ICPC Sinchon

2022 Winter Algorithm Camp

11. 분리 집합 & 최소 신장 트리

서강대학교 임지환

2022 Winter Algorithm Camp

목차

2022 Winter Algorithm Camp

Introduction

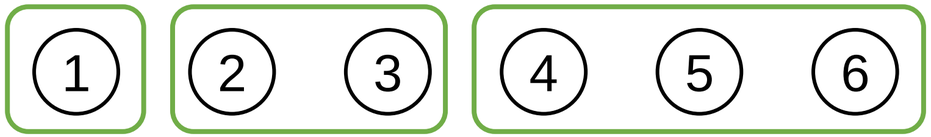
* 마지막 강의를 시작하며...

2022 Winter Algorithm Camp

Disjoint Set – Introduction

* 분리 집합

- 서로 겹치는 원소가 없는 집합들을 관리하는 자료구조
- 서로소 집합, 유니온-파인드(Union-Find)
- 포레스트의 일종



Disjoint Set – Operations

* Find

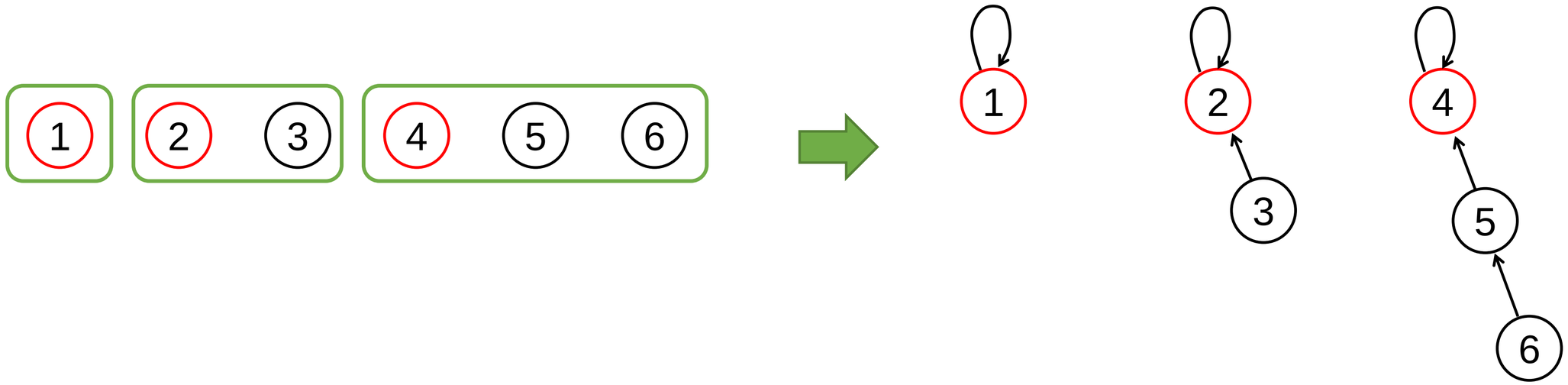
- 주어진 원소가 포함된 집합의 대푯값을 구하는 연산
- 집합마다 유일한 대푯값 존재
- 나머지 다른 원소에서 대표로 가는 경로 존재

* Union

- 두 집합을 합치는 연산
- 한 집합의 대푯값을 다른 집합의 대푯값으로 linking

Disjoint Set – Expression

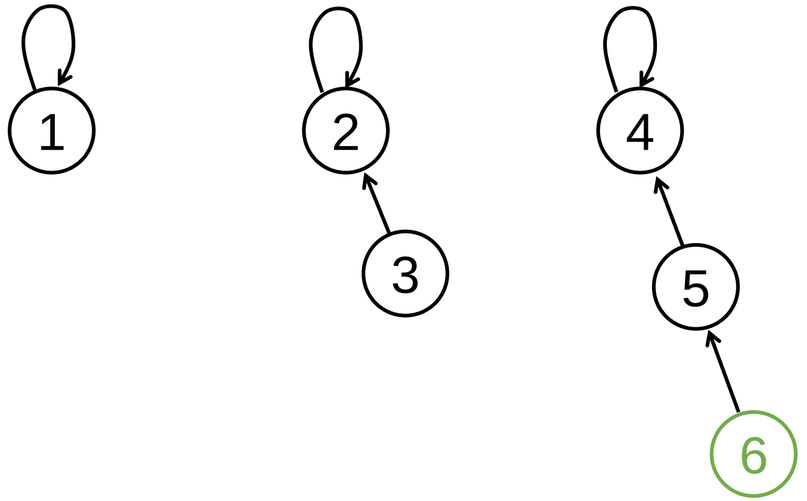
Disjoint Set Tree Model



Disjoint Set – Expression

Disjoint Set Tree Model – Find

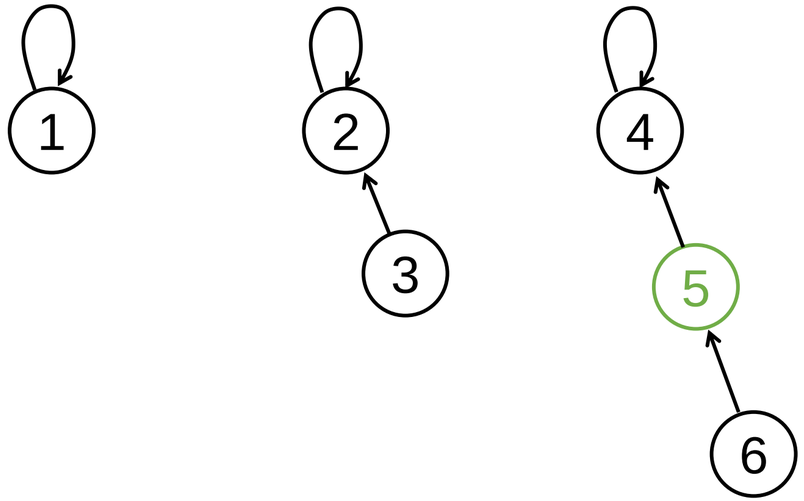
find(6):



Disjoint Set – Expression

Disjoint Set Tree Model – Find

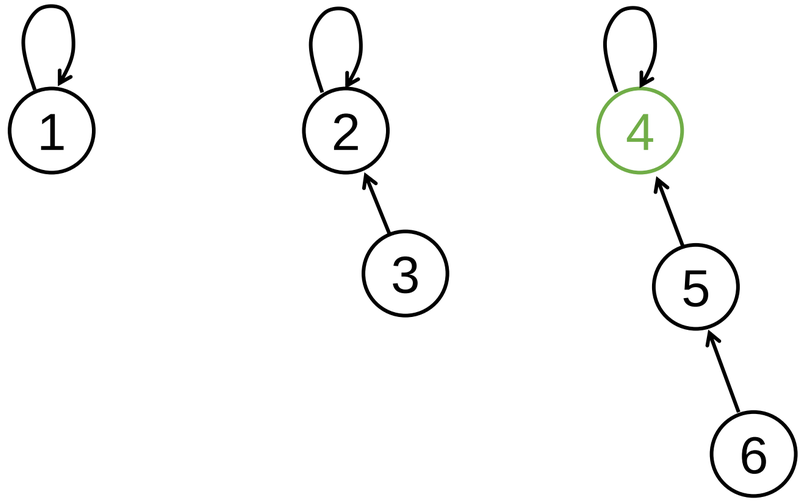
find(6):



Disjoint Set – Expression

Disjoint Set Tree Model – Find

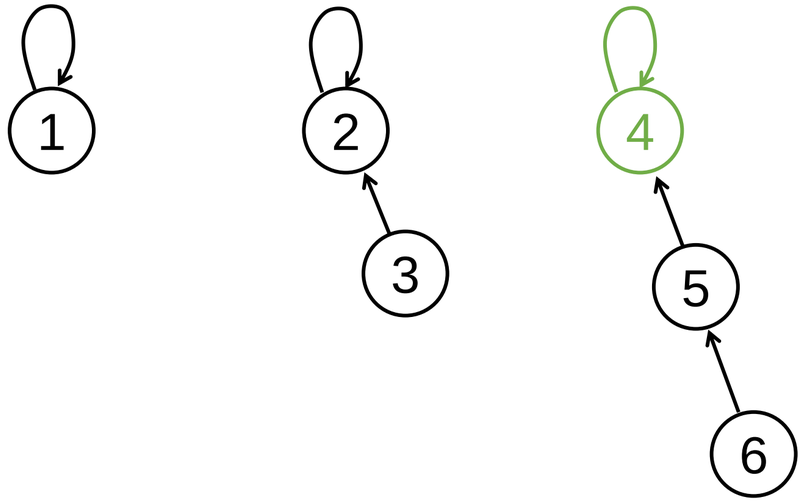
find(6):



Disjoint Set – Expression

Disjoint Set Tree Model – Find

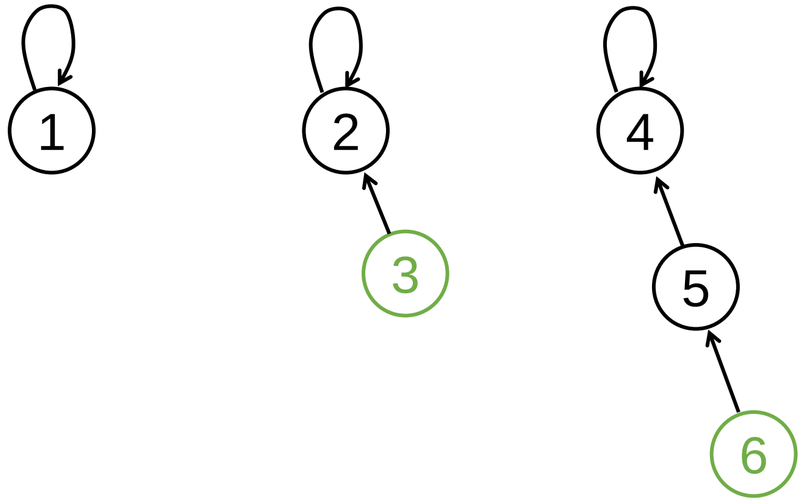
find(6):



