Chapter. 10

그래프 고급 탐색 알고리즘

핵심 유형 문제풀이

FAST CAMPUS ONLINE

유형별 문제풀이

강사. 나동빈

Chapter. 10

그래프 고급 탐색 알고리즘(핵심 유형 문제풀이)

FAST CAMPUS
ONLINE
나동빈 강사.

Copyright FASTCAMPUS Corp. All Rights Reserved

I 혼자 힘으로 풀어 보기

해킹: https://www.acmicpc.net/problem/10282

문제 제목: 해킹

문제 난이도: 중(Medium)

문제 유형: 다익스트라 최단 경로

추천 풀이 시간: 50분

Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

I 문제 풀이 핵심 아이디어

- 기본적인 다익스트라 최단 경로 알고리즘 문제입니다.
- 도달할 수 있는 정점들의 개수와 최대 거리를 출력합니다.
- 정점의 개수 N이 최대 10,000이고, 간선의 개수 D는 최대 100,000입니다.
- 우선순위 큐를 이용하여, 시간 복잡도는 $O(NlogD)$로 해결할 수 있습니다.

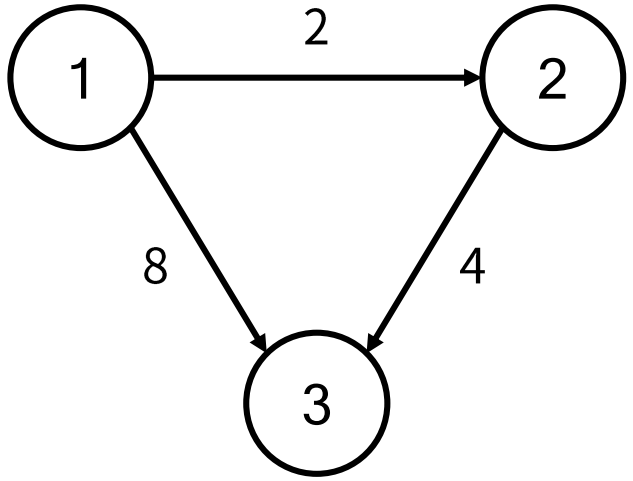
Life Changing Education

Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

문제 풀이 핵심 아이디어

- 정점 개수 = 3, 간선 개수 = 3, 시작 정점 번호 = 1



[최단 거리 테이블]

1	2	3
0	2	6

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

소스코드

```
import heapq
import sys
input = sys.stdin.readline

def dijkstra(start):
    heap_data = []
    heapq.heappush(heap_data, (0, start))
    distance[start] = 0
    while heap_data:
        dist, now = heapq.heappop(heap_data)
        if distance[now] < dist:
            continue
        for i in adj[now]:
            cost = dist + i[1]
            if distance[i[0]] > cost:
                distance[i[0]] = cost
                heapq.heappush(heap_data, (cost, i[0]))
```

```
for _ in range(int(input())):
    n, m, start = map(int, input().split())
    adj = [[] for i in range(n + 1)]
    distance = [1e9] * (n + 1)
    for _ in range(m):
        x, y, cost = map(int, input().split())
        adj[y].append((x, cost))
    dijkstra(start)
    count = 0
    max_distance = 0
    for i in distance:
        if i != 1e9:
            count += 1
            if i > max_distance:
                max_distance = i
    print(count, max_distance)
```


Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

I 혼자 힘으로 풀어 보기

거의 최단 경로: https://www.acmicpc.net/problem/5719

문제 제목: 거의 최단 경로

문제 난이도: 중(Medium)

