

그라프 2

소스코드

최백준 choi@startlink.io

C++14

```

1 #include <cstdio>
2 #include <vector>
3 #include <queue>
4 using namespace std;
5 vector<int> a[32001];
6 int ind[32001];
7 int main() {
8     int n,m;
9     scanf("%d %d",&n,&m);
10    for (int i=0; i<m; i++) {
11        int x,y;
12        scanf("%d %d",&x,&y);
13        a[x].push_back(y);
14        ind[y] += 1;
15    }
16    priority_queue<int> q;
17    for (int i=1; i<=n; i++) {
18        if (ind[i] == 0) {
19            q.push(-i);
20        }
21    }
22    while (!q.empty()) {
23        int x = -q.top();
24        q.pop();
25        printf("%d ",x);
26        for (int i=0; i<a[x].size(); i++) {
27            int y = a[x][i];
28            ind[y] -= 1;
29            if (ind[y] == 0) {
30                q.push(-y);
31            }
32        }
33    }
34    printf("\n");
35    return 0;
36 }

```

← 인접 리스트

> ind[i] = i의 in-degree

⊗ → ⊙

C++의 pq : Max Heap

② < 3
② > -3

O(E)

결과

메모리

시간

코드 길이

맞았습니다!!

2980 KB

44 ms

733 B

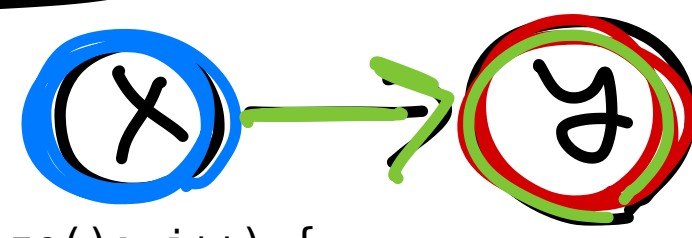
C++14

```
1 #include <cstdio>
2 #include <vector>
3 #include <queue>
4 using namespace std;
5 vector<int> a[10001];
6 int ind[10001];
7 int work[10001];
8 int d[10001];
9 int main() {
10     int n;
11     scanf("%d",&n);
12     for (int i=1; i<=n; i++) {
13         scanf("%d",&work[i]);
14         int cnt;
15         scanf("%d",&cnt);
16         for (int j=0; j<cnt; j++) {
17             int x;
18             scanf("%d",&x);
19             a[x].push_back(i);
20             ind[i] += 1;
21         }
22     }
23     queue<int> q;
24     for (int i=1; i<=n; i++) {
25         if (ind[i] == 0) {
26             q.push(i);
27             d[i] = work[i];
28         }
29     }
30     while (!q.empty()) {
31         int x = q.front();
32         q.pop();
33         for (int i=0; i<a[x].size(); i++) {
34             int y = a[x][i];
35             ind[y] -= 1;
36             if (d[y] < d[x]+work[y]) {
37                 d[y] = d[x]+work[y];
38             }
39             if (ind[y] == 0) {
40                 q.push(y);
41             }
42         }
43     }
44     int ans = 0;
45     for (int i=1; i<=n; i++) {
46         if (ans < d[i]) {
47             ans = d[i];
48         }
49     }
50     printf("%d\n",ans);
51     return 0;
52 }
```

→ $d[i] = T$ 를 마치는
가장 빠른 시간

이런

BFS 시작??



$O(E)$

C++14

```
1 #include <iostream>
2 #include <queue>
3 #include <vector>
4 using namespace std;
5 struct Edge {
6     int to;
7     int cost;
8     bool operator < (const Edge &other) const {
9         return cost > other.cost;
10    }
11 };
12 vector<Edge> a[1001];
13 bool c[1001];
14 int main() {
15     ios_base::sync_with_stdio(false);
16     cin.tie(nullptr);
17     int n, m;
18     cin >> n >> m;
19     for (int i=0; i<m; i++) {
20         int from, to, cost;
21         cin >> from >> to >> cost;
22         a[from].push_back(Edge({to, cost}));
23         a[to].push_back(Edge({from, cost}));
24     }
25     c[1] = true;
26     priority_queue<Edge> q;
27     for (Edge e : a[1]) {
28         q.push(e);
29     }
30     int ans = 0;
31     while (!q.empty()) {
32         Edge e = q.top();
33         q.pop();
34         if (c[e.to] == true) {
35             continue;
36         }
37         c[e.to] = true;
38         ans += e.cost;
39         int x = e.to;
40         for (Edge ee : a[x]) {
41             q.push(ee);
42         }
43     }
44     cout << ans << '\n';
45     return 0;
46 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

8068 KB

56 ms

962 B

C++14

Kruskal

