

그래프 2 (연속)

소스코드

최백준 choi@startlink.io

C++14

```
1 #include <iostream>
2 #include <cstdio>
3 #include <vector>
4 #include <queue>
5 using namespace std;
6 vector<pair<int,int>> a[10001];
7 vector<pair<int,int>> b[10001];
8 int ind[10001];
9 int ind2[10001];
10 int d[10001];
11 bool c[10001];
12 int main() {
13     int n,m;
14     scanf("%d %d",&n,&m);
15     for (int i=0; i<m; i++) {
16         int t1,t2,t3;
17         scanf("%d %d %d",&t1,&t2,&t3);
18         a[t1].push_back(make_pair(t2,t3));
19         b[t2].push_back(make_pair(t1,t3));
20         ind[t2] += 1;
21         ind2[t1] += 1;
22     }
23     int st,ed;
24     scanf("%d %d",&st,&ed);
25     queue<int> q;
26     for (int i=1; i<=n; i++) {
27         if (ind[i] == 0) {
28             q.push(i);
29         }
30     }
31     while (!q.empty()) {
32         int x = q.front();
33         q.pop();
34         for (int k=0; k<a[x].size(); k++) {
35             int y = a[x][k].first;
36             d[y] = max(d[y],d[x]+a[x][k].second);
37             ind[y] -= 1;
38             if (ind[y] == 0) {
39                 q.push(y);
40             }
41         }
42     }
43     printf("%d\n",d[ed]);
44     for (int i=1; i<=n; i++) {
45         ind[i] = ind2[i];
46     }
47     int ans = 0;
48     c[ed] = true;
49     for (int i=1; i<=n; i++) {
50         if (ind[i] == 0) {
51             q.push(i);
52         }
53     }
54     while (!q.empty()) {
55         int x = q.front();
56         q.pop();
57         for (int k=0; k<b[x].size(); k++) {
58             int y = b[x][k].first;
59             if (c[x] && d[x]-d[y] == b[x][k].second) {
60                 c[y] = true;
61                 ans += 1;
62             }
63             ind[y] -= 1;
64             if (ind[y] == 0) {
65                 q.push(y);
66             }
67         }
68     }
69     printf("%d\n",ans);
70     return 0;
71 }
```

C++14

```
1 #include <vector>
2 #include <algorithm>
3 #include <cstdio>
4 using namespace std;
5 int a[1001][1001];
6 int dist[1001];
7 bool check[1001];
8 int inf = 100000000;
9 int n;
10 vector<int> dijkstra(int start) {
11     for (int i=1; i<=n; i++) {
12         dist[i] = inf;
13         check[i] = false;
14     }
15     dist[start] = 0;
16     for (int k=0; k<n-1; k++) {
17         int m = inf+1;
18         int x = -1;
19         for (int i=1; i<=n; i++) {
20             if (check[i] == false && m > dist[i]) {
21                 m = dist[i];
22                 x = i;
23             }
24         }
25         check[x] = true;
26         for (int i=1; i<=n; i++) {
27             if (dist[i] > dist[x] + a[x][i]) {
28                 dist[i] = dist[x] + a[x][i];
29             }
30         }
31     }
32     return vector<int>(dist, dist+n+1);
33 }
34 int main() {
35     scanf("%d", &n);
36     for (int i=1; i<=n; i++) {
37         for (int j=1; j<=n; j++) {
38             a[i][j] = inf;
39         }
40     }
41     int m;
42     scanf("%d", &m);
43     for (int i=0; i<m; i++) {
44         int x, y, z;
45         scanf("%d %d %d", &x, &y, &z);
46         if (a[x][y] > z) {
47             a[x][y] = z;
48         }
49         if (a[y][x] > z) {
50             a[y][x] = z;
51         }
52     }
53     int v1, v2;
54     scanf("%d %d", &v1, &v2);
55     vector<int> dstart = dijkstra(1);
56     vector<int> d1 = dijkstra(v1);
57     vector<int> d2 = dijkstra(v2);
58     int ans = dstart[v1] + d1[v2] + d2[n];
59     int ans2 = dstart[v2] + d2[v1] + d1[n];
60     if (ans > ans2) {
61         ans = ans2;
62     }
63     if (ans >= inf) {
64         ans = -1;
65     }
66     printf("%d\n", ans);
67     return 0;
68 }
```

무한대 (10억)
2147483647
 $10^9 + 10^9 = 20^9 < \text{inf}$
30억 (inf)