문제 유형: 다익스트라 최단 경로

추천 풀이 시간: 50분

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

문제 풀이 핵심 아이디어

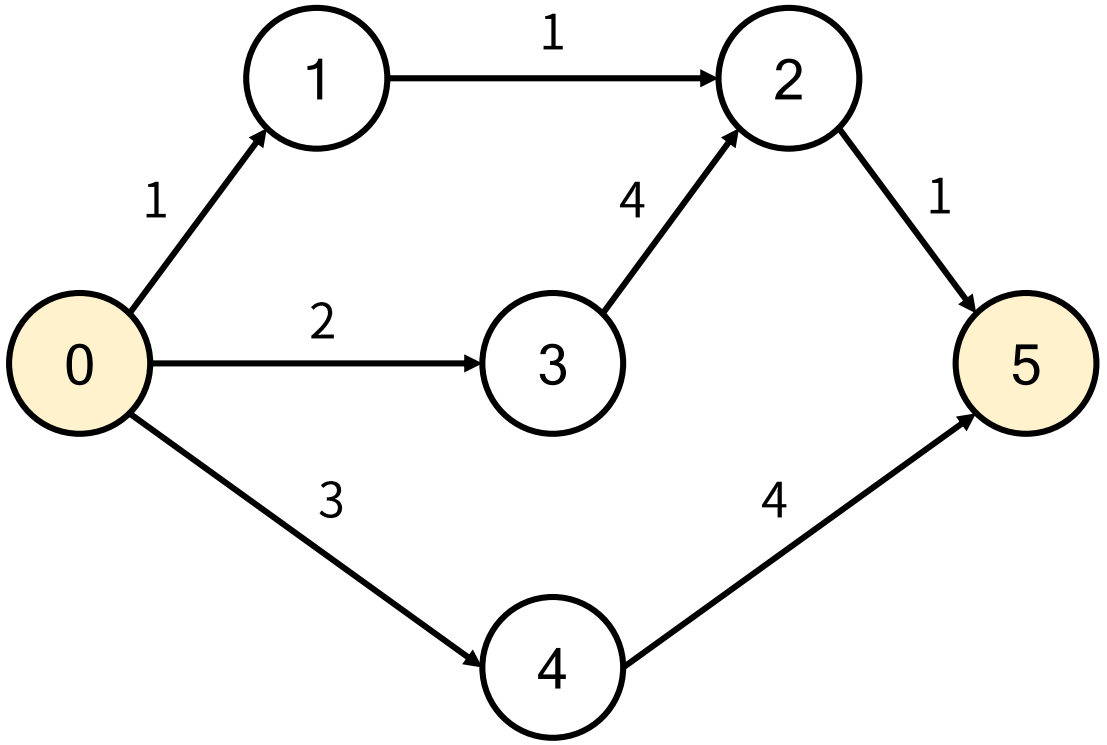
- 다익스트라 최단 경로 알고리즘을 수행합니다.
- 다익스트라 최단 경로에 포함되는 **모든 간선을 추적**해야 합니다.
- 초기 최단 경로에 포함된 간선을 제외한 뒤에, 다시 최단 경로를 탐색하면 됩니다.

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

문제 풀이 핵심 아이디어

- 최단 경로를 구성하는 간선들을 찾는 방법은 무엇일까요?