```
1 #include <cstdio>
2 #include <algorithm>
3 #include <vector>
4 #include <queue>
5 using namespace std;
6 struct Edge {
7     int from, to, cost;
8     bool operator < (const Edge &other) const {
9         return cost < other.cost;
10    }
11 };
12 int p[10001];
13 int Find(int x) {
14     if (x == p[x]) {
15         return x;
16     } else {
17         return p[x] = Find(p[x]);
18     }
19 }
20 void Union(int x, int y) {
21     x = Find(x);
22     y = Find(y);
23     p[x] = y;
24 }
25 int main() {
26     int n, m;
27     scanf("%d %d", &n, &m);
28     for (int i=1; i<=n; i++) {
29         p[i] = i;
30     }
31     vector<Edge> a(m);
32     for (int i=0; i<m; i++) {
33         scanf("%d %d %d", &a[i].from, &a[i].to, &a[i].cost);
34     }
35     sort(a.begin(), a.end());
36     int ans = 0;
37     for (int i=0; i<m; i++) {
38         Edge e = a[i];
39         int x = Find(e.from);
40         int y = Find(e.to);
41         if (x != y) {
42             Union(e.from, e.to);
43             ans += e.cost;
44         }
45     }
46     printf("%d\n", ans);
47     return 0;
48 }
```

$O(E \log E)$
 $O(E)$

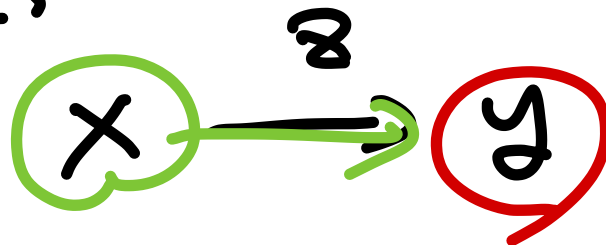
C++14

```
1 #include <cstdio>
2 #include <vector>
3 using namespace std;
4 struct Edge {
5     int from;
6     int to;
7     int cost;
8 };
9 int dist[501];
10 int inf = 1000000000;
11 int main() {
12     int t;
13     int n,m;
14     scanf("%d %d",&n,&m);
15     vector<Edge> a(m);
16     for (int i=0; i<m; i++) {
17         scanf("%d %d %d",&a[i].from, &a[i].to, &a[i].cost);
18     }
19     for (int i=1; i<=n; i++) {
20         dist[i] = inf;
21     }
22     dist[1] = 0;
23     bool negative_cycle = false;
24     for (int i=1; i<=n; i++) {
25         for (int j=0; j<m; j++) {
26             int x = a[j].from;
27             int y = a[j].to;
28             int z = a[j].cost;
29             if (dist[x] != inf && dist[y] > dist[x]+z) {
30                 dist[y] = dist[x]+z;
31                 if (i == n) {
32                     negative_cycle = true;
33                 }
34             }
35         }
36     }
37     if (negative_cycle) {
38         printf("-1\n");
39     } else {
40         for (int i=2; i<=n; i++) {
41             if (dist[i] == inf) dist[i] = -1;
42             printf("%d\n",dist[i]);
43         }
44     }
45     return 0;
46 }
47
```

모든 간지

$n-1$ 번 $\rightarrow N$

방문 횟수



결과

메모리

시간

코드 길이

맞았습니다!!

1228 KB

4 ms

1025 B

C++14