Disjoint Set – Expression

Disjoint Set Tree Model – Union

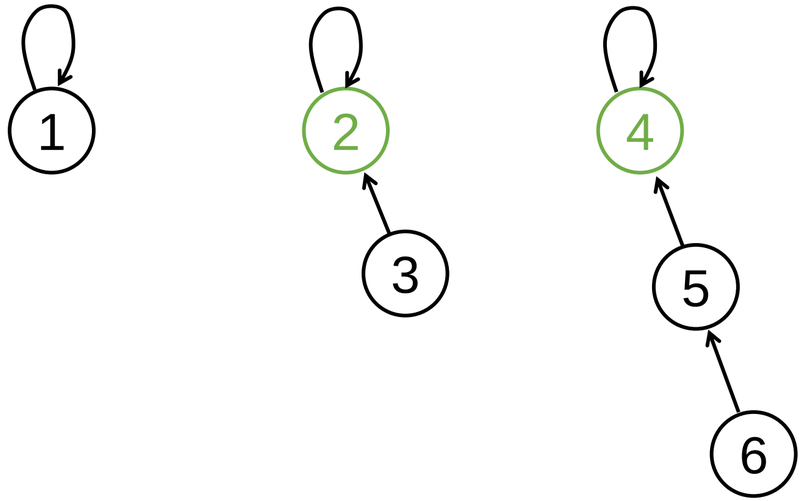
union(3, 6):



Disjoint Set – Expression

Disjoint Set Tree Model – Union

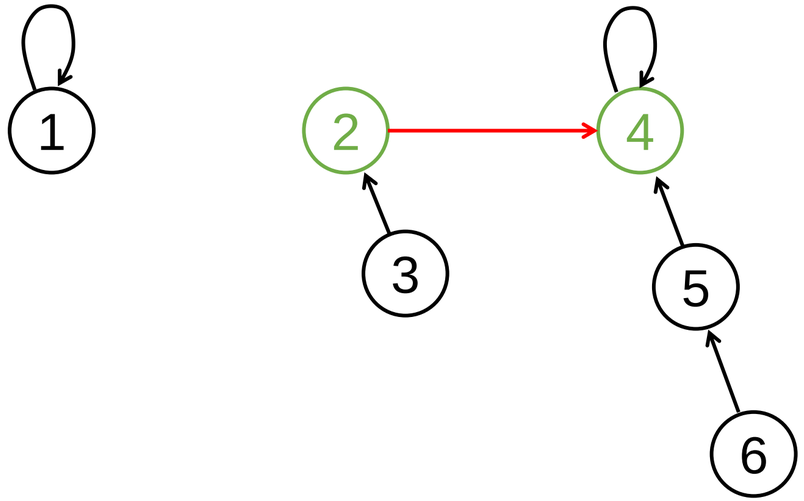
union(3, 6):



Disjoint Set – Expression

Disjoint Set Tree Model – Union

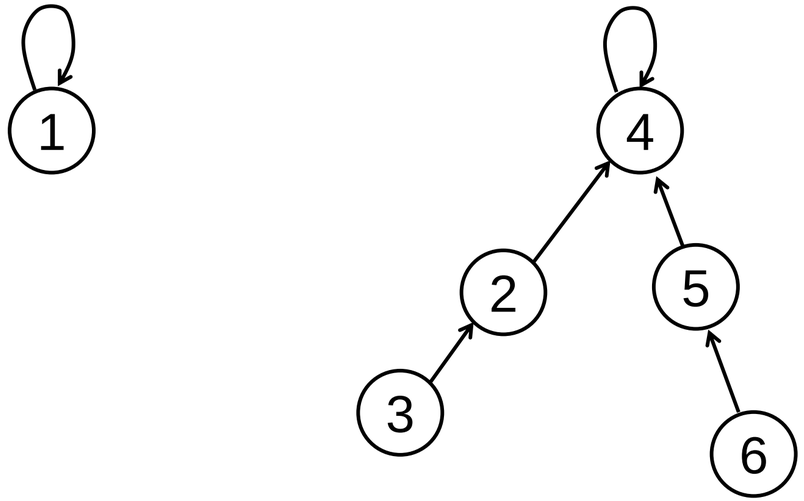
union(3, 6):



Disjoint Set – Expression

Disjoint Set Tree Model – Union

union(3, 6):

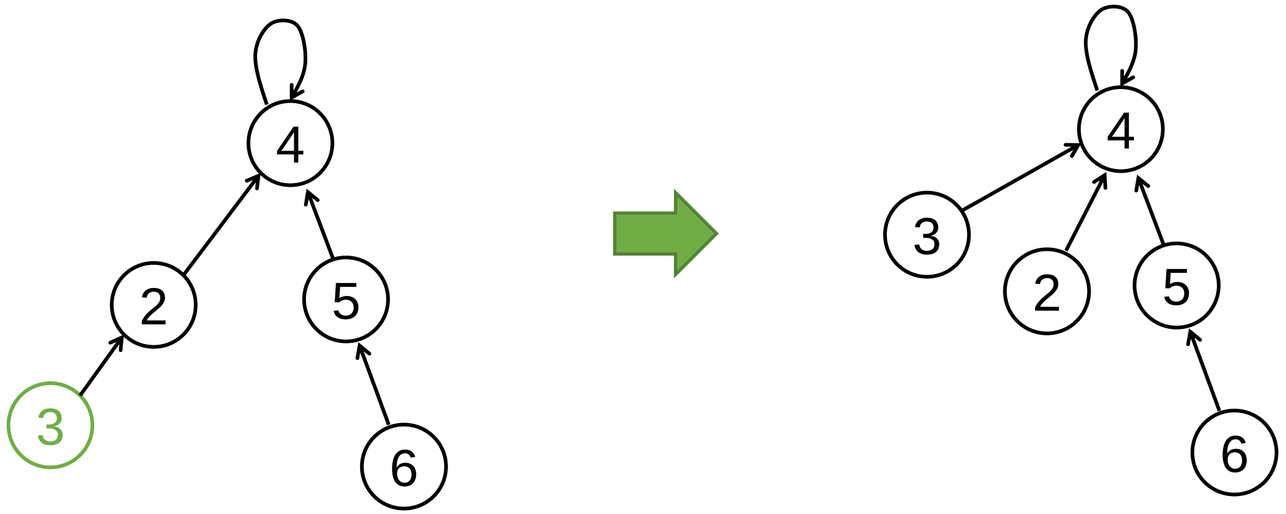


2022 Winter Algorithm Camp

Disjoint Set – Optimization

* 경로 압축(Path Compression)

- Find연산 시 대푯값을 찾는 과정에서 만나는 경로 상의 모든 원소가 대푯값을 바로 가리키게 함
- 최악 시간복잡도 $O(N) \rightarrow O(1)$로 만들어줌



2022 Winter Algorithm Camp

4803. 트리

N개의 정점과 M개의 간선 $\left(1 \leq N \leq 500, 0 \leq M \leq \frac{N(N-1)}{2}\right)$

입력으로 주어진 그래프가 트리인가, 트리가 없는가, 몇 개의 트리로 구성된 포레스트인가를 구해보자.

2022 Winter Algorithm Camp

4803. 트리

주어진 간선을 하나씩 추가하여 컴포넌트끼리 연결시켜보자.

최초(추가된 간선이 없는 경우): N개의 트리로 구성된 포레스트

주어진 간선의 양 끝 정점에 대하여

1) 이미 같은 집합이라면: 해당 간선에 의해 사이클 발생 -> Not a tree, graph
2) 서로 다른 집합이라면: 포레스트에서 트리의 개수가 하나 감소

2022 Winter Algorithm Camp

13904. 과제

N개의 과제와 각각의 마감일 d, 점수 w $(1 \leq N, d \leq 1000, 1 \leq w \leq 100)$

하루에 한 과제를 끝낼 수 있고, 마감일이 지난 과제는 점수를 받을 수 없다.

마감일에 맞춰 적절히 과제를 해결할 때 얻을 수 있는 점수의 최댓값을 구해보자.

2022 Winter Algorithm Camp

13904. 과제

1. 과제를 마감일에 매칭시킨다고 생각해보자.

ex: (4, 60), (4, 40), (1, 20), (2, 50), (3, 30), (4, 10), (6, 5)

Day slot: 6

Day	1	2	3	4	5	6
Occupied						

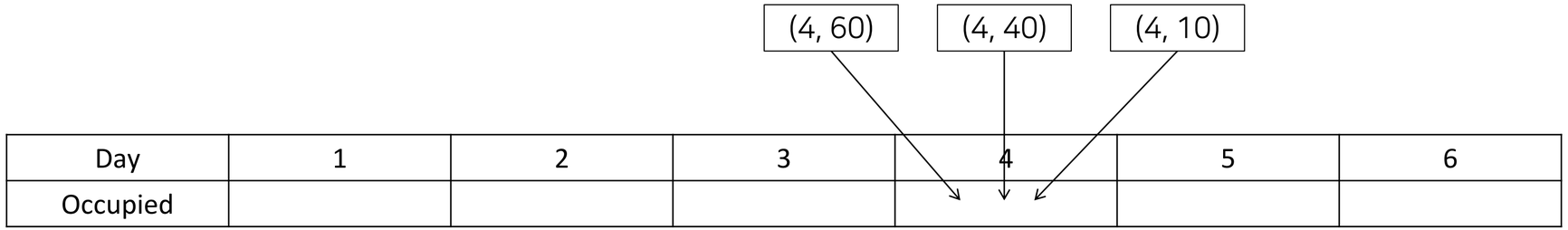
2022 Winter Algorithm Camp

13904. 과제

2. 성질 찾아보기

ex: (4, 60), (4, 40), (1, 20), (2, 50), (3, 30), (4, 10), (6, 5)

마감일이 4일인 과제들 중 굳이 4일에 넣는다면 어떤 것을 넣어야 할까?