시작지
v1
v2

$1 \rightarrow v_1 \rightarrow v_2 \rightarrow N$
 $1 \rightarrow v_2 \rightarrow v_1 \rightarrow N$

C++14

```
1 #include <cstdio>
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[10001];
12 priority_queue<int> dist[10001];
13 int inf = 1000000000;
14 int main() {
15     int n,m,k;
16     scanf("%d %d %d",&n,&m,&k);
17     for (int i=0; i<m; i++) {
18         int x,y,z;
19         scanf("%d %d %d",&x,&y,&z);
20         a[x].push_back(Edge(y,z));
21     }
22     dist[1].push(0);
23     priority_queue<pair<int,int>> q;
24     q.push(make_pair(0, 1));
25     while (!q.empty()) {
26         auto p = q.top();
27         q.pop();
28         int cur = -p.first;
29         int x = p.second;
30         for (int i=0; i<a[x].size(); i++) {
31             int y = a[x][i].to;
32             if (dist[y].size() < k || dist[y].top() > cur + a[x][i].cost) {
33                 if (dist[y].size() == k)
34                     dist[y].pop();
35             }
36             dist[y].push(cur+a[x][i].cost);
37             q.push(make_pair(-(cur+a[x][i].cost), y));
38         }
39     }
40 }
41
42 for (int i=1; i<=n; i++) {
43     if (dist[i].size() != k) {
44         printf("-1\n");
45     } else {
46         printf("%d\n",dist[i].top());
47     }
48 }
49 return 0;
50 }
```

Max Heap

← 인접리스트

int dist

~~dist[1].push(0);~~

크기

← 출력

결과	메모리	시간	코드 길이
맞았습니다!!	5620 KB	116 ms	1228 B

C++14

```

1 #include <iostream>
2 using namespace std;
3 int d[20][20];
4 bool unused[20][20];
5 int main() {
6     int n;
7     cin >> n;
8     for (int i=0; i<n; i++) {
9         for (int j=0; j<n; j++) {
10             cin >> d[i][j];
11         }
12     }
13     for (int k=0; k<n; k++) {
14         for (int i=0; i<n; i++) {
15             if (i == k) continue;
16             for (int j=0; j<n; j++) {
17                 if (i == j) continue;
18                 if (k == j) continue;
19                 if (d[i][j] > d[i][k] + d[k][j]) {
20                     cout << -1 << '\n';
21                     return 0;
22                 }
23                 if (d[i][j] == d[i][k] + d[k][j]) {
24                     unused[i][j] = true;
25                 }
26             }
27         }
28     }
29     int ans = 0;
30     for (int i=0; i<n; i++) {
31         for (int j=0; j<n; j++) {
32             if (unused[i][j] == false) {
33                 ans += d[i][j];
34             }
35         }
36     }
37     ans /= 2;
38     cout << ans << '\n';
39     return 0;
40 }

```

결과

메모리

시간

코드 길이

맞았습니다!!

1988 KB

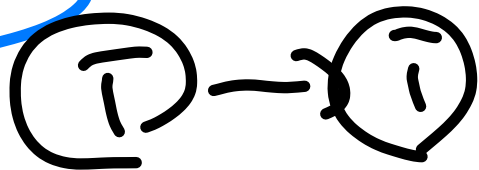
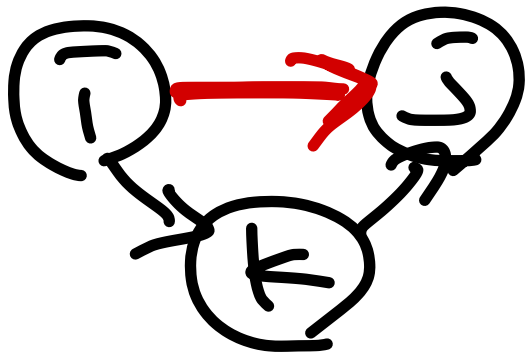
0 ms

