2022-2 HI-ARC 중급스터디

5주차. 최단 경로1

이지은 (leeju1013)

목차

1. 우선순위 큐
   - 11279. 최대 힙
   - 1927. 최소 힙

2. 다익스트라
   - 1753. 최단경로

힙(Heap) 복습

- What?

- 최댓값이나 최솟값을 빠르게 찾아내도록 만들어진 자료구조

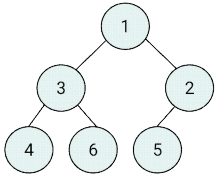
- 완전 이진 트리 + heap property

- 루트 노드의 우선순위 > 해당 서브트리 노드들의 우선순위

- **우선순위 큐** 구현에 사용되는 자료구조

힙(Heap) 복습

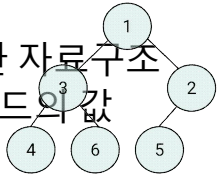
- 최소 힙(Min heap)
  - 최솟값을 빠르게 찾기위한 자료구조
  - 부모 노드의 값 <= 자식 노드의 값



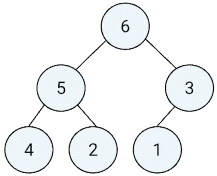
Min heap

사진 출처

- 최대 힙(Max heap)
  - 최댓값을 빠르게 찾기위한 자료구조
  - 부모 노드의 값 >=자식 노드의 값



Min heap



Max Heap

힙(Heap) 복습

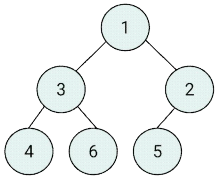
- 시간 복잡도?

- 트리의 높이 : O(logN)에 비례 (∵완전 이진 트리)

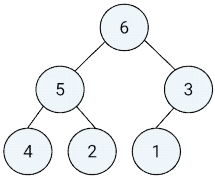
- 삽입(Insertion) : O(logN)

- 삭제(Deletion) : O(logN)

- 접근(Top access) : O(1)



Min heap



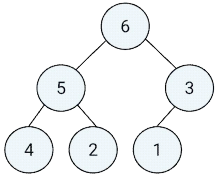
Max Heap

1. 우선순위 큐(Priority Queue)

- What?

- 입력된 순서에 상관없이,
  우선순위가 가장 높은 자료가 가장 먼저 꺼내지는 자료구조

- **힙**(heap)으로 구현



Max Heap

- How?

- #include <queue>
- push(x) : 원소 추가
- pop() : top에 있는 원소 제거
- top() : 우선순위가 가장 높은 원소 확인

최대 힙

삼급

실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람
1 초 (추가 시간 없음) (하단 참고)	256 MB	49779	22649	17720

문제

널리 잘 알려진 자료구조 중 최대 힙이 있다. 최대 힙을 이용하여 다음과 같은 연산을 지원하는 프로그램을 작성하시오.

1. 배열에 자연수 x를 넣는다.
2. 배열에서 가장 큰 값을 출력하고, 그 값을 배열에서 제거한다.

프로그램은 처음에 비어있는 배열에서 시작하게 된다.

입력

첫째 줄에 연산의 개수 N(1 ≤ N ≤ 100,000)이 주어진다. 다음 N개의 줄에는 연산에 대한 정보를 나타내는 정수 x가 주어진다. 만약 x가 자연수라면 배열에 x라는 값을 넣는(추가하는) 연산이고, x가 0이라면 배열에서 가장 큰 값을 출력하고 그 값을 배열에서 제거하는 경우이다. 입력되는 자연수는 2^{31}보다 작다.

출력

입력에서 0이 주어진 회수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 가장 큰 값을 출력하라고 한 경우에는 0을 출력하면 된다.