2022 Winter Algorithm Camp

13904. 과제

2. 성질 찾아보기

ex: (4, 60), (4, 40), (1, 20), (2, 50), (3, 30), (4, 10), (6, 5)

마감일이 4일인 과제들 중 굳이 4일에 넣는다면 어떤 것을 넣어야 할까?

Day	1	2	3	4	5	6
Occupied				(4, 60)		

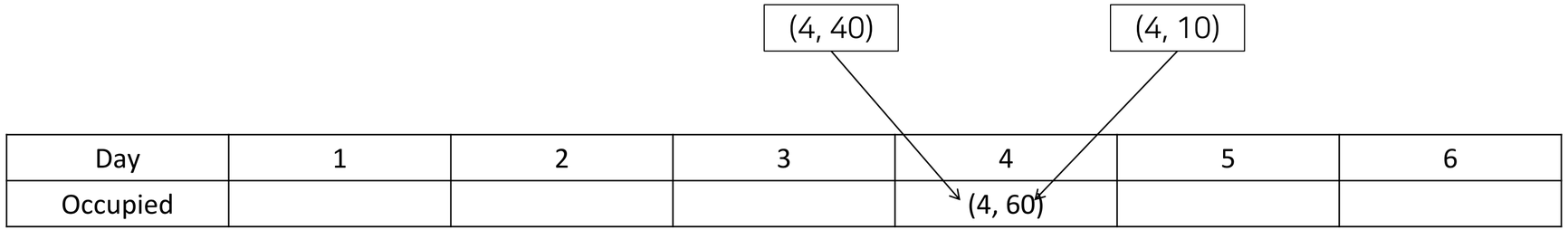
2022 Winter Algorithm Camp

13904. 과제

2. 성질 찾아보기

ex: (4, 60), (4, 40), (1, 20), (2, 50), (3, 30), (4, 10), (6, 5)

특정 일자에 끝낼 과제가 정해졌다면, 마감 기한이 같은 다른 과제는 어떻게 해야 할까?



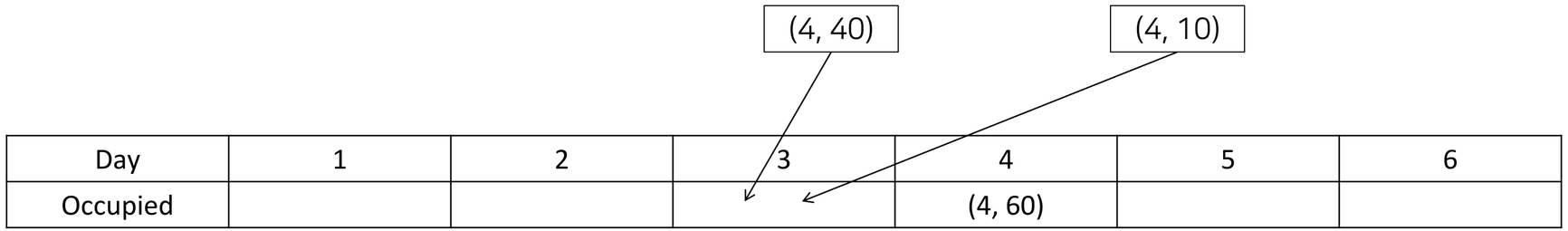
2022 Winter Algorithm Camp

13904. 과제

2. 성질 찾아보기

ex: (4, 60), (4, 40), (1, 20), (2, 50), (3, 30), (4, 10), (6, 5)

특정 일자에 끝낼 과제가 정해졌다면, 마감 기한이 같은 다른 과제는 어떻게 해야 할까?



2022 Winter Algorithm Camp

13904. 과제

3. 구현

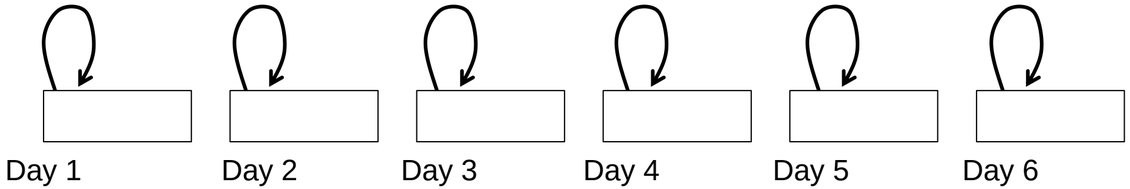
1) 점수가 높은 과제는 우선적으로 처리해야 한다. -> 단순 정렬
2) 마감 기한이 같은 다른 과제는 그 전에 끝내야 한다. -> 서로소 집합으로 해결

2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: **(4, 60)**, (2, 50), (4, 40), (3, 30), (1, 20), (4, 10), (6, 5)

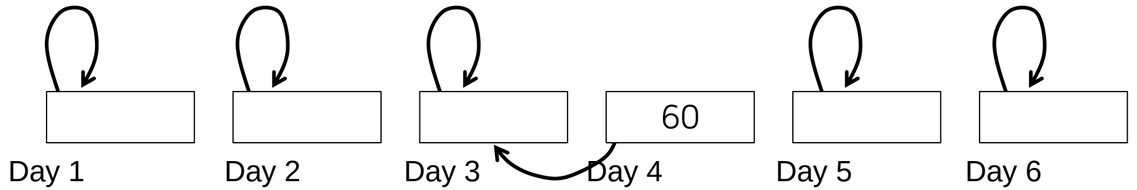


2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: (4, 60), (2, 50), (4, 40), (3, 30), (1, 20), (4, 10), (6, 5)

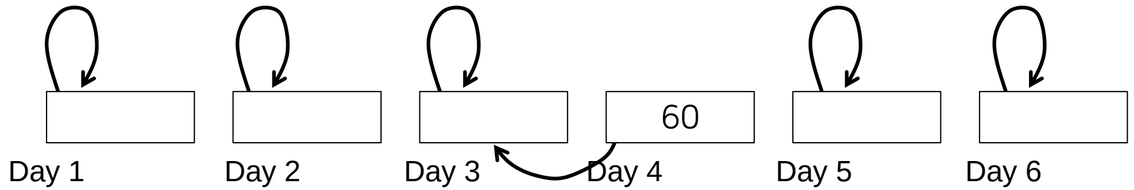


2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: (4, 60), **(2, 50)**, (4, 40), (3, 30), (1, 20), (4, 10), (6, 5)

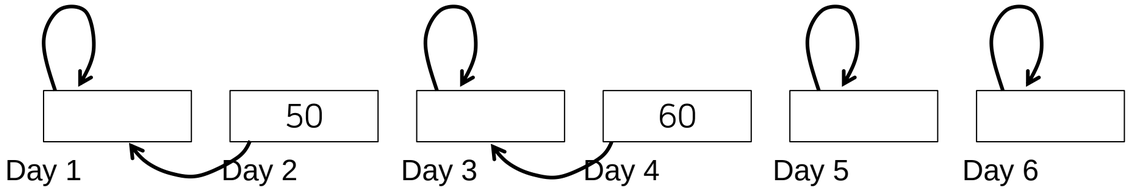


2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: (4, 60), (2, 50), **(4, 40)**, (3, 30), (1, 20), (4, 10), (6, 5)

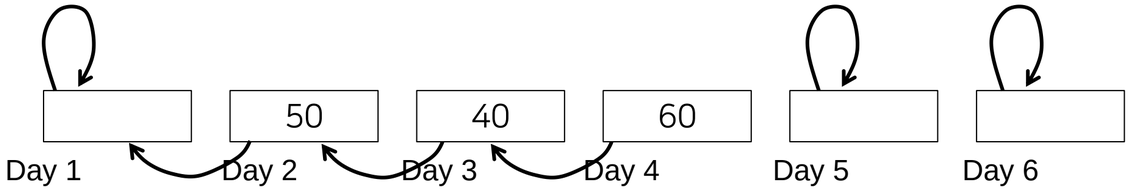


2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: (4, 60), (2, 50), (4, 40), **(3, 30)**, (1, 20), (4, 10), (6, 5)

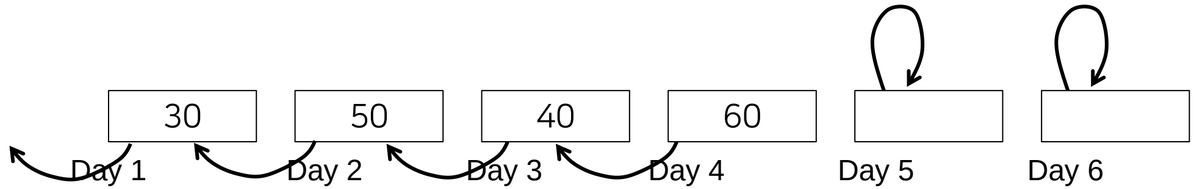


2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

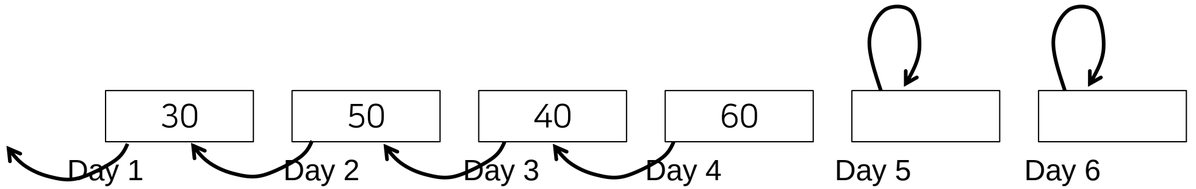
ex: (4, 60), (2, 50), (4, 40), (3, 30), **(1, 20)**, (4, 10), (6, 5)