953 B

C++14

```
1 #include <cstdio>
2 int d[400][400];
3 int main() {
4     int n,m;
5     scanf("%d %d",&n,&m);
6     while (m--) {
7         int x,y,z;
8         scanf("%d %d %d",&x,&y,&z);
9         x--;y--;
10        if (d[x][y] == 0) {
11            d[x][y] = z;
12        } else if (d[x][y] > z) {
13            d[x][y] = z;
14        }
15    }
16    for (int k=0; k<n; k++) {
17        for (int i=0; i<n; i++) {
18            for (int j=0; j<n; j++) {
19                if (d[i][k] != 0 && d[k][j] != 0) {
20                    if (d[i][j] == 0 || d[i][j] > d[i][k] + d[k][j]) {
21                        d[i][j] = d[i][k]+d[k][j];
22                    }
23                }
24            }
25        }
26    }
27    int ans = -1;
28    for (int i=0; i<n; i++) {
29        if (d[i][i] != 0) {
30            if (ans == -1 || ans > d[i][i]) {
31                ans = d[i][i];
32            }
33        }
34    }
35    printf("%d\n",ans);
36    return 0;
37 }
```

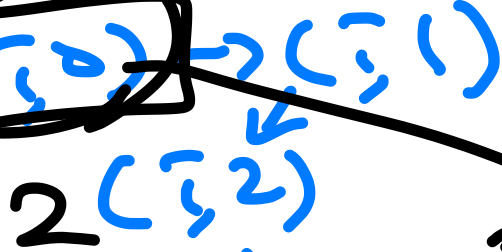
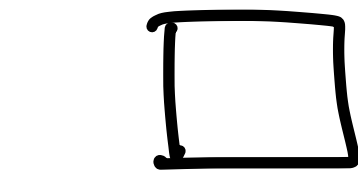
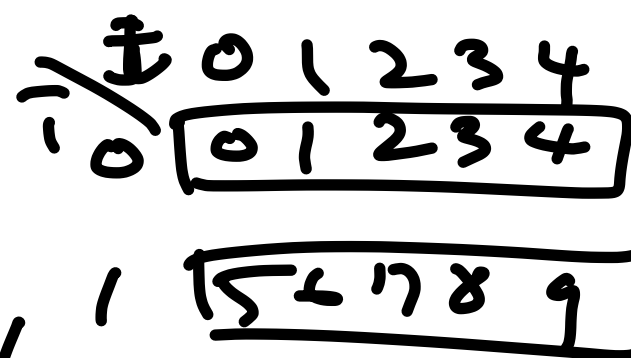
다익스트라 = 0
↑
가늠이 필요