예제 입력 1

```
13
0
1
2
0
0
3
2
1
0
0
0
0
0
```

예제 출력 1

```
0
2
1
3
2
1
0
0
```

입력

첫째 줄에 연산의 개수 N(1 ≤ N ≤ 100,000)이 주어진다. 다음 N개의 줄에는 연산에 대한 정보를 나타내는 정수 x가 주어진다. 만약 x가 자연수라면 배열에 x라는 값을 넣는(추가하는) 연산이고, x가 0이라면 배열에서 가장 큰 값을 출력하고 그 값을 배열에서 제거하는 경우이다. 입력되는 자연수는 2^{31}보다 작다.

출력

입력에서 0이 주어진 회수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 가장 큰 값을 출력하라고 한 경우에는 0을 출력하면 된다.

예제 입력 1

```
13
0
1
2
0
0
3
2
1
0
0
0
0
0
```

예제 출력 1

```
0
2
1
3
2
1
0
0
```

```cpp
#include <iostream>
#include <queue>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false); cin.tie(NULL);

    priority_queue<int>pq; //우선순위 큐 선언

    int n; cin>>n;
    while(n--){
        int x; cin>>x;
        if(x) // x!=0
            pq.push(x);
        else{ // x==0
            if(pq.empty()) cout <<"0\n";
            else{
                cout << pq.top()<<'\n';
                pq.pop();
            }
        }
    }
    return 0;
}
```

최소 힙

성공

2 실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람
1 초 (추가 시간 없음) (하단 참고)	128 MB	50556	23491	18466

문제

널리 잘 알려진 자료구조 중 최소 힙이 있다. 최소 힙을 이용하여 다음과 같은 연산을 지원하는 프로그램을 작성하시오.

1. 배열에 자연수 x를 넣는다.
2. 배열에서 가장 작은 값을 출력하고, 그 값을 배열에서 제거한다.

프로그램은 처음에 비어있는 배열에서 시작하게 된다.

입력

첫째 줄에 연산의 개수 N(1 ≤ N ≤ 100,000)이 주어진다. 다음 N개의 줄에는 연산에 대한 정보를 나타내는 정수 x가 주어진다. 만약 x가 자연수라면 배열에 x라는 값을 넣는(추가하는) 연산이고, x가 0이라면 배열에서 가장 작은 값을 출력하고 그 값을 배열에서 제거하는 경우이다. x는 2^{31}보다 작은 자연수 또는 0이고, 음의 정수는 입력으로 주어지지 않는다.

출력

입력에서 0이 주어진 횟수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 가장 작은 값을 출력하라고 한 경우에는 0을 출력하면 된다.

예제 입력 1

```
9
0
12345678
1
2
0
0
0
0
32
```

예제 출력 1

```
0
1
2
12345678
0
```

```cpp
#include <iostream>
#include <queue>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false); cin.tie(NULL);

    priority_queue<int>pq; //우선순위 큐 선언(max heap)

    int n; cin>>n;
    while(n--){
        int x; cin>>x;
        if(x) // x!=0
            pq.push(-x); // 부호 뒤집고 넣기
        else { // x==0
            if(pq.empty()) cout <<"0\n";
            else{
                cout << -pq.top()<<'\n'; // 부호 원상태로 돌리고 출력
                pq.pop();
            }
        }
    }
    return 0;
}
```

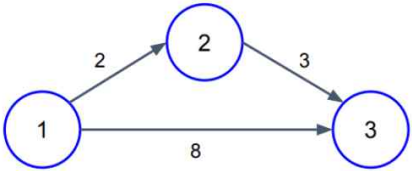
예제 입력 1	예제 출력 1
9 0 12345678 1 2 0 0 0 0 32	0 1 2 12345678 0

```cpp
#include <iostream>
#include <queue>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false); cin.tie(NULL);

    priority_queue<int, vector<int>, greater<int>>pq; //우선순위 큐 선언(min heap)

    int n; cin>>n;
    while(n--){
        int x; cin>>x;
        if(x) // x!=0
            pq.push(x);
        else { // x==0
            if(pq.empty()) cout <<"0\n";
            else{
                cout << pq.top()<<'\n';
                pq.pop();
            }
        }
    }
    return 0;
}
```

예제 입력 1	예제 출력 1
9 0 12345678 1 2 0 0 0 0 32	0 1 2 12345678 0

최단경로 알고리즘

- BFS
- 다익스트라
- 벨만포드
- 플로이드 와샬



2. 다익스트라(Dijkstra's algorithm)

- What?