13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: (4, 60), (2, 50), (4, 40), (3, 30), (1, 20), (4, 10), **(6, 5)**

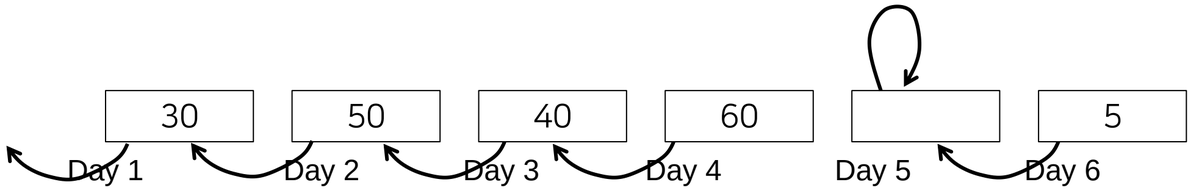


2022 Winter Algorithm Camp

13904. 과제

4. 과제의 마감기한 d가 주어졌을 때, d일 이전의 Day slot중 비어 있는 날 찾기

ex: (4, 60), (2, 50), (4, 40), (3, 30), (1, 20), (4, 10), (6, 5)

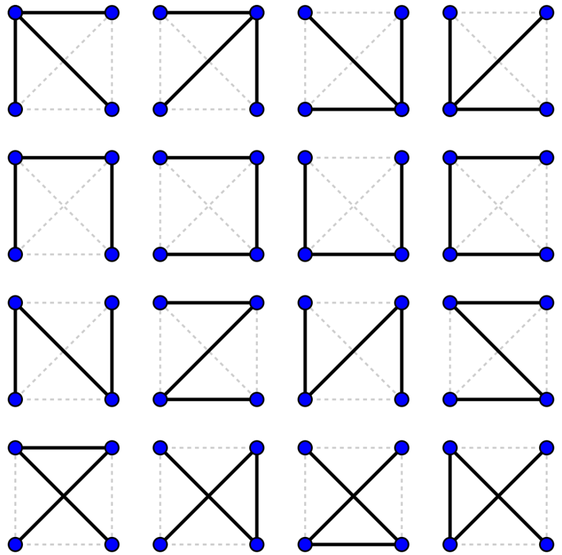


2022 Winter Algorithm Camp

Minimum Spanning Tree

* Spanning Tree(신장 트리)

- 어떤 그래프의 모든 노드와 간선 일부를 포함하며 모든 노드 간에 경로가 존재하도록 만들어진 부분 그래프 중 트리인 것
- 유일하지 않음
- 신장트리 가중치: 간선 가중치의 합

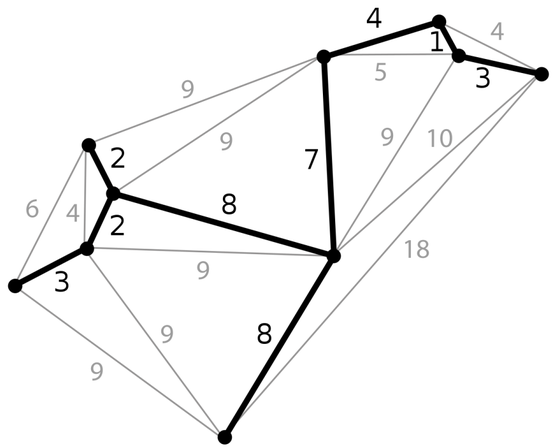


2022 Winter Algorithm Camp

Minimum Spanning Tree

* Minimum Spanning Tree(최소 신장 트리)

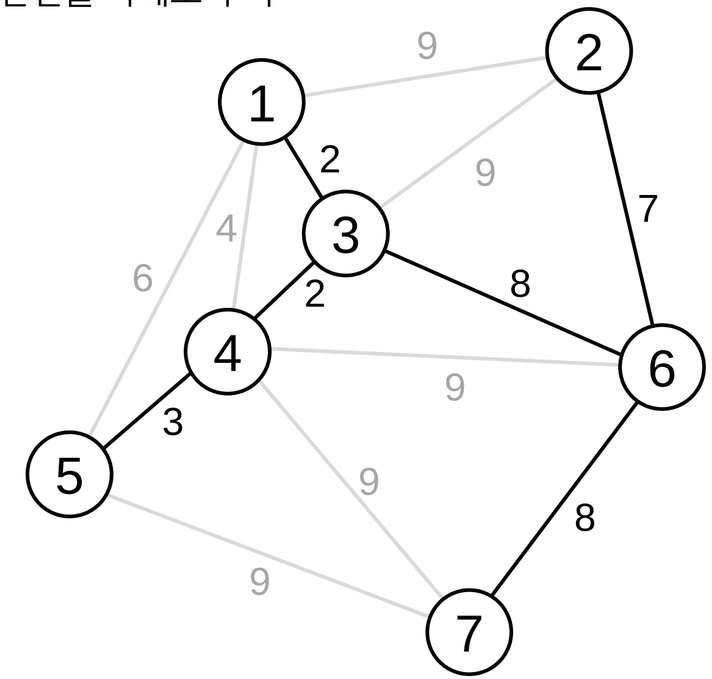
- 신장트리 중 가중치가 가장 작은 것
- 신장트리와 마찬가지로, 최소 신장 트리도 유일하지 않을 수 있음



Minimum Spanning Tree – Algorithms

* Kruskal's Algorithm

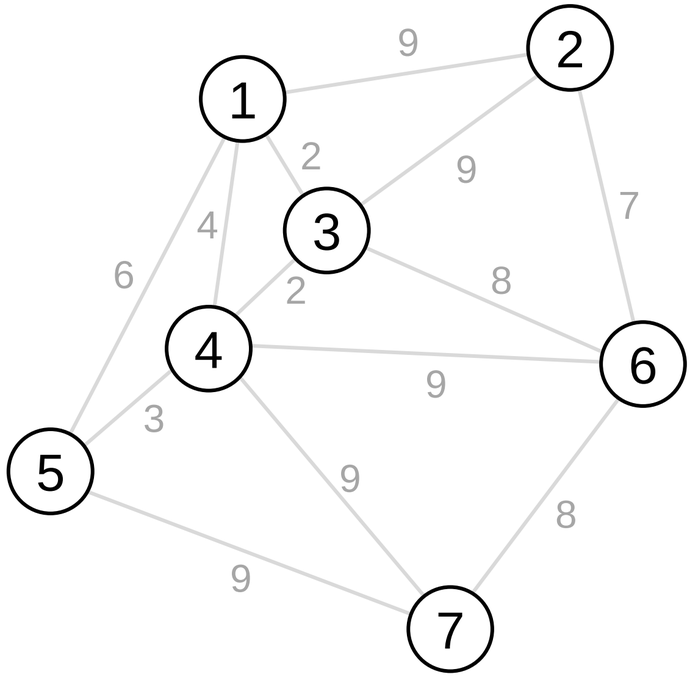
- 탐욕법을 기반으로 간선을 추가해 가면서 mst를 구성하는 방법
- Tree임을 만족하면서 가중치가 작은 간선을 차례로 추가



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

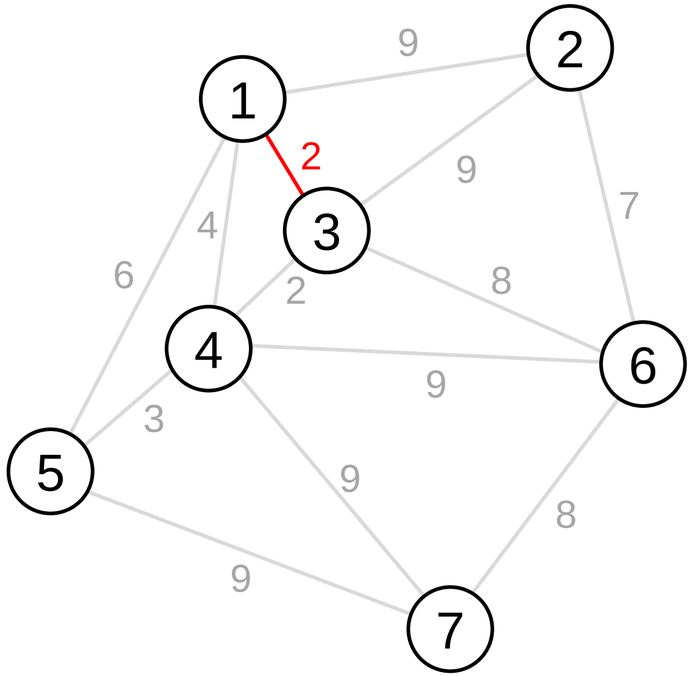
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