```
1 #include <cstdio>
2 #include <vector>
3 using namespace std;
4 struct Edge {
5     int to;
6     int cost;
7     Edge(int to, int cost) : to(to), cost(cost) {
8     }
9 };
10 vector<Edge> a[1001];
11 int dist[1001];
12 bool check[1001];
13 int inf = 1000000000;
14 int main() {
15     int n;
16     scanf("%d",&n);
17     int m;
18     scanf("%d",&m);
19     for (int i=0; i<m; i++) {
20         int x,y,z;
21         scanf("%d %d %d",&x,&y,&z);
22         a[x].push_back(Edge(y,z));
23     }
24     int start, end;
25     scanf("%d %d",&start,&end);
26     for (int i=1; i<=n; i++) {
27         dist[i] = inf;
28     }
29     dist[start] = 0;
30     for (int k=0; k<n-1; k++) {
31         int m = inf+1;
32         int x = -1;
33         for (int i=1; i<=n; i++) {
34             if (check[i] == false && n > dist[i]) {
35                 m = dist[i];
36                 x = i;
37             }
38         }
39         check[x] = true;
40         for (int i=0; i<a[x].size(); i++) {
41             int y = a[x][i].to;
42             if (dist[y] > dist[x] + a[x][i].cost) {
43                 dist[y] = dist[x] + a[x][i].cost;
44             }
45         }
46     }
47     printf("%d\n",dist[end]);
48     return 0;
49 }
```

→ 인접리스트
← false
← ∞

초기화
V-1
V-2
V
V-1
V-2

X → ⑤

C++14

```
1 #include <stdio>
2 #include <vector>
3 #include <queue>
4 using namespace std;
5 struct Edge {
6     int to;
7     int cost;
8     Edge(int to, int cost) : to(to), cost(cost) {
9     }
10 };
11 vector<Edge> a[20001];
12 int dist[20001];
13 bool check[20001];
14 int inf = 1000000000;
15 int main() {
16     int n,m;
17     scanf("%d %d",&n,&m);
18     int start;
19     scanf("%d",&start);
20     for (int i=0; i<m; i++) {
21         int x,y,z;
22         scanf("%d %d %d",&x,&y,&z);
23         a[x].push_back(Edge(y,z));
24     }
25     for (int i=1; i<=n; i++) {
26         dist[i] = inf;
27     }
28     dist[start] = 0;
29     priority_queue<pair<int,int>> q;
30     q.push(make_pair(0, start));
31     while (!q.empty()) {
32         auto p = q.top();
33         q.pop();
34         int x = p.second;
35         if (check[x]) {
36             continue;
37         }
38         check[x] = true;
39         for (int i=0; i<a[x].size(); i++) {
40             int y = a[x][i].to;
41             if (dist[y] > dist[x] + a[x][i].cost) {
42                 dist[y] = dist[x] + a[x][i].cost;
43                 q.push(make_pair(-dist[y], y));
44             }
45         }
46     }
47     for (int i=1; i<=n; i++) {
48         if (dist[i] >= inf) {
49             printf("INF\n");
50         } else {
51             printf("%d\n",dist[i]);
52         }
53     }
54     return 0;
55 }
```

결과	메모리	시간	코드 길이
맞았습니다!!	8244 KB	160 ms	1227 B

C++14