- 그래프의 한 정점에서 다른 정점들까지의 *최단거리*를 구하는 알고리즘

- 가중치가 음수인 간선이 존재할 때는 사용 불가

- How?

- 우선순위 큐 사용

2. 다익스트라(Dijkstra's algorithm)

- How?

1. 아직 방문하지 않은 정점들 중
   거리가 가장 짧은 정점u를 하나 선택해 방문한다.

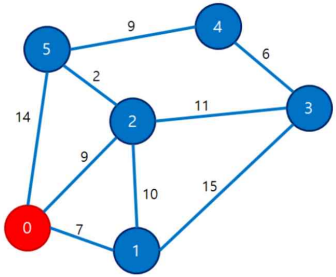
2. 해당 정점 u와 인접하고 아직 방문하지 않은 정점들의
   거리를 갱신한다.

3. 1과 2를 v번 반복한다.

1. 아직 방문하지 않은 정점들 중 거리가 가장 짧은 정점u를 하나 선택해 방문한다.

2. 해당 정점 u와 인접하고 아직 방문하지 않은 정점들의 거리를 갱신한다.

3. 1과 2를 v번 반복한다.

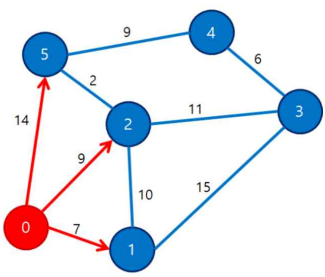


0	1	2	3	4	5
0	∞	∞	∞	∞	∞

시작점인 0번 정점에서
나머지 정점들로의 최단거리 구하기!

초기값으로 시작점은 0,
나머지는 max num으로 설정.

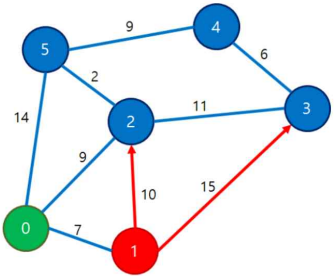
dist가 가장 작은 0번 정점 방문.



0	1	2	3	4	5
0	7	9	∞	∞	14

0번 정점과 인접하고 아직 방문하지 않은 정점인 1,2,5번 정점의 거리를 갱신.

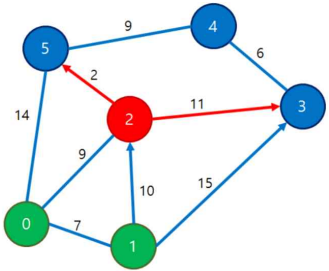
=> 0번 정점을 통해 k번 정점으로 가는 거리인 dist[0] + dist[0][k]가 dist[k]보다 작다면 갱신.



0	1	2	3	4	5
0	7	9	22	∞	14

아직 방문하지 않은 정점 중 dist가 가장 작은 1번 정점 방문.

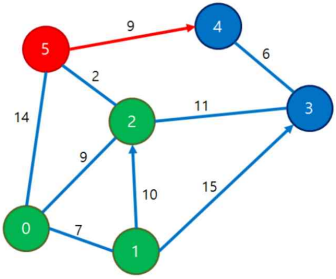
1번 정점과 인접하고 아직 방문하지 않은 정점인 2,3번 정점의 거리를 갱신. (3번 정점만 갱신됨)



0	1	2	3	4	5
0	7	9	20	∞	11

아직 방문하지 않은 정점 중 dist가 가장 작은 2번 정점 방문.

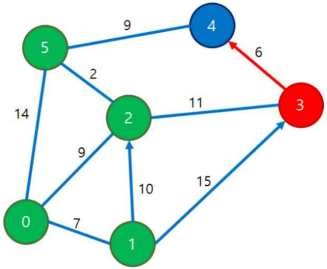
2번 정점과 인접하고 아직 방문하지 않은 정점인 3,5번 정점의 거리를 갱신.