C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <vector>
4 using namespace std;
5 typedef vector<vector<long long>> matrix;
6 const long long mod = 1000003;
7 matrix operator * (const matrix &a, const matrix &b) {
8     int n = a.size();
9     matrix c(n, vector<long long>(n));
10    for (int i=0; i<n; i++) {
11        for (int j=0; j<n; j++) {
12            for (int k=0; k<n; k++) {
13                c[i][j] += a[i][k] * b[k][j];
14            }
15            c[i][j] %= mod;
16        }
17    }
18    return c;
19 }
20 int main() {
21     int n, s, e, t;
22     scanf("%d %d %d %d",&n,&s,&e,&t);
23     matrix ans(n*5, vector<long long>(n*5));
24     for (int i=0; i<n*5; i++) {
25         ans[i][i] = 1;
26     }
27     matrix a(n*5, vector<long long>(n*5));
28     for (int i=0; i<n; i++) {
29         for (int j=0; j<4; j++) {
30             a[5*i+j][5*i+j+1] = 1;
31         }
32     }
33     for (int i=0; i<n; i++) {
34         for (int j=0; j<n; j++) {
35             int v;
36             scanf("%d",&v);
37             if (v > 0) {
38                 a[5*i+v-1][5*j] = 1;
39             }
40         }
41     }
42     while (t > 0) {
43         if (t & 1) {
44             ans = ans * a;
45         }
46         a = a * a;
47         t >>= 1;
48     }
49     s -= 1;
50     e -= 1;
51     printf("%lld\n", ans[5*s][5*e]);
52     return 0;
53 }
```

0 ~ 4



(v, t)

5 * v + t

i -> j

(i, v-1) -> (i, v)

S -> P. (t)

결과

메모리

시간

코드 길이

맞았습니다!!

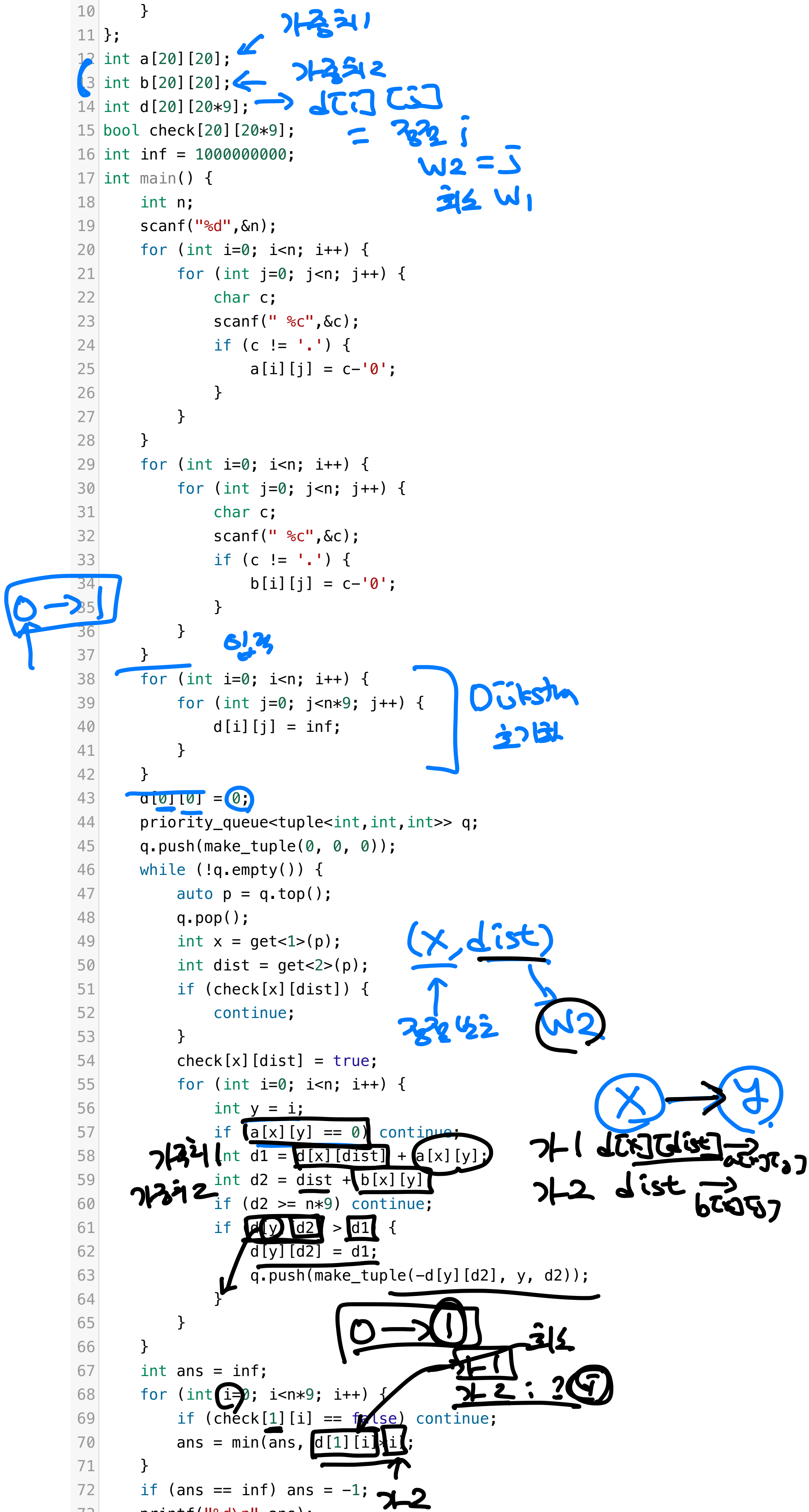
1988 KB

8 ms

1225 B

C++14