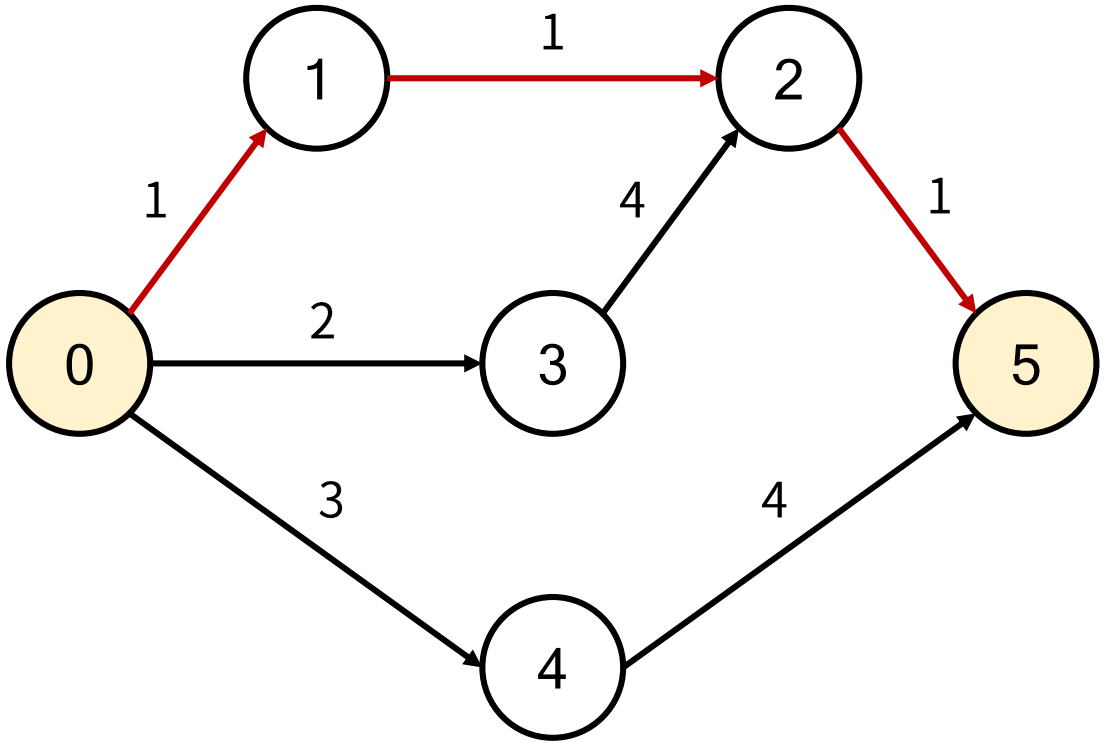


Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

문제 풀이 핵심 아이디어

- 일단 모든 정점으로의 최단 거리를 찾습니다.



[최단 거리 테이블]