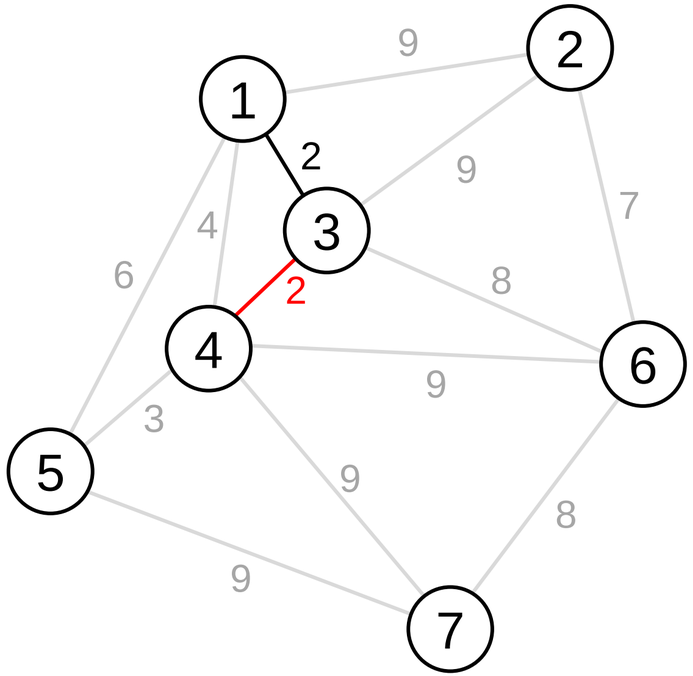
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

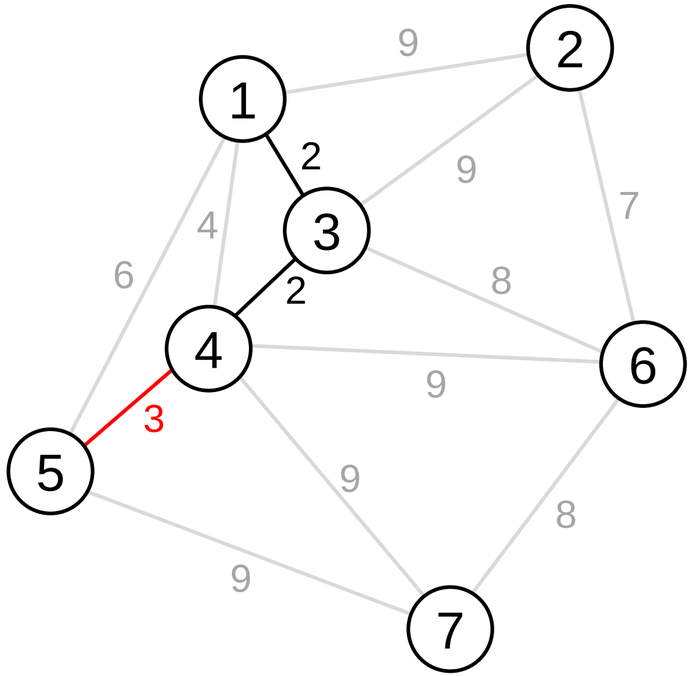
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

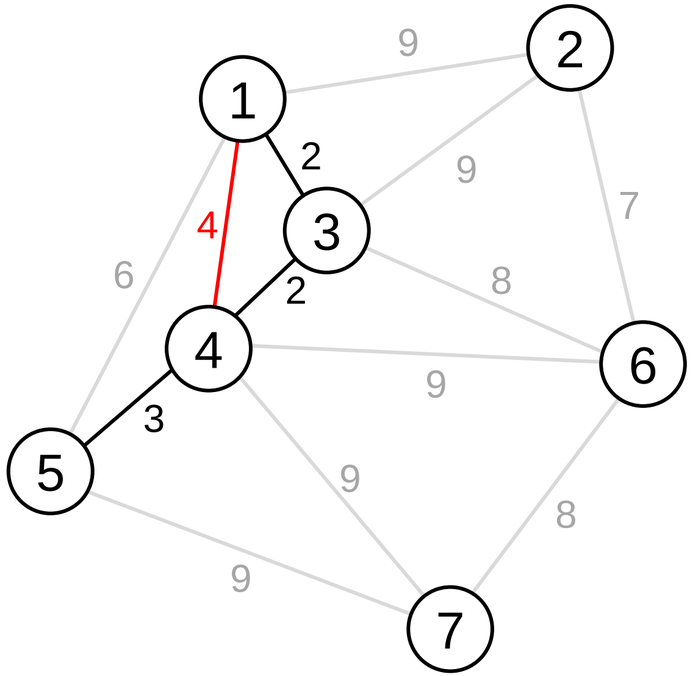
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

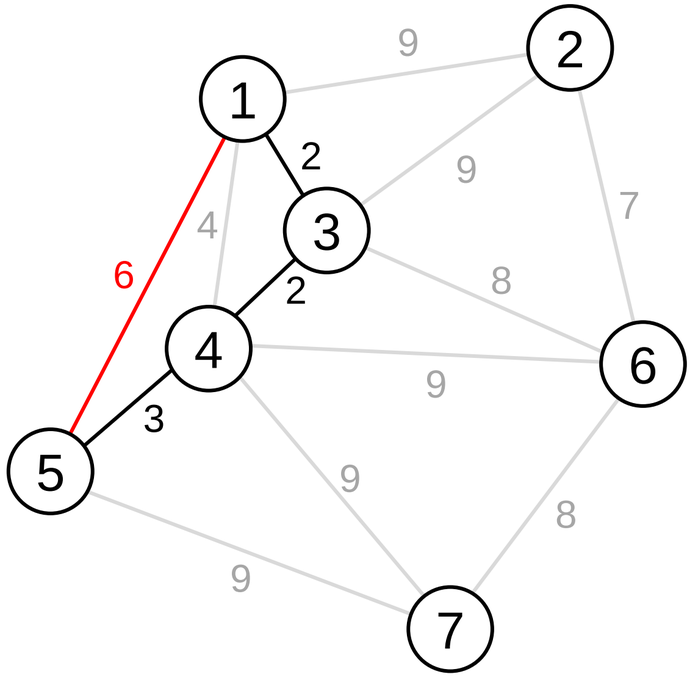
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

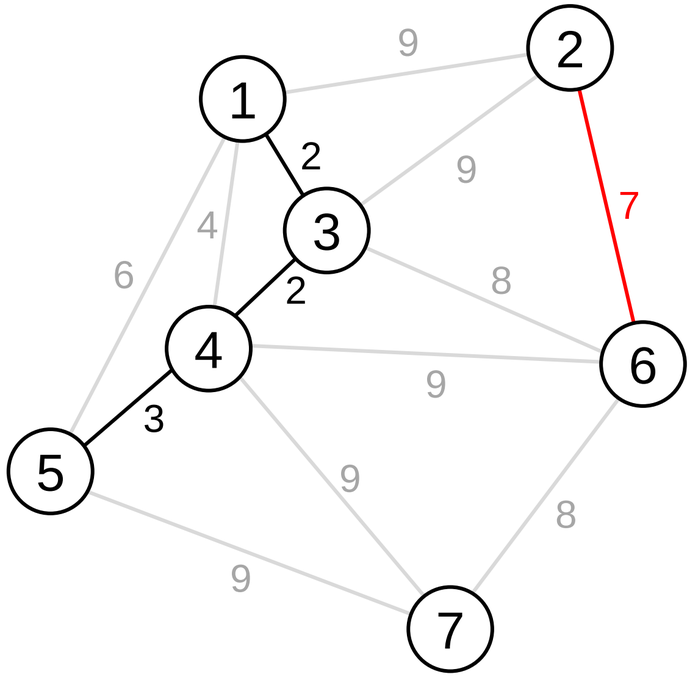
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

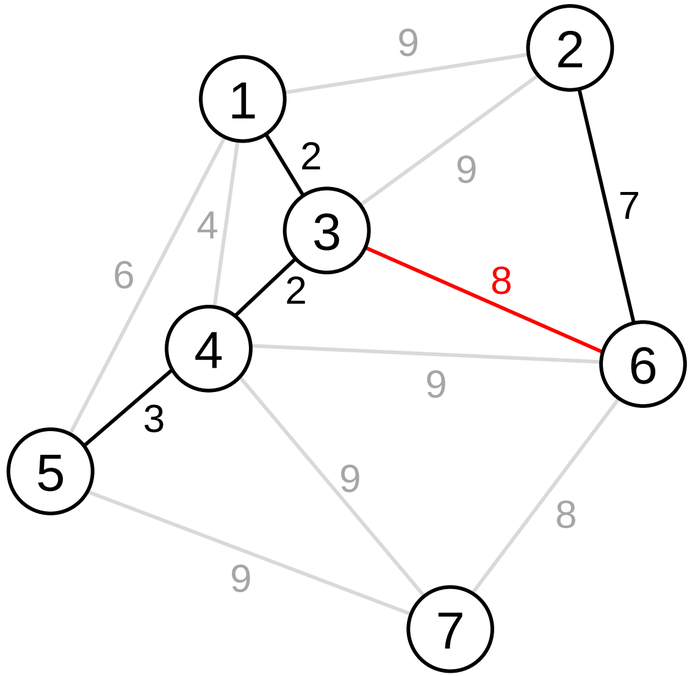
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