```
1 #include <cstdio>
2 #include <vector>
3 #include <queue>
4 #include <tuple>
5 using namespace std;
6 struct Edge {
7     int to;
8     int cost;
9     Edge(int to, int cost) : to(to), cost(cost) {
10     }
11 };
12 int a[20][20];
13 int b[20][20];
14 int d[20][20*9];
15 bool check[20][20*9];
16 int inf = 1000000000;
17 int main() {
18     int n;
19     scanf("%d",&n);
20     for (int i=0; i<n; i++) {
21         for (int j=0; j<n; j++) {
22             char c;
23             scanf(" %c",&c);
24             if (c != '.') {
25                 a[i][j] = c-'0';
26             }
27         }
28     }
29     for (int i=0; i<n; i++) {
30         for (int j=0; j<n; j++) {
31             char c;
32             scanf(" %c",&c);
33             if (c != '.') {
34                 b[i][j] = c-'0';
35             }
36         }
37     }
38     for (int i=0; i<n; i++) {
39         for (int j=0; j<n*9; j++) {
40             d[i][j] = inf;
41         }
42     }
43     d[0][0] = 0;
44     priority_queue<tuple<int,int,int>> q;
45     q.push(make_tuple(0, 0, 0));
46     while (!q.empty()) {
47         auto p = q.top();
48         q.pop();
49         int x = get<1>(p);
50         int dist = get<2>(p);
51         if (check[x][dist]) {
52             continue;
53         }
54         check[x][dist] = true;
55         for (int i=0; i<n; i++) {
56             int y = i;
57             if (a[x][y] == 0) continue;
58             int d1 = d[x][dist] + a[x][y];
59             int d2 = dist + b[x][y];
60             if (d2 >= n*9) continue;
61             if (d[y][d2] > d1) {
62                 d[y][d2] = d1;
63                 q.push(make_tuple(-d[y][d2], y, d2));
64             }
65         }
66     }
67     int ans = inf;
68     for (int i=0; i<n*9; i++) {
69         if (check[1][i] == false) continue;
70         ans = min(ans, d[1][i]-i);
71     }
72     if (ans == inf) ans = -1;
73     printf("%d\n",ans);
74     return 0;
75 }
```



결과	메모리	시간	코드 길이
맞았습니다!!	1200 KB	0 ms	1719 B

C++14

```

1 #include <iostream>
2 #include <string>
3 #include <vector>
4 using namespace std;
5 bool d[111][111];
6 int main() {
7     int n;
8     cin >> n;
9     vector<string> a(n);
10    for (int i=0; i<n; i++) {
11        cin >> a[i];
12    }
13    for (int i=0; i<n; i++) {
14        for (int j=0; j<n; j++) {
15            if (a[i][j] == 'Y' && a[j][i] == 'N') {
16                d[i][j] = true;
17            }
18        }
19    }
20    for (int k=0; k<n; k++) {
21        for (int i=0; i<n; i++) {
22            for (int j=0; j<n; j++) {
23                if (d[i][k] && d[k][j]) {
24                    d[i][j] = true;
25                }
26            }
27        }
28    }
29    for (int i=0; i<n; i++) {
30        if (d[i][i]) {
31            cout << "NO\n";
32            return 0;
33        }
34    }
35    cout << "YES\n";
36    return 0;
37 }

```

결과

메모리

시간

코드 길이

맞았습니다!!

2000 KB

0 ms

776 B

C++14

