

알고리즘 설계과제 2

[내비게이션 프로그램 설계]

1. 과제 개요

인하내비에서는 도로교통상황을 파악하여, 출발지부터 목적지까지의 최적경로를 안내하는 내비게이션 애플리케이션을 서비스하고 있다. 한편, 태풍으로 인해 우리나라의 많은 지역이 침수되었다. 출발지에서 목적지로 가는 경로 중 침수된 지역을 피하여 안전하고 가장 빠른 경로를 안내하려 한다. 이를 위해 출발지와 목적지 간의 안전한 최단경로를 강의시간에 배운 Dijkstra 알고리즘을 이용하여 계산하자. 먼저 도로공사에서 각 지역들과 지역 간의 도로 정보들을 입력받은 뒤, 아래에 제시된 기능들을 수행하는 프로그램을 구현하시오. 단, 모든 도로는 양방향 도로라 가정하고, Dijkstra 알고리즘에서 decreaseKey() 함수는 정점의 최단거리가 더 작아지는 경우에만 수행된다.

2. 정보 형식

(1) 지역 정보

- | |
|---|
| ① 지역번호 (기준키, 유일함): 100,000 ~ 999,999 사이의 정수. |
| ② 지역이름: 공백 없는 20Byte 이내의 문자열. 예) "Juan-dong" |
| ③ 침수여부: 1 또는 0. 예) "1" (침수된 경우), "0" (침수되지 않은 경우) |

(2) 두 지역 간 도로 정보

- | |
|----------------------------|
| ① 지역번호1: 하나의 지역번호 |
| ② 지역번호2: 지역번호1과 다른 지역번호 |
| ③ 도로의 거리: 1 ~ 5,000 사이의 정수 |

3. 수행할 기능

(1) 데이터 입력받기

프로그램을 실행하면 무향그래프(undirected graph) 정보를 표준입력(standard input)으로 입력받아, 인접리스트 기반(adjacency list representation) 그래프자료구조를 생성하여 관리한다.

첫 번째 줄에는 지역정보의 수 $n(2 \leq n \leq 10^5)$ 과 지역 간의 도로 정보의 수 $m(1 \leq m \leq 2 \times 10^5)$, 질의의 수 $q(1 \leq q \leq 50)$ 가 공백으로 구분되어 주어진다.

이후 n 개의 줄을 통해, 지역정보가 아래와 같이 한 줄로 주어지고, 각 필드는 공백으로 구분된다.

필드1	필드2	필드3
지역번호	지역이름	침수여부

이후 m 개의 줄을 통해, 지역 간 도로 정보가 아래와 같이 한 줄로 주어지며, 각 필드는 공백으로 구분된다.

필드1	필드2	필드3
지역번호1	지역번호2	도로의 거리

단, 최대거리는 10^6 이며, 출발지에서 목적지까지의 거리가 10^6 을 초과하면 출발지에서 목적지까지 갈 수 없다고 가정한다.

(2) 두 지역 간의 최단거리 찾기

- 질의형식: "A S D"

A: 최단거리 찾기에 대한 질의

S: 출발 지역번호

D: 목적 지역번호

- 출력형식: "N W SN DN" or "None"

N: 출발지부터 목적지까지 Tree vertex들의 set에 추가된 지역들의 수

W: 출발지부터 목적지까지의 최단거리

SN: 출발 지역이름

DN: 목적 지역이름

- 설명: 출발지와 목적지가 주어지면, 두 지역 간의 최단거리를 계산한다. 만약 계산 중 최단거리가 동일한 후보정점이 2개 이상 존재한다면, 지역번호가 더 작은 정점(높은 우선순위)을 우선적으로 계산한다. 출발지부터 목적지까지의 최단경로를 계산했을 때, Tree vertex가 된 지역들의 수(N), 최단거리, 출발 지역이름, 목적 지역이름을 표준출력(standard output)으로 공백으로 구분하여 출력한다. 만약 출발지 또는 목적지가 침수되어 있거나 두 지역의 최단거리가 10^6 을 초과하는 경우, "None"을 출력한다.

(3) 두 지역 간의 최단경로 찾기

- 질의형식: "B S D"

B: 최단경로 찾기에 대한 질의

S: 출발 지역번호

D: 목적 지역번호

- 출력형식: "N V_0 V_1 ... V_k " or "None"

N: 출발지부터 목적지까지 Tree vertex들의 set에 추가된 지역들의 수

V_0 V_1 ... V_k : 출발지부터 목적지까지의 최단경로 (V_i : 지역번호 ($0 \leq i \leq k$), V_0 : 출발지, V_k : 목적지)

- 설명: 출발지와 목적지가 주어지면, 두 지역 간의 최단경로를 계산한다. 만약 계산 중 최단거리가 동일한 후보정점이 2개 이상 존재한다면, 지역번호가 더 작은 정점(높은 우선순위)을 우선적으로 계산한다. 출발지부터 목적지까지의 최단경로를 계산했을 때, Tree vertex가 된 지역들의 수(N)와 최단경로를 표준출력으로 공백으로 구분하여 출력한다. 만약 출발지 또는 목적지가 침수되어 있거나 두 지역의 최단거리가 10^6 을 초과하는 경우, "None"을 출력한다.