0	1	2	3	4	5
0	7	9	20	20	11

아직 방문하지 않은 정점 중 dist가 가장 작은 5번 정점 방문.

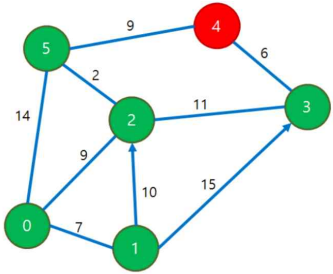
5번 정점과 인접하고 아직 방문하지 않은 정점인 4번 정점의 거리를 갱신.



0	1	2	3	4	5
0	7	9	20	20	11

아직 방문하지 않은 정점 중 dist가 가장 작은 3번 정점 방문.
(dist[3]==dist[4]인데 둘 중 아무거나 방문)

3번 정점과 인접하고 아직 방문하지 않은 정점인 4번 정점의 거리를 갱신.
(갱신되지 않음)



0	1	2	3	4	5
0	7	9	20	20	11

마지막으로 4번 정점만 남았을 때는 나머지 정점을 다 방문한 상태이므로 아무것도 하지 않음.

=> dist 배열에 0번 정점으로부터의 최단 거리가 저장되어있음.
(매 루프에서 초록색 정점의 dist값은 이미 최단거리이고 앞으로 절대 바뀌지 않음)

Q1. 시간복잡도?
O(ElogV)

Q2. 제약조건?
간선의 가중치가 양수일때만 성립함.

Q3. 어떻게 구현할까?
매 루프마다 dist가 가장 작은 정점을 찾는 방법?

-> dist값들을 다 비교해서 찾으면 O(V)이고
루프는 v-1번 실행되므로
총 O(V^2)

-> 우선순위 큐 사용!!
O(ElogV)

쉬는 시간

1. 우선순위 큐
   - 11279. 최대 힙
   - 1927. 최소 힙

2. 다익스트라
   - 1753. 최단경로

2. 다익스트라(Dijkstra's algorithm)

- 시간복잡도

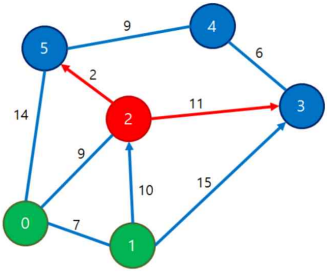
- 정점 개수가 V, 간선 개수가 E일 때 O(ElogV)

우선순위 큐를 이용해서
방문하지 않은 정점 중 거리가 가장 짧은 정점을 뽑아
인접한 정점들의 최단거리를 업데이트하는 과정은 최대
간선 개수 E만큼 이루어짐.

방문하지 않은 정점 중
시작점에서 가장 가까운 정점을 뽑는 것은
우선순위 큐의 특성상 O(logN)임. (N은 우선순위 큐의 원소 개수)

우선순위 큐의 최대 원소 개수는 간선 개수와 같고 대개의 경우
그래프에서 간선의 개수 E는 V^2보다 작으므로
O(logN) = O(logE) = O(logV^2) = O(2logV) = O(logV)임.

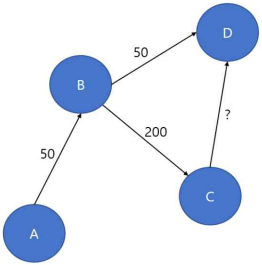
따라서 다익스트라 알고리즘의 총 시간 복잡도는 O(ElogV)



0	1	2	3	4	5
0	7	9	20	∞	11

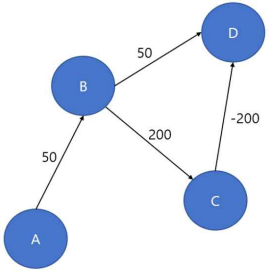
2. 다익스트라(Dijkstra's algorithm)

- 가중치가 음수인 간선이 존재할 때는 사용 불가



A에서 D로 가는 최단거리를 구할 때,
다익스트라를 이용하면
A->B->D로 가게 됨.
(C를 거치지 않음)

다익스트라는 근시안적 관점을 유지하기 때문에 가중치가 감소할 수도 있다는 것을 고려하지 못함.