0	1	2	3	4	5
0	1	2	2	3	3

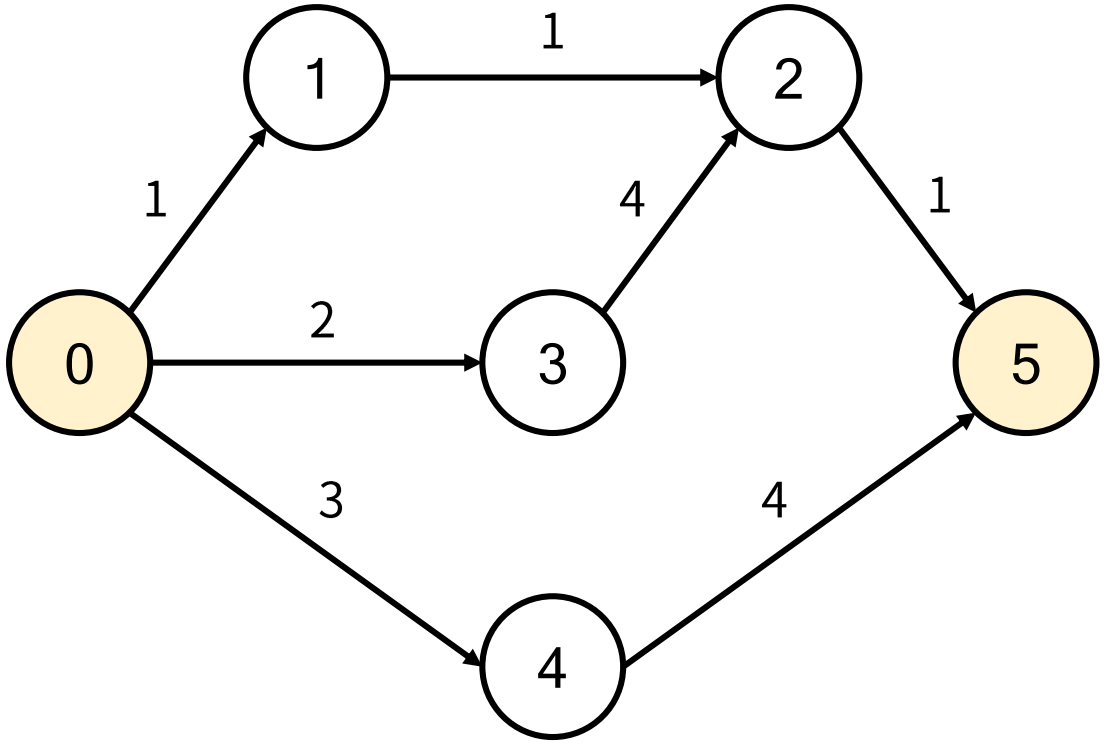
Life Changing Education

Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

I 문제 풀이 핵심 아이디어

- BFS를 이용하여 최단 경로에 포함되어 있는 모든 간선을 역으로 추적 할 수 있습니다.
- 최단 거리 = 3



[최단 거리 테이블]

0	1	2	3	4	5
0	1	2	2	3	3

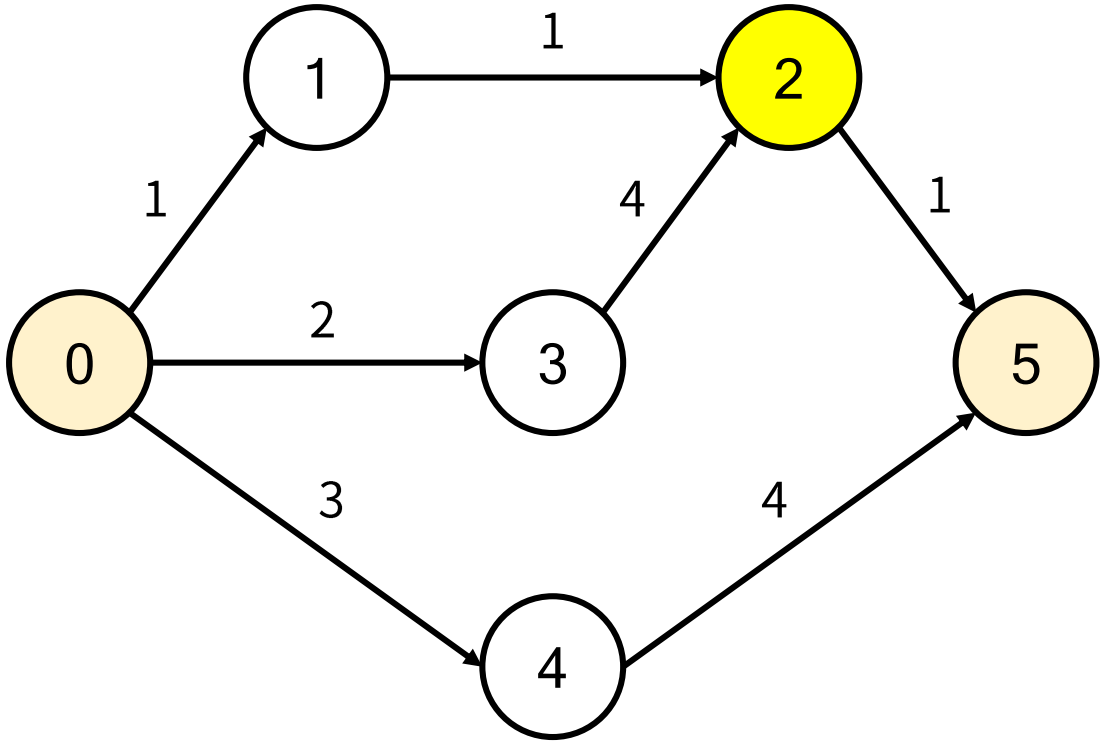
5

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

문제 풀이 핵심 아이디어

- BFS를 이용하여 최단 경로에 포함되어 있는 모든 간선을 역으로 추적 할 수 있습니다.
- 최단 거리 = 3



[최단 거리 테이블]