```

1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 bool d[401][401];
5 int main() {
6     ios_base::sync_with_stdio(false);
7     int n, m;
8     cin >> n >> m;
9     while (m--) {
10         int x, y;
11         cin >> x >> y;
12         d[x][y] = true;
13     }
14     for (int k=1; k<=n; k++) {
15         for (int i=1; i<=n; i++) {
16             for (int j=1; j<=n; j++) {
17                 if (d[i][k] && d[k][j]) d[i][j] = true;
18             }
19         }
20     }
21     cin >> m;
22     while (m--) {
23         int x, y;
24         cin >> x >> y;
25         if (d[x][y] && !d[y][x]) {
26             cout << -1 << '\n';
27         } else if (!d[x][y] && d[y][x]) {
28             cout << 1 << '\n';
29         } else {
30             cout << 0 << '\n';
31         }
32     }
33     return 0;
34 }

```

결과

메모리

시간

코드 길이

맞았습니다!!

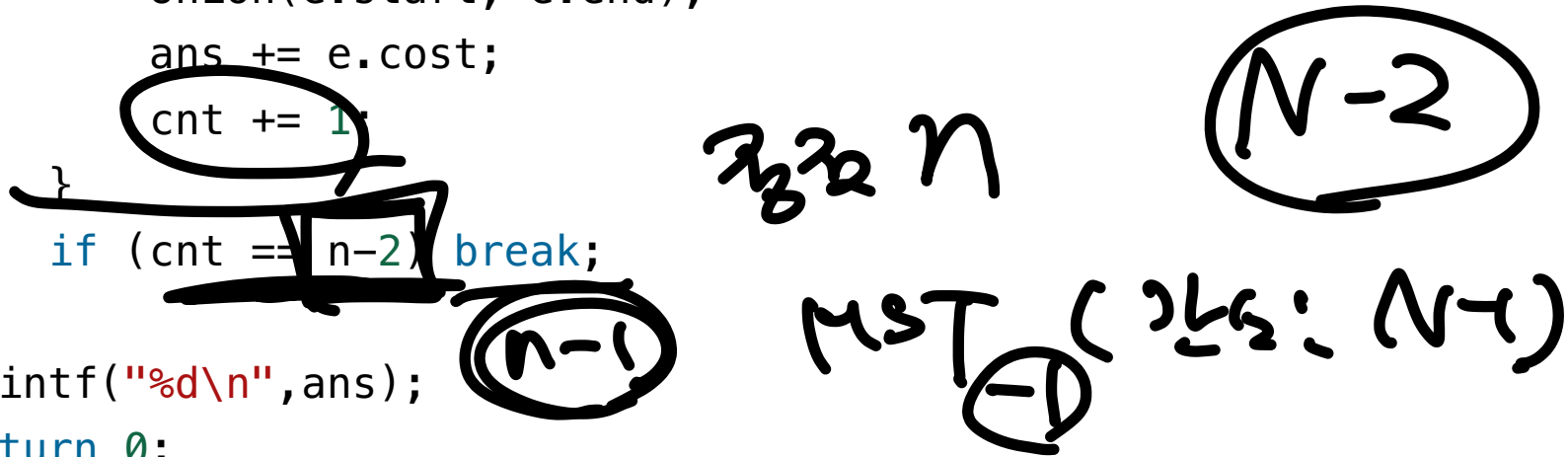
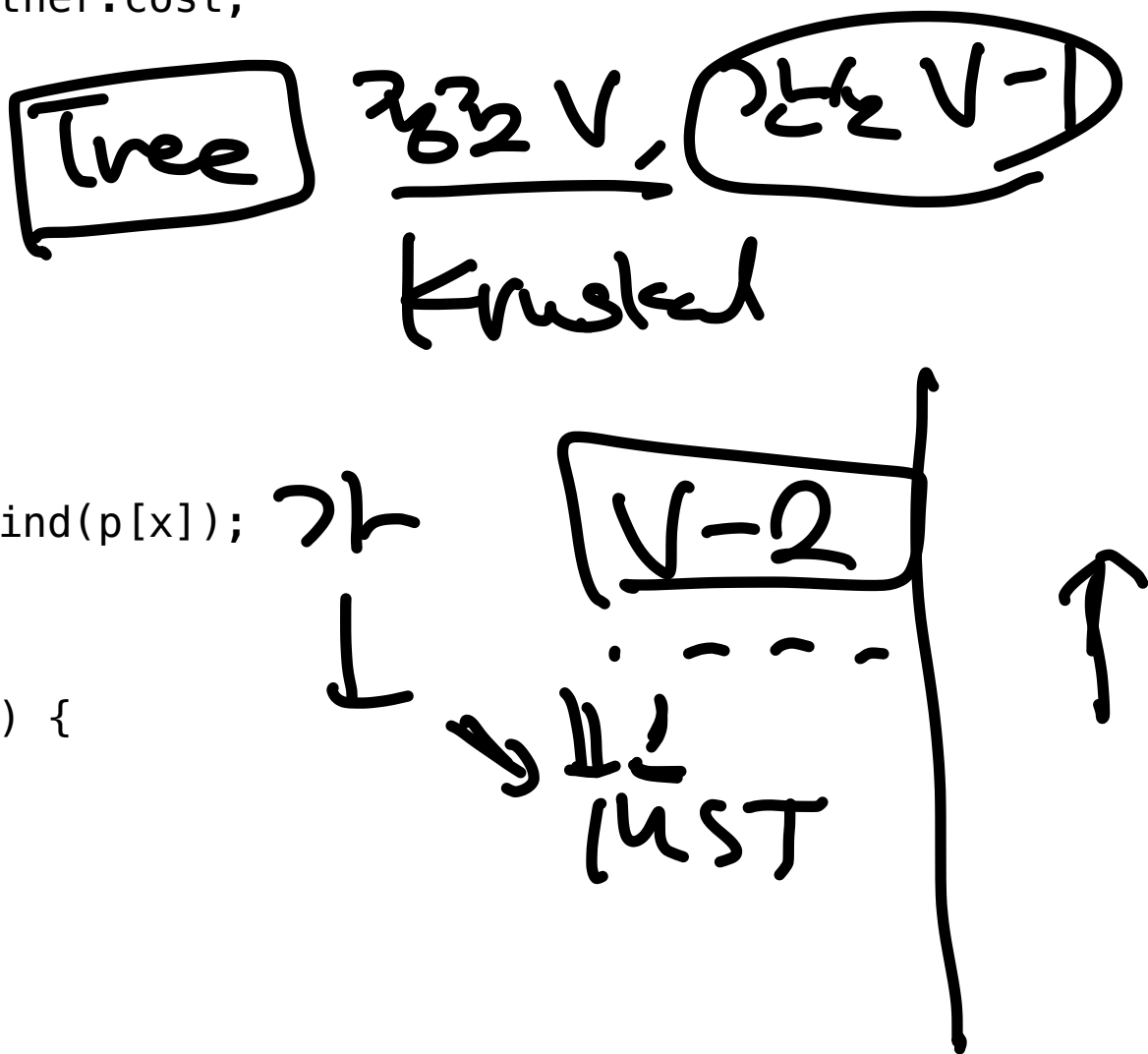
2144 KB

2060 ms

732 B

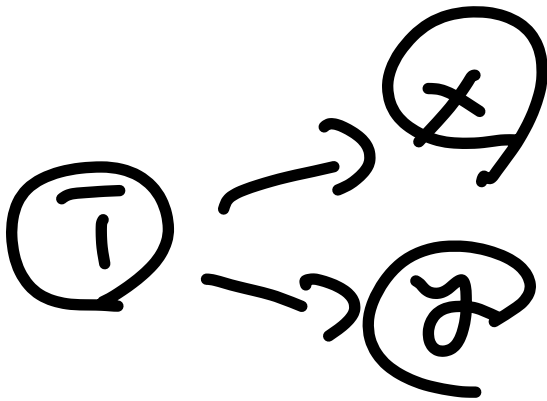
C++14