왼쪽 그림처럼
가중치가 음수라서 실제 최단거리가 되는 A->B->C->D 경로를 구할 수 없게 됨.

음수 가중치가 존재할 때는 모든 경로를 전부 고려해야 함.

따라서 다익스트라 알고리즘의 그리디 관점으로 최단거리를 구할 수 없음.
(벨만포드 알고리즘 사용)

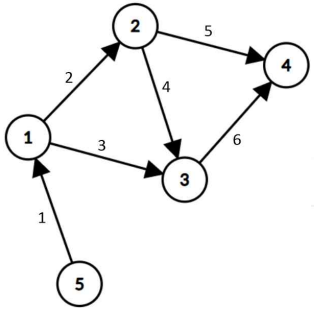
(사진 출처)

최단경로

성공

골드 IV

시간 제한	메모리 제한	제출
1 초	256 MB	145671



예제 입력 1

```
5 6
1
5 1 1
1 2 2
1 3 3
2 3 4
2 4 5
3 4 6
```

예제 출력 1

```
0
2
3
7
INF
```

문제

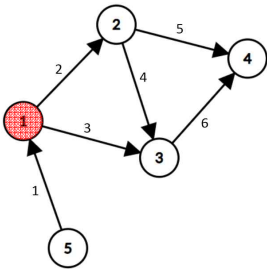
방향그래프가 주어지면 주어진 시작점에서 다른 모든 정점으로의 최단 경로를 구하는 프로그램을 작성하시오. 단, 모든 간선의 가중치는 10 이하의 자연수이다.

입력

첫째 줄에 정점의 개수 V와 간선의 개수 E가 주어진다. (1 ≤ V ≤ 20,000, 1 ≤ E ≤ 300,000) 모든 정점에는 1부터 V까지 번호가 매겨져 있다고 가정한다. 둘째 줄에는 시작 정점의 번호 K(1 ≤ K ≤ V)가 주어진다. 셋째 줄부터 E개의 줄에 걸쳐 각 간선을 나타내는 세 개의 정수 (u, v, w)가 순서대로 주어진다. 이는 u에서 v로 가는 가중치 w인 간선이 존재한다는 뜻이다. u와 v는 서로 다르며 w는 10 이하의 자연수이다. 서로 다른 두 정점 사이에 여러 개의 간선이 존재할 수도 있음에 유의한다.

출력

첫째 줄부터 V개의 줄에 걸쳐, i번째 줄에 i번 정점으로의 최단 경로의 경로값을 출력한다. 시작점 자신은 0으로 출력하고, 경로가 존재하지 않는 경우에는 INF를 출력하면 된다.



예제 입력 1

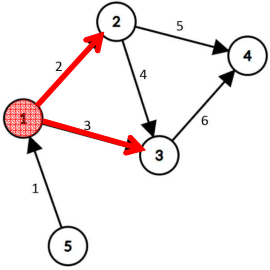
```
5 6
1
5 1 1
1 2 2
1 3 3
2 3 4
2 4 5
3 4 6
```

1	2	3	4	5
0	INF	INF	INF	INF

시작점인 1번 정점에서
나머지 정점들로의 최단거리 구하기!

초기값으로 시작점은 1,
나머지는 max num으로 설정.

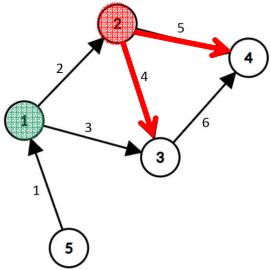
dist가 가장 작은 1번 정점 방문.



1	2	3	4	5
0	2	3	INF	INF

PQ { 2,2 }, { 3,3 }

1번 정점과 인접하고 아직 방문하지 않은 정점인 2,3번 정점의 거리를 갱신.

