 구현한) 우선순위 큐를 사용한 각 경우에 대하여,
deleteMin() 과 decreaseKey() 문이 수행되는 횟수를 카운터해
비교하라.

문제 B: 위의 알고리즘에는 경로 정보가 저장된다. MST 를 찾은 경우에
저장된 경로 정보를 사용하여, 그 경로를 출력하는 프로그램을 작성하라.

문제 C: 테스트 케이스를 하나 만들어 결과를 출력해 보인다.

보고서

- 본인이 사용한 입력 그래프 (그림)
- 위 각 질문에 답하는 결과물
- 결과에 대한 본인의 생각

제출:

- 소스코드와 보고서를 zip 하여 클래스넷에 제출

Enjoy!!!