```
1 #include <stdio>
2 #include <vector>
3 #include <set>
4 using namespace std;
5 struct Edge {
6     int to;
7     int cost;
8     Edge(int to, int cost) : to(to), cost(cost) {
9     }
10 };
11 vector<Edge> a[20001];
12 int dist[20001];
13 int inf = 1000000000;
14 int main() {
15     int n,m;
16     scanf("%d %d",&n,&m);
17     int start;
18     scanf("%d",&start);
19     for (int i=0; i<m; i++) {
20         int x,y,z;
21         scanf("%d %d %d",&x,&y,&z);
22         a[x].push_back(Edge(y,z));
23     }
24     for (int i=1; i<=n; i++) {
25         dist[i] = inf;
26     }
27     dist[start] = 0;
28     set<pair<int,int>> s;
29     s.insert(make_pair(0, start));
30     while (!s.empty()) {
31         auto p = *s.begin();
32         s.erase(s.begin());
33         int x = p.second;
34         for (int i=0; i<a[x].size(); i++) {
35             int y = a[x][i].to;
36             if (dist[y] > dist[x] + a[x][i].cost) {
37                 if (dist[y] != inf) {
38                     s.erase(s.find(make_pair(dist[y],y)));
39                 }
40                 dist[y] = dist[x] + a[x][i].cost;
41                 s.insert(make_pair(dist[y],y));
42             }
43         }
44     }
45     for (int i=1; i<=n; i++) {
46         if (dist[i] >= inf) {
47             printf("INF\n");
48         } else {
49             printf("%d\n",dist[i]);
50         }
51     }
52     return 0;
53 }
```

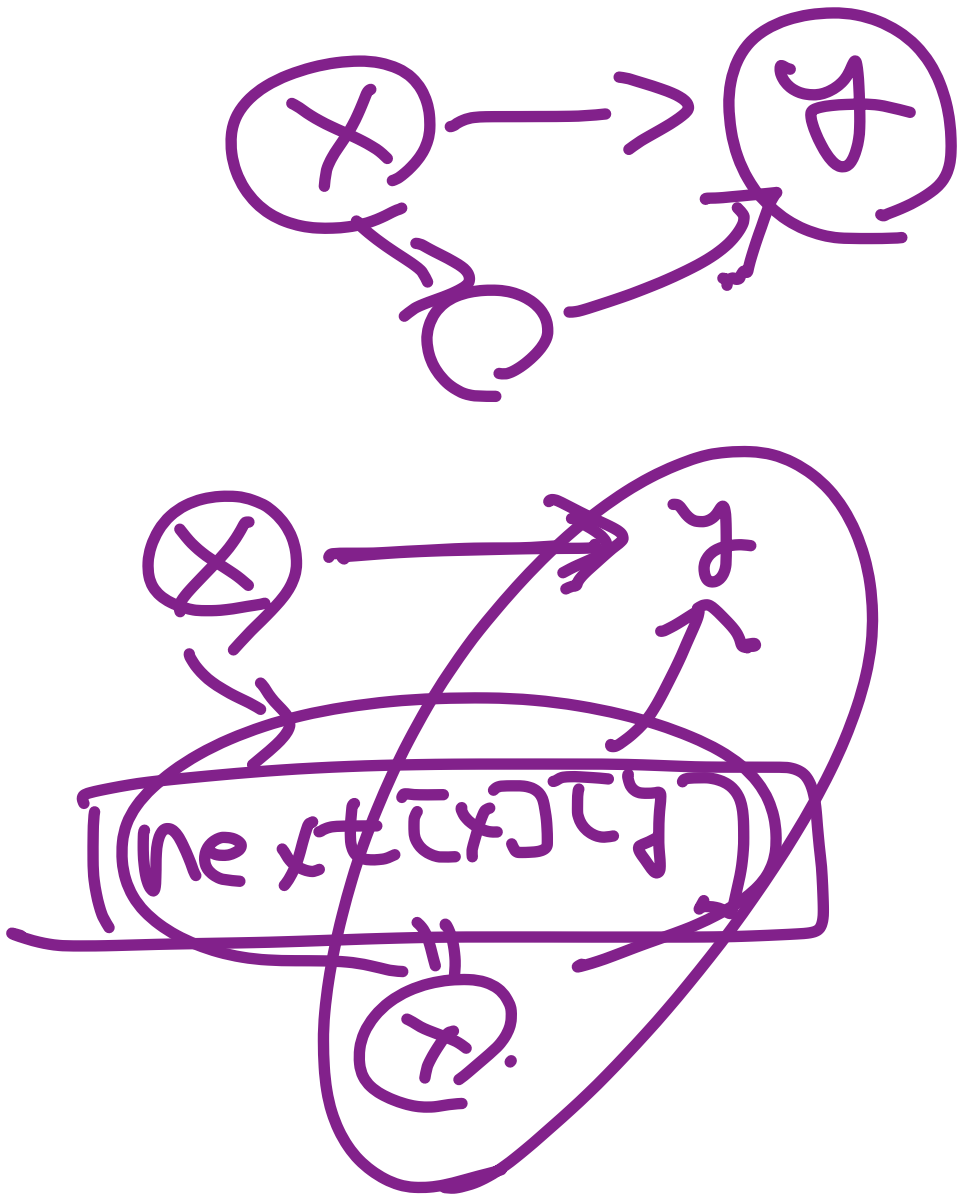
(dist, v)

X → y

결과	메모리	시간	코드 길이
맞았습니다!!	8224 KB	152 ms	1245 B

C++14