0	1	2	3	4	5
0	1	2	2	3	3

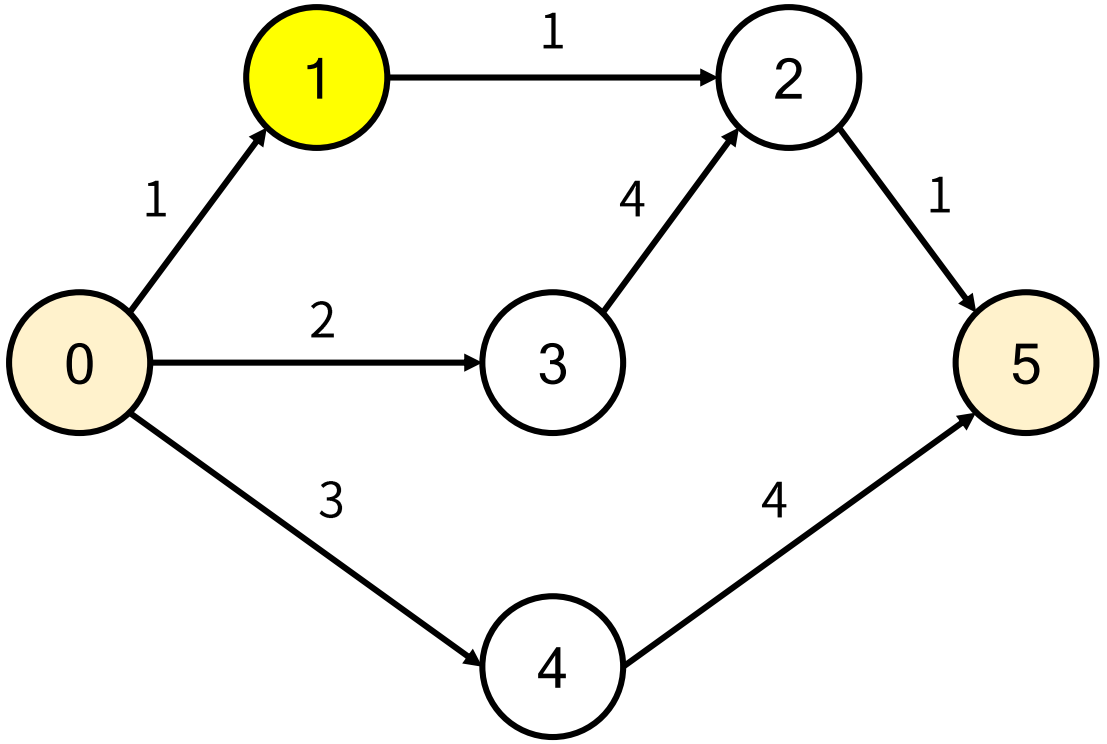
2

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

문제 풀이 핵심 아이디어

- BFS를 이용하여 최단 경로에 포함되어 있는 모든 간선을 역으로 추적 할 수 있습니다.
- 최단 거리 = 3



[최단 거리 테이블]