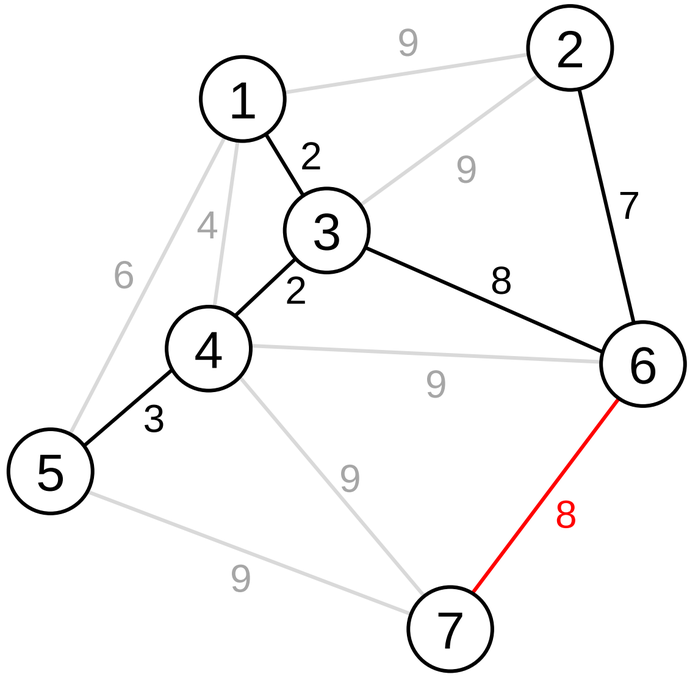
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

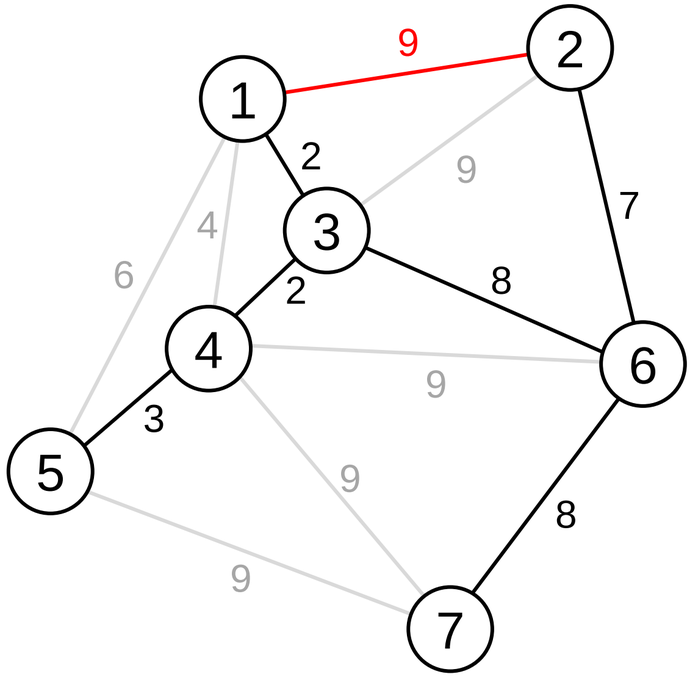
(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



Minimum Spanning Tree – Algorithms

Kruskal's Algorithm – Process

(1, 3) – 2
(3, 4) – 2
(4, 5) – 3
(1, 4) – 4
(1, 5) – 6
(2, 6) – 7
(3, 6) – 8
(6, 7) – 8
(1, 2) – 9
(2, 3) – 9
(4, 6) – 9
(4, 7) – 9
(5, 7) - 9



2022 Winter Algorithm Camp

Minimum Spanning Tree

Kruskal's Algorithm – Implementation

Issue:

1) 간선을 선택하여 두 서브트리를 연결한다.
2) 간선을 선택하게 될 경우 사이클이 생긴다.

2022 Winter Algorithm Camp

Minimum Spanning Tree

Kruskal's Algorithm – Implementation

Issue:

1) 간선을 선택하여 두 서브트리를 연결한다.
2) 간선을 선택하게 될 경우 사이클이 생긴다.

Union Find

```
KRUSKAL(G):
  A <- []
  for each vertex v in G.V:
    Make_Set(v)
  for each edge (u,v) in G.E ordered by weight in increasing order(u,v):
    if Find(u) != Find(v):
      A.append((u,v))
      Union(u,v)
  return A
```

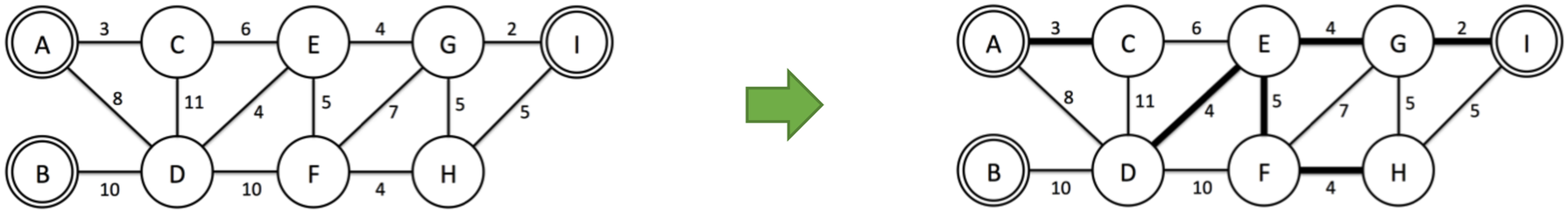
2022 Winter Algorithm Camp

10423. 전기가 부족해

N개의 도시, M개의 케이블 수, 발전소 K개 $(1 \le N \le 1000, 1 \le M \le 100{,}000, 1 \le K \le N)$

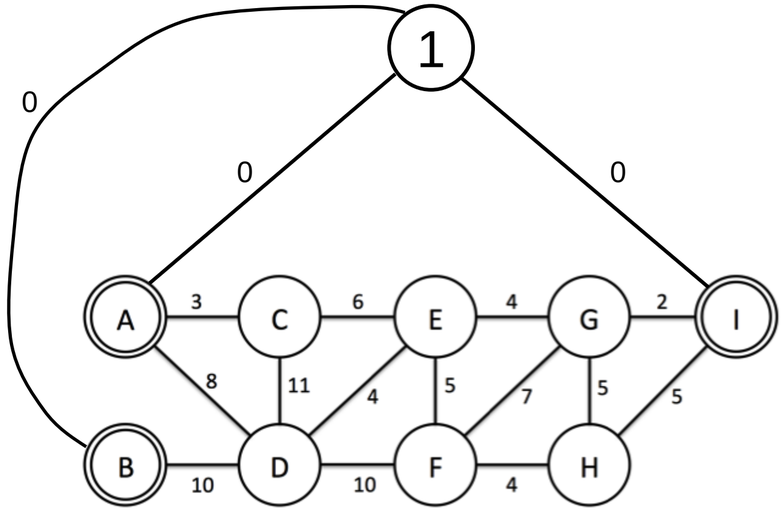
양방향, 비용(가중치)가 있는 케이블 $(1 \le u, v \le N, 1 \le w \le 10000)$

발전소로부터 전원이 공급될 때, 모든 도시에 전기를 공급할 수 있도록 케이블을 설치하는데
드는 최소비용을 구해보자.



2022 Winter Algorithm Camp

10423. 전기가 부족해



Minimum Spanning Tree – Algorithms

* Prim's Algorithm

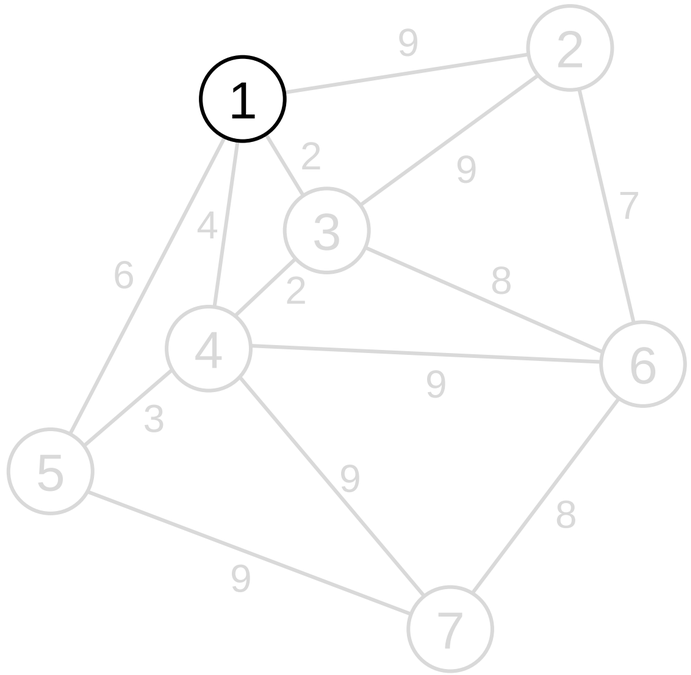
- 정점을 append하는 방식
- 임의의 노드로부터 새 노드와 트리를 연결하는 노드 중 가중치가 가장 작은 간선 선택
- Single Source Shortest Path 알고리즘 중 Dijkstra's algorithm과 유사

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

임의의 정점 선택



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

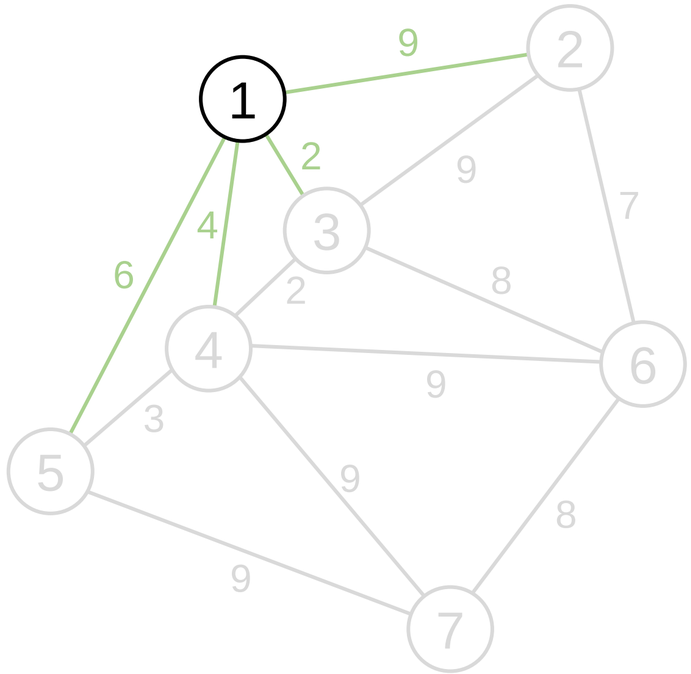
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