4. 입출력 제한사항

- (1) 프로그램의 한 입력에 대해 정점 정보는 최대 10^5 개, 간선 정보는 최대 2×10^5 개, 질의는 최대 50개가 입력된다.
- (2) 채점서버의 입력들에 대해 총 2초의 제한시간 이내에 수행되어야 한다.
- (3) 제시한 입출력 형식대로 표준입출력을 사용하여 처리한다.
- (4) 문제에서 설명되지 않은 예외처리를 해야 할 질의는 입력되지 않는다.

5. 프로그램 입출력 예

(1) 표준입출력 예시

파란색: 프로그램 표준입력 내용

빨간색: 프로그램 표준출력 내용

```
=====
9 11 7
105691 Juan-dong 0
124129 Madu-dong 0
109711 Baeksuk-dong 0
108353 Balsan-dong 0
112317 Daechi-dong 0
114748 Gamma-dong 0
111062 Alpha-dong 0
128522 Beta-dong 0
101234 Charlie-dong 1
105691 111062 450
105691 124129 1000
111062 114748 950
114748 124129 500
124129 109711 5000
124129 108353 900
109711 108353 3
109711 112317 2
128522 105691 3
128522 101234 1
101234 114748 1
B 112317 111062
8 112317 109711 108353 124129 114748 111062
A 101234 114748
None
B 101234 109711
None
B 114748 112317
8 114748 124129 108353 109711 112317
A 128522 114748
5 1403 Beta-dong Gamma-dong
B 124129 109711
4 124129 108353 109711
A 109711 124129
4 903 Baeksuk-dong Madu-dong
(프로그램 종료)
```

(2) 표준입출력 예시에서의 각 입/출력 내용 (구현을 검증할 때, 활용할 것)

Input
9 11 7 105691 Juan-dong 0 124129 Madu-dong 0 109711 Baeksuk-dong 0 108353 Balsan-dong 0 112317 Daechi-dong 0 114748 Gamma-dong 0 111062 Alpha-dong 0 128522 Beta-dong 0 101234 Charlie-dong 1 105691 111062 450 105691 124129 1000 111062 114748 950 114748 124129 500 124129 109711 5000 124129 108353 900 109711 108353 3 109711 112317 2 128522 105691 3 128522 101234 1 101234 114748 1 B 112317 111062 A 101234 114748 B 101234 109711 B 114748 112317 A 128522 114748 B 124129 109711 A 109711 124129

Output
8 112317 109711 108353 124129 114748 111062 None None 8 114748 124129 108353 109711 112317 5 1403 Beta-dong Gamma-dong 4 124129 108353 109711 4 903 Baeksuk-dong Madu-dong

6. 주의 사항 (지키지 않으면, 0점 또는 감점)

(1) 개발언어 및 채점서버 환경

- ① 허용된 개발언어: C, C++
- ② OS: Ubuntu 18.04 (64-bit)
- ③ gcc 버전: gcc (Ubuntu 7.3.0-16ubuntu3) 7.3.0 (c++14 지원)

(2) 제출 파일

- ① 보고서 (2차 과제에서는 과제의 부담을 줄여주고자, 없음)
 - (a) 파일형식: ~~아래아한글 문서(.hwp), MS Word 문서(.doc, .docx), PDF 문서(.pdf)~~
 - (b) 양식: ~~첨부된 파일 참조~~
 - (c) 파일명: ~~"학번_이름"으로 통일 (예: 12059876_홍길동)~~
 - ② 소스코드 (채점서버에 제출)
 - (a) 하나의 소스코드 파일로 구현하여 서버에 제출 및 채점 (예: 12059876_GildongHong.cpp)
(채점서버에서 파일명에 대한 한글 인식이 되지 않기 때문에, 파일명은 영문이름으로 제출)
 - (b) 소스코드에 반드시 주석이 기재돼 있어야 함
- ※ 파일명 형식 어길 시, 감점

(3) 기타

- ① 제출 마감: 5월 30일 일요일 23:59까지 (마감 후에 제출되는 과제는 받지 않음)
- ② 마감 직전에는 I-class 및 채점 서버에 학생들이 많이 몰리기 때문에 혼잡할 수 있으니, 마감 시간보다 1시간 정도 이전에 여유 있게 제출하는 것을 권장
- ③ 제출 후에는 제대로 제출되었는지 반드시 확인
- ④ 부정행위가 적발될 경우, 베낀 학생과 원본을 제공한 학생 모두 0점 처리 및 성적의 총점에서 -10점 감점. 인터넷이나 책(교재, 참고서 등)의 소스코드, 예전에 제출했던 과제 등을 베껴서 제출해도 부정행위로 처리