0	1	2	3	4	5
0	1	2	2	3	3

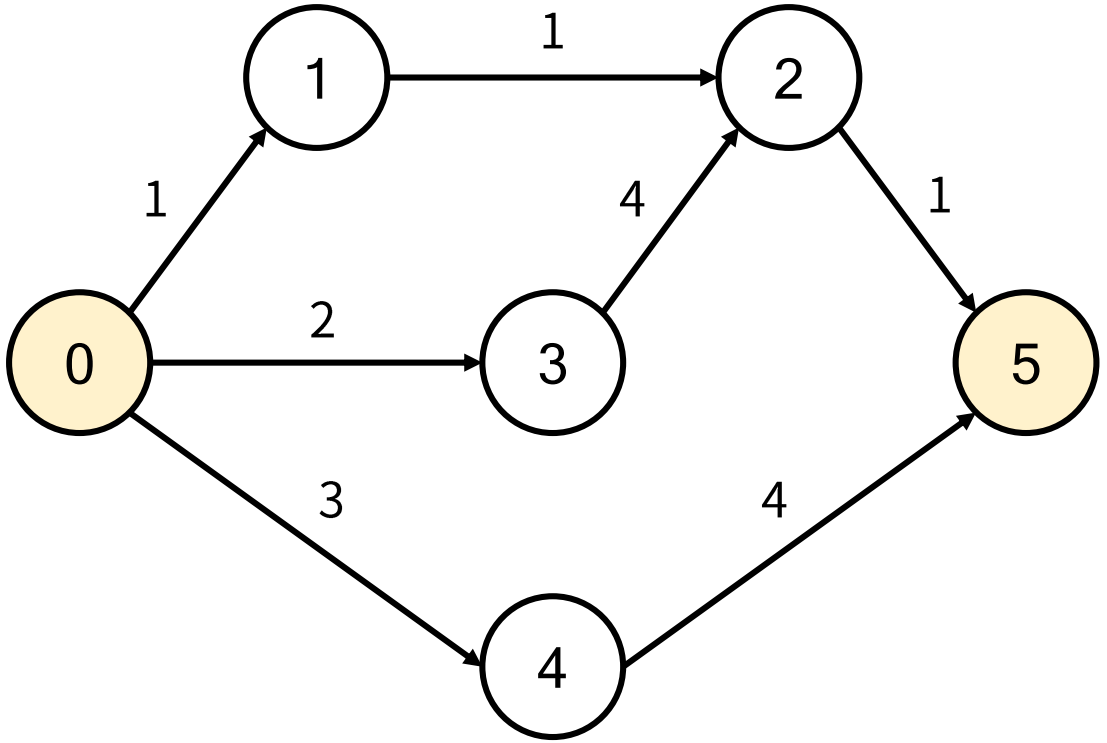
1

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

해킹: https://www.acmicpc.net/problem/10282

I 문제 풀이 핵심 아이디어

- BFS를 이용하여 최단 경로에 포함되어 있는 모든 간선을 역으로 추적 할 수 있습니다.
- 최단 거리 = 3



[최단 거리 테이블]

0	1	2	3	4	5
0	1	2	2	3	3

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

거의 최단 경로: https://www.acmicpc.net/problem/5719

소스코드

```
from collections import deque
import heapq
import sys
input = sys.stdin.readline

def dijkstra():
    heap_data = []
    heapq.heappush(heap_data, (0, start))
    distance[start] = 0
    while heap_data:
        dist, now = heapq.heappop(heap_data)
        if distance[now] < dist:
            continue
        for i in adj[now]:
            cost = dist + i[1]
            if distance[i[0]] > cost and not dropped[now][i[0]]:
                distance[i[0]] = cost
                heapq.heappush(heap_data, (cost, i[0]))

def bfs():
    q = deque()
    q.append(end)
    while q:
        now = q.popleft()
        if now == start:
            continue
        for prev, cost in reverse_adj[now]:
            if distance[now] == distance[prev] + cost:
                dropped[prev][now] = True
                q.append(prev)
```

```
while True:
    n, m = map(int, input().split())
    if n == 0:
        break
    start, end = map(int, input().split())
    adj = [[] for _ in range(n + 1)]
    reverse_adj = [[] for _ in range(n + 1)]
    for _ in range(m):
        x, y, cost = map(int, input().split())
        adj[x].append((y, cost))
        reverse_adj[y].append((x, cost))
    dropped = [[False] * (n + 1) for _ in range(n + 1)]
    distance= [1e9] * (n + 1)
    dijkstra()
    bfs()
    distance= [1e9] * (n + 1)
    dijkstra()
    if distance[end] != 1e9:
        print(distance[end])
    else:
        print(-1)
```


Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

우주신과의 교감: https://www.acmicpc.net/problem/1774

I 혼자 힘으로 풀어 보기

문제 제목: 우주신과의 교감

문제 난이도: 중(Medium)