새로운 노드와 트리를 연결하는 간선들 선택



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

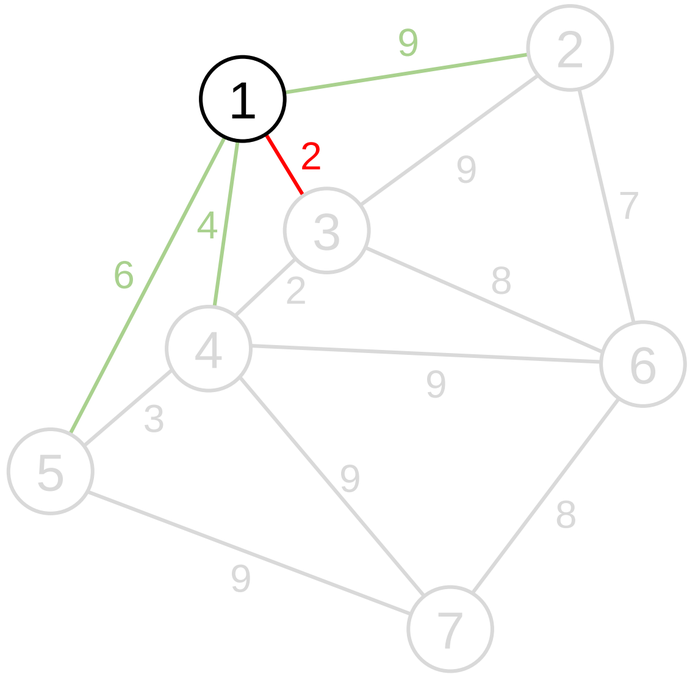
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

선택된 간선들 중 가중치가 가장 작은 간선 선택



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

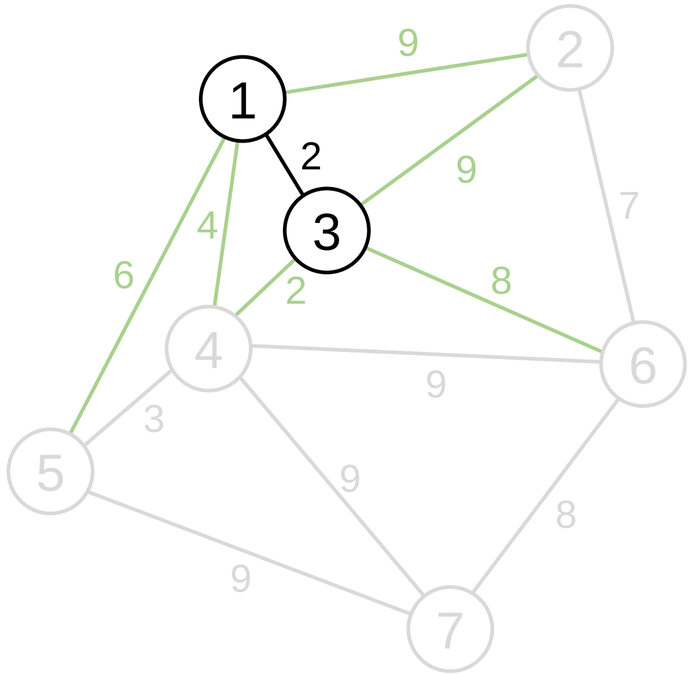
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

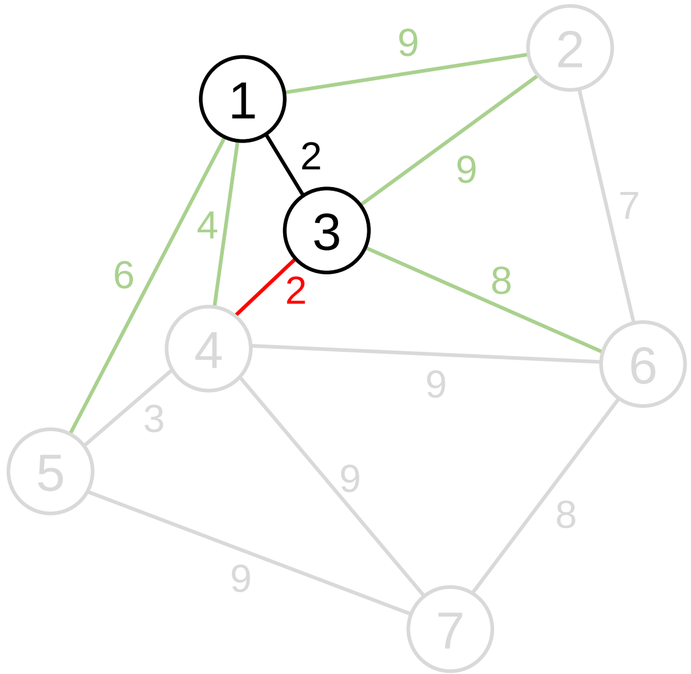
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

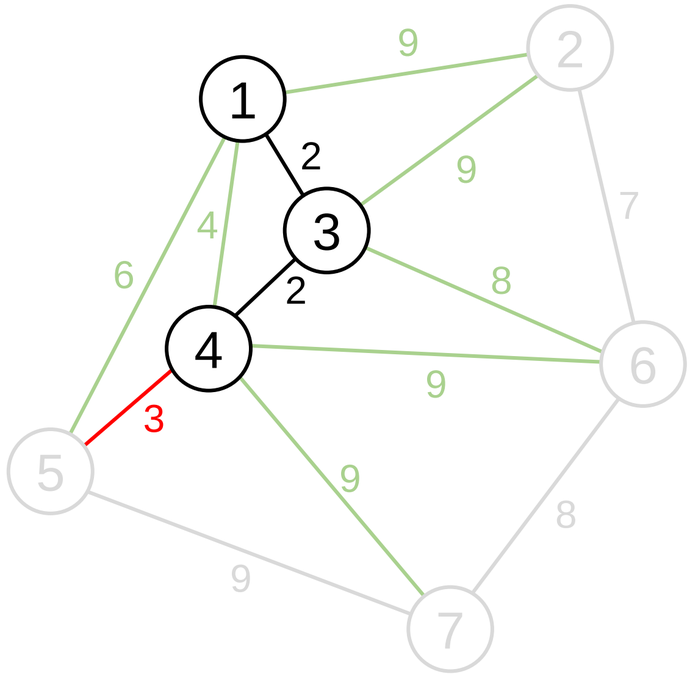
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

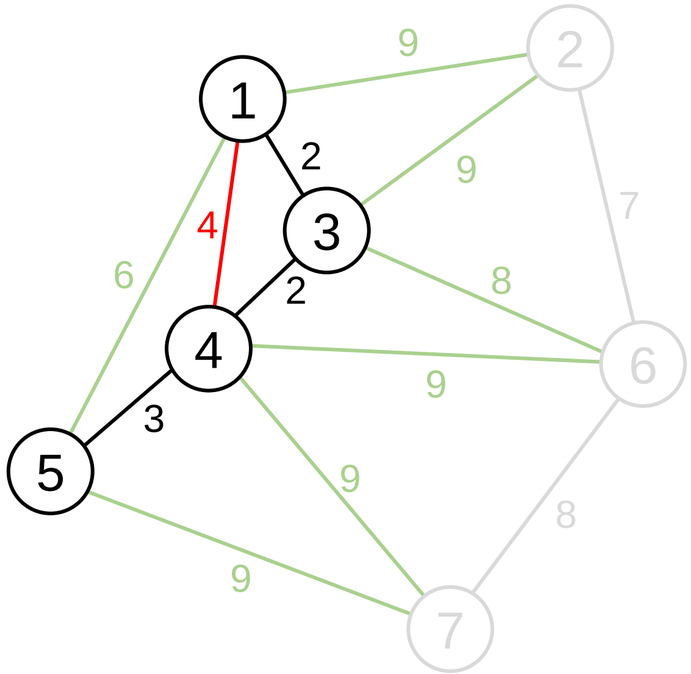
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

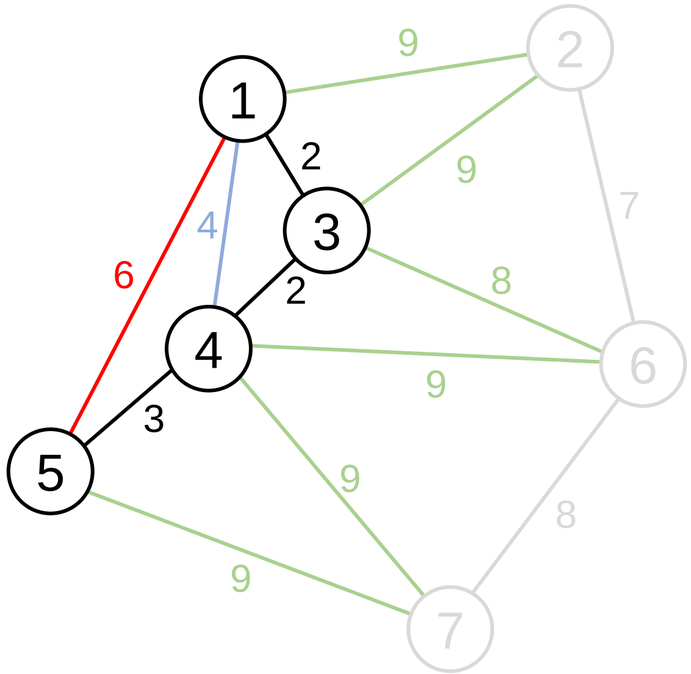
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