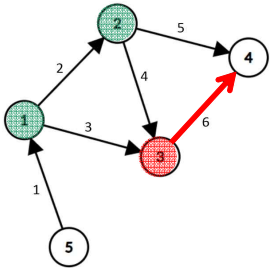


1	2	3	4	5
0	2	3	7	INF

PQ { 3,3 }, { 7,4 }

아직 방문하지 않은 정점 중 dist가 가장 작은 2번 정점 방문.

2번 정점과 인접하고 아직 방문하지 않은 정점인 3,4번 정점의 거리를 갱신. (4번 정점만 갱신됨)

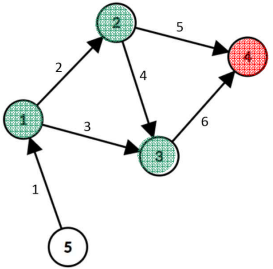


1	2	3	4	5
0	2	3	7	INF

PQ { 7,4 }

아직 방문하지 않은 정점 중 dist가 가장 작은 3번 정점 방문.

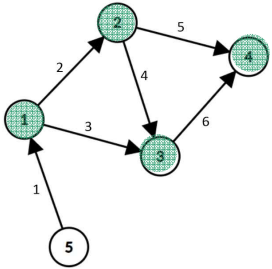
3번 정점과 인접하고 아직 방문하지 않은 정점인 4번 정점의 거리를 갱신.



1	2	3	4	5
0	2	3	7	INF

PQ

마지막으로 4번 정점만 남았을 때는 나머지 정점을 다 방문한 상태이므로 아무것도 하지 않음.

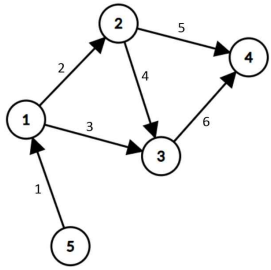


1	2	3	4	5
0	2	3	7	INF

예제 출력 1

```
0
2
3
7
INF
```

```cpp
#include <iostream>
#include <queue>
#include <vector>
#include <climits>
using namespace std;
vector<pair<int,int>> adj[20001]; //그래프 정보(간선의 가중치) 저장
vector<int> dist(20001, INT_MAX); //시작정점으로부터의 최단거리
int v,e,k; //정점, 간선, 시작정점

void dijkstra(int k){
    dist[k]=0; //시작점은 0으로 출력
    priority_queue<pair<int,int>,vector<pair<int,int>>,greater<>>pq; //(가중치 기준) 최소힙
    pq.push({0,k});

    while(!pq.empty()){ //PQ가 비면 종료
        int cur_dist = pq.top().first;
        int cur = pq.top().second;
        pq.pop();

        if(dist[cur] < cur_dist) continue; //방문했던 정점이면 패스

        for(auto &k : adj[cur]){
            int next_dist = k.first;
            int next = k.second;
            if(dist[next] > next_dist+cur_dist){
                //거리가 갱신될 경우
                dist[next]=next_dist+cur_dist;
                pq.push({dist[next],next});
            }
        }
    }
    // 결과 출력
    for(int i=1; i<v+1; i++){
        if(dist[i] == INT_MAX) cout<<"INF\n";
        else cout<<dist[i]<<'\n';
    }
}
int main() {
    cin>>v>>e>>k;
    while(e--){ //간선의 개수만큼 가중치 정보 입력받기
        int a,b,c; cin>>a>>b>>c;
        adj[a].push_back({c,b}); //a에서 b로 가는 가중치 c
    }
    dijkstra(k); //시작점 k
    return 0;
}
```



감사합니다

- 필수 문제 11279. 최대 힙
  1927. 최소 힙
  1753. 최단경로
- 연습 문제 11286. 절댓값 힙
  11000. 강의실배정
  1446. 지름길
  1504. 특정한 최단 경로
  11779. 최소비용 구하기
  4485. 녹색 옷 입은 애가 젤다지?
- 11월 9일(화요일) 저녁 6시 T702