```
1 #include <cstdio>
2 #include <queue>
3 using namespace std;
4 #define next _next
5 int a[1001][1001];
6 int next[1001][1001];
7 int distance[1001];
8 bool check[1001];
9 const int inf = 1000000000;
10 void path(int x, int y) {
11     if (next[x][y] == -1) {
12         printf("0\n");
13         return;
14     }
15     queue<int> q;
16     q.push(x);
17     while (x != y) {
18         x = next[x][y];
19         q.push(x);
20     }
21     printf("%d ", q.size());
22     while (!q.empty()) {
23         printf("%d ", q.front());
24         q.pop();
25     }
26     printf("\n");
27 }
28 int main() {
29     int n, m;
30     scanf("%d %d", &n, &m);
31     for (int i=1; i<=n; i++) {
32         for (int j=1; j<=n; j++) {
33             if (i == j) {
34                 a[i][j] = 0;
35             } else {
36                 a[i][j] = inf;
37             }
38             next[i][j] = -1;
39         }
40     }
41     for (int i=0; i<m; i++) {
42         int from, to, cost;
43         scanf("%d %d %d", &from, &to, &cost);
44         if (a[from][to] > cost) {
45             a[from][to] = cost;
46             next[from][to] = to;
47         }
48     }
49     for (int k=1; k<=n; k++) {
50         for (int i=1; i<=n; i++) {
51             for (int j=1; j<=n; j++) {
52                 if (a[i][j] > a[i][k] + a[k][j]) {
53                     a[i][j] = a[i][k] + a[k][j];
54                     next[i][j] = next[i][k];
55                 }
56             }
57         }
58     }
59     for (int i=1; i<=n; i++) {
60         for (int j=1; j<=n; j++) {
61             if (a[i][j] == inf) {
62                 printf("%d ", 0);
63             } else {
64                 printf("%d ", a[i][j]);
65             }
66         }
67         printf("\n");
68     }
69     for (int i=1; i<=n; i++) {
70         for (int j=1; j<=n; j++) {
71             if (i == j) {
72                 printf("0\n");
73             } else if (a[i][j] == inf) {
74                 printf("0\n");
75             } else {
76                 path(i, j);
77             }
78         }
79     }
80     return 0;
81 }
```



결과

메모리

시간

코드 길이

맞았습니다!!

9060 KB

44 ms

1853 B

C++14

1 #include <cstdio>

2 #include <queue>

3 #include <vector>

4 using namespace std;

5 struct Edge {

6 int to, cost;

7 Edge() {

8 }

9 Edge(int to, int cost) : to(to), cost(cost) {

10 }

11 };

12 vector<Edge> a[501];

13 const int inf = 1000000000;

14 int d[501];

15 bool c[501];

16 int cnt[501];

17 int main() {

18 int n, m;

19 scanf("%d %d",&n,&m);

20 for (int i=0; i<m; i++) {

21 int from, to, cost;

22 scanf("%d %d %d",&from, &to, &cost);

23 a[from].push_back(Edge(to, cost));

24 }

25 for (int i=1; i<=n; i++) {

26 d[i] = inf;

27 }

28 d[1] = 0;

29 queue<int> q;

30 q.push(1);

31 c[1] = true;

32 while (!q.empty()) {

33 int from = q.front();

34 c[from] = false;

35 q.pop();

36 for (Edge &e : a[from]) {

37 int to = e.to;

38 int cost = e.cost;

39 if (d[to] > d[from] + cost) {

40 d[to] = d[from] + cost;

41 if (c[to] == false) {

42 q.push(to);

43 c[to] = true;

44 cnt[to] += 1;

45 if (cnt[to] >= n) {

46 printf("-1\n");

47 return 0;

48 }

49 }

50 }

51 }

52 }

53 for (int i=2; i<=n; i++) {

54 if (d[i] == inf) {

55 printf("-1\n");

56 } else {

57 printf("%d\n",d[i]);

58 }

59 }

60 return 0;

61 }

62

결과

메모리

시간

코드 길이

맞았습니다!!

1244 KB

8 ms

1362 B

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.