문제 유형: 최소 신장 트리

추천 풀이 시간: 40분

Life Changing Education

Fast campus

Copyright FASTCAMPUS Corp. All Rights Reserved

우주신과의 교감: https://www.acmicpc.net/problem/1774

I 혼자 힘으로 풀어 보기

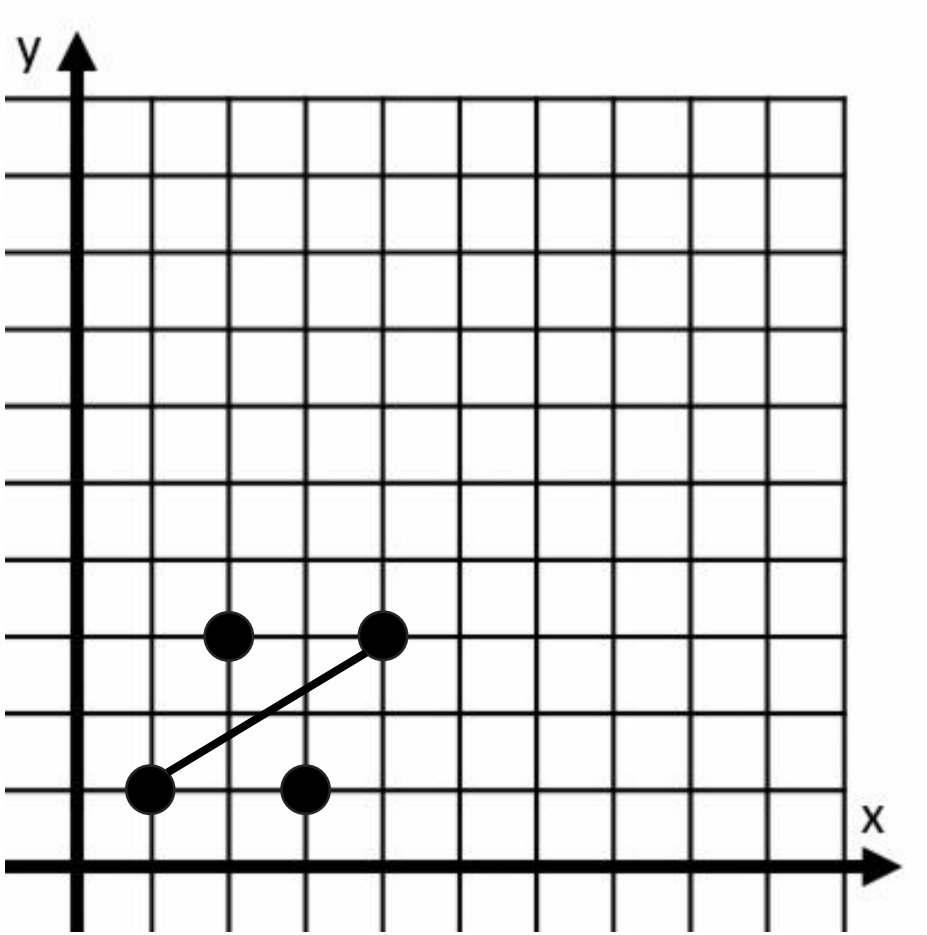
- 2차원 좌표가 주어졌을 때, 모든 좌표를 잇는 최소 신장 트리를 만들어야 합니다.
- 따라서 2차원 좌표 상의 점을 잇는 통로들을 고려해야 합니다.
- 정점의 개수 N이 최대 1,000이므로, 가능한 통로의 개수는 약 N^2입니다.
- 크루스칼 알고리즘은 간선의 개수가 E일 때 O(ElogE)로 동작합니다.
- 따라서 이 문제는 **크루스칼 알고리즘으로 해결**할 수 있습니다.

Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

우주신과의 교감: https://www.acmicpc.net/problem/1774

혼자 힘으로 풀어 보기

- 예를 들어 다음과 같은 예시를 확인해 봅시다.