```
1 #include <cstdio>
2 #include <algorithm>
3 #include <vector>
4 #include <queue>
5 using namespace std;
6 struct Edge {
7     int start, end, cost;
8     bool operator < (const Edge &other) const {
9         return cost < other.cost;
10    }
11 };
12 int p[100001];
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].start, &a[i].end, &a[i].cost);
34     }
35     sort(a.begin(), a.end());
36     int ans = 0;
37     int cnt = 0;
38     for (int i=0; i<m; i++) {
39         Edge e = a[i];
40         int x = Find(e.start);
41         int y = Find(e.end);
42         if (x != y) {
43             Union(e.start, e.end);
44             ans += e.cost;
45             cnt += 1;
46         }
47         if (cnt == n-2) break;
48     }
49     printf("%d\n",ans);
50     return 0;
51 }
```



C++14

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 typedef vector<vector<bool>> matrix;
5 matrix operator * (const matrix &a, const matrix &b) {
6     int n = a.size();
7     matrix c(n, vector<bool>(n));
8     for (int i=0; i<n; i++) {
9         for (int j=0; j<n; j++) {
10             for (int k=0; k<n; k++) {
11                 if (a[i][k] && b[k][j]) {
12                     c[i][j] = true;
13                 }
14             }
15         }
16     }
17     return c;
18 }
19 matrix pow(matrix a, int m) {
20     int n = a.size();
21     matrix ans(n, vector<bool>(n));
22     for (int i=0; i<n; i++) ans[i][i] = true;
23     while (m > 0) {
24         if (m % 2 == 1) {
25             ans = ans * a;
26         }
27         a = a * a;
28         m /= 2;
29     }
30     return ans;
31 }
32 int main() {
33     ios_base::sync_with_stdio(false);
34     cin.tie(NULL);
35     int n, k, m;
36     cin >> n >> k >> m;
37     matrix a(n, vector<bool>(n));
38     for (int i=0; i<n; i++) {
39         int x, y;
40         cin >> x >> y;
41         a[i][x-1] = true;
42         a[i][y-1] = true;
43     }
44     matrix ans = pow(a, k);
45     while (m--) {
46         int x, y;
47         cin >> x >> y;
48         if (ans[x-1][y-1]) {
49             cout << "death" << '\n';
50         } else {
51             cout << "life" << '\n';
52         }
53     }
54     return 0;
55 }
```



a[i][x-1] = true;
a[i][y-1] = true;

pow(a, k);

if (ans[x-1][y-1]) {
 cout << "death" << '\n';
} else {
 cout << "life" << '\n';
}

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.