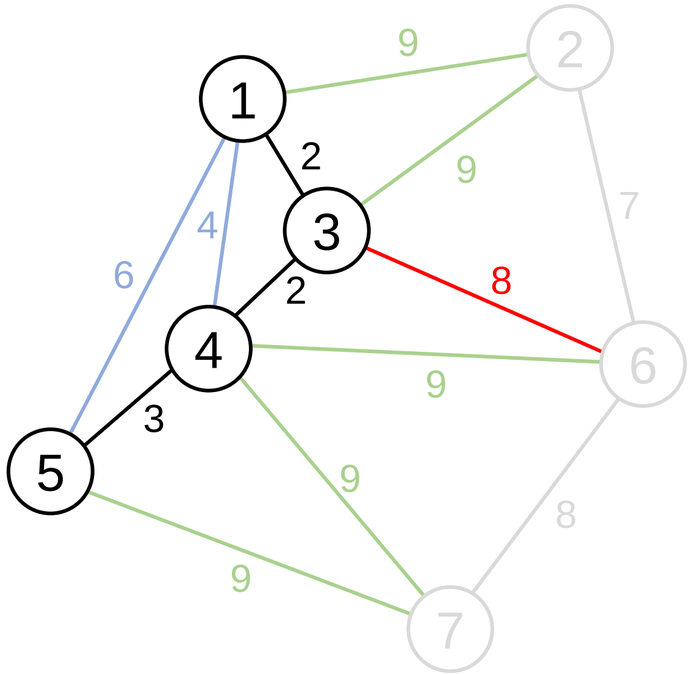
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

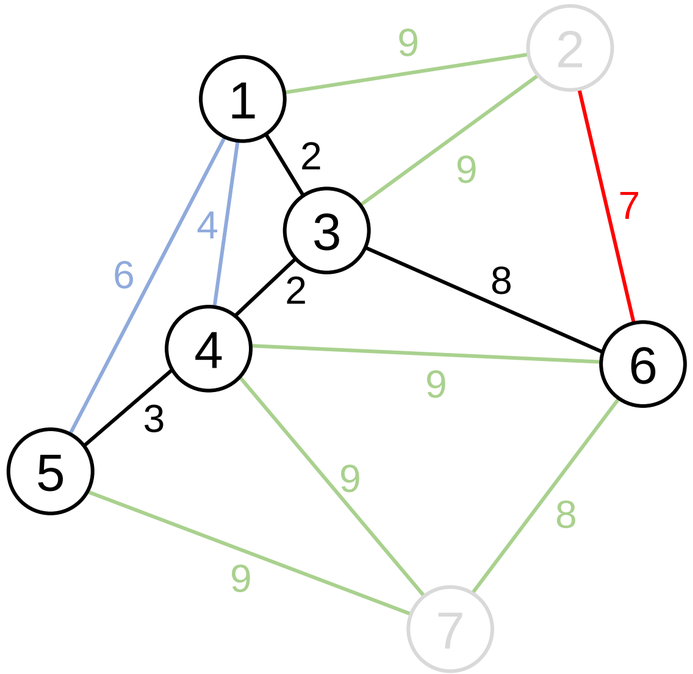
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

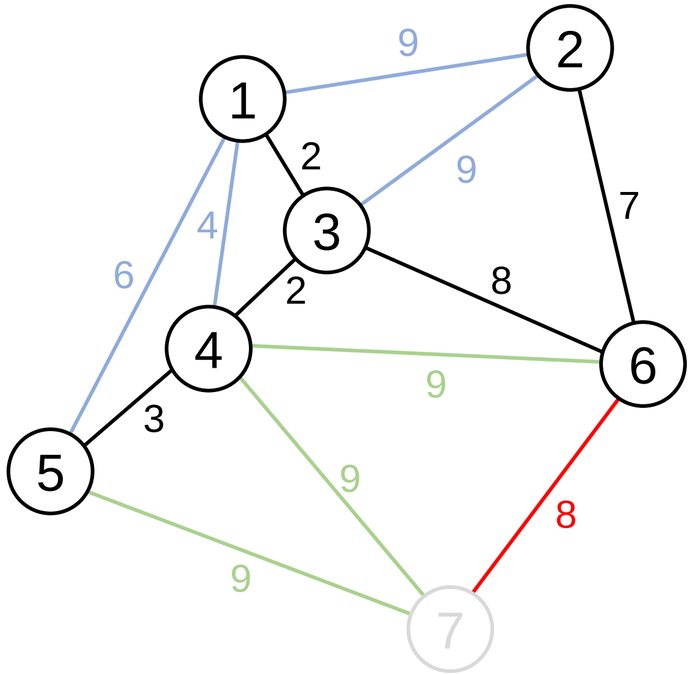
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

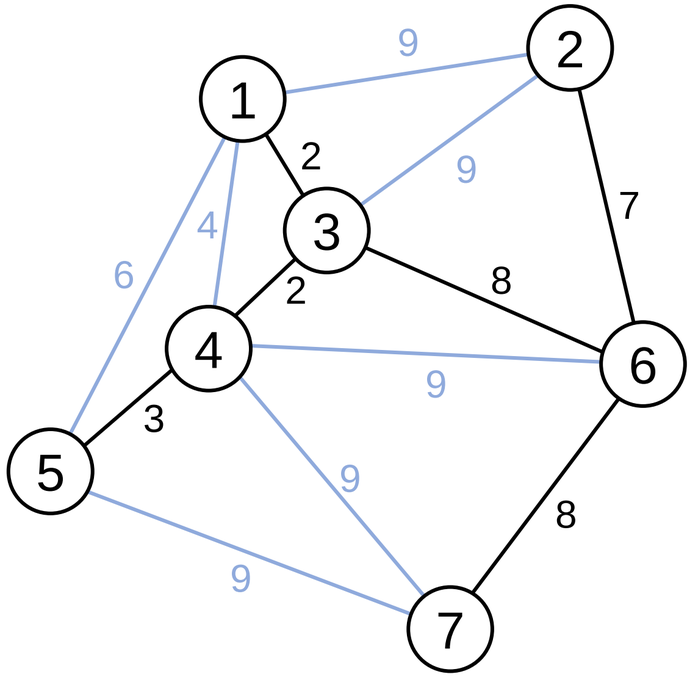
chosen edge

2022 Winter Algorithm Camp

Minimum Spanning Tree – Algorithms

Prim's Algorithm – Process

연결된 노드에서 같은 행위 반복



Unseen edge

Discarded edge

edge in queue

minimum weight edge in queue

chosen edge

Minimum Spanning Tree

Prim's Algorithm – Implementation

Issue:

1) 선택된 후보 간선들 중 가중치가 가장 작은 간선 선택: Priority Queue
2) 사이클 체크: visit check

```
1 Prim(G, w, r):
2   for each u in G.V:
3     u.key = inf
4
5   r.key = 0
6   Q = G.V
7   while Q != empty:
8     u = extract_min(Q)
9     for each v in G.adj[u]:
10      if v in Q and w(u,v) < v.key:
11        v.key = w(u,v)
```

2022 Winter Algorithm Camp

20390. 완전그래프의 최소 스패닝 트리

N개의 정점으로 구성된 완전그래프 $(1 \leq N \leq 10{,}000)$

간선의 가중치는 $O(1)$에 구할 수 있다.

시간 제한 5초, 메모리 제한 16MB 안에 최소 신장 트리의 가중치를 구해보자.

2022 Winter Algorithm Camp

20390. 완전그래프의 최소 스패닝 트리

모든 간선들의 정보를 담은 리스트를 만들 수 없다.

+ 각 정점으로부터 다른 모든 정점들에 대해 고려를 한다면

정점마다 $O(N)$만큼의 반복

정점의 개수 N개 => $O(N^2)$의 시간으로 prim algorithm 적용 가능

Appendix – Additional Topics

* Small to Large

Disjoint Set의 Union 연산을 하는 과정에서, 시간복잡도를 고려하여 높이가 낮은 트리를 높은 트리에 합쳐야 함을 보였습니다.

반복적으로 집합을 합치는 연산을 하여 최종적으로 모든 집합이 하나의 집합으로 합쳐지는 상황을 가정해봅시다. 그리고 합쳐지는 과정에서 특정 집합에 포함된 수의 종류를 묻는 쿼리도 있다고 해봅시다.

종류를 묻는 쿼리를 처리하기 위해서는 단순히 집합의 크기 관리만으로는 힘들기 때문에 set과 같은 자료구조를 써야 합니다. 이후 집합을 합친다면 set을 합치는 연산을 하게 되고, 한 집합의 개수에 비례하는 시간이 걸리게 됩니다.

Disjoint Set의 Union과 유사하게, 집합의 크기가 작은 것을 큰 집합에 합쳐주게 된다면 각 원소가 이동하는 최대 횟수는 $O(logN)$이 됩니다. 최종적으로 모든 원소의 이동횟수는 $O(NlogN)$이고, set등으로 자료를 관리한다 해도 $O(Nlog^2N)$의 시간복잡도를 갖습니다.

2022 Winter Algorithm Camp

Appendix – Problems

필수문제

1717	집합의 표현
4803	트리
13904	과제
6497	전력난
10423	전기가 부족해
20390	완전그래프의 최소 스패닝 트리

연습문제

20040	사이클 게임
18116	로봇 조립
16562	친구비
10775	공항
17619	개구리 점프
1197	최소 스패닝 트리
14167	Moocast
13151	Model Railroad
23034	조별과제 멈춰!