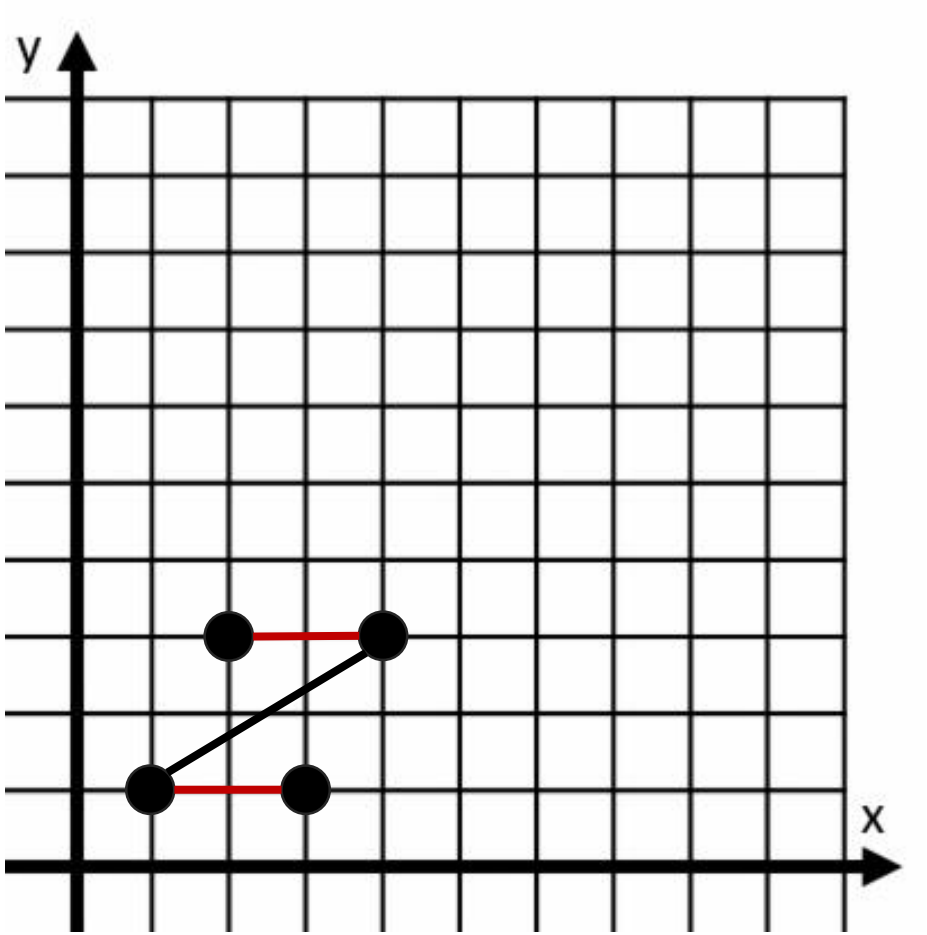


Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

우주신과의 교감: https://www.acmicpc.net/problem/1774

I 혼자 힘으로 풀어 보기

- 최소 신장 트리는 다음과 같습니다. (만들어야 할 최소의 통로 길이: 4.00)



Life Changing Education
Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

우주신과의 교감: https://www.acmicpc.net/problem/1774

소스코드

```python
import math
import sys
input = sys.stdin.readline

def get_distance(p1, p2):
    a = p1[0] - p2[0]
    b = p1[1] - p2[1]
    return math.sqrt((a * a) + (b * b))

def get_parent(parent, n):
    if parent[n] == n:
        return n
    return get_parent(parent, parent[n])

def union_parent(parent, a, b):
    a = get_parent(parent, a)
    b = get_parent(parent, b)
    if a < b:
        parent[b] = a
    else:
        parent[a] = b

def find_parent(parent, a, b):
    a = get_parent(parent, a)
    b = get_parent(parent, b)
    if a == b:
        return True
    else:
        return False
```

```python
edges = []
parent = {}
locations = []
n, m = map(int, input().split())

for _ in range(n):
    x, y = map(int, input().split())
    locations.append((x, y))

length = len(locations)

for i in range(length - 1):
    for j in range(i + 1, length):
        edges.append((i + 1, j + 1, get_distance(locations[i], locations[j])))

for i in range(1, n + 1):
    parent[i] = i

for i in range(m):
    a, b = map(int, input().split())
    union_parent(parent, a, b)

edges.sort(key=lambda data: data[2])

result = 0
for a, b, cost in edges:
    if not find_parent(parent, a, b):
        union_parent(parent, a, b)
        result += cost

print("%0.2f" % result)
```


Copyright FASTCAMPUS Corp. All Rights Reserved