Sogang ICPC Team

2021-1 중급 스터디 6회차

Tree

이윤제

yjyj102742@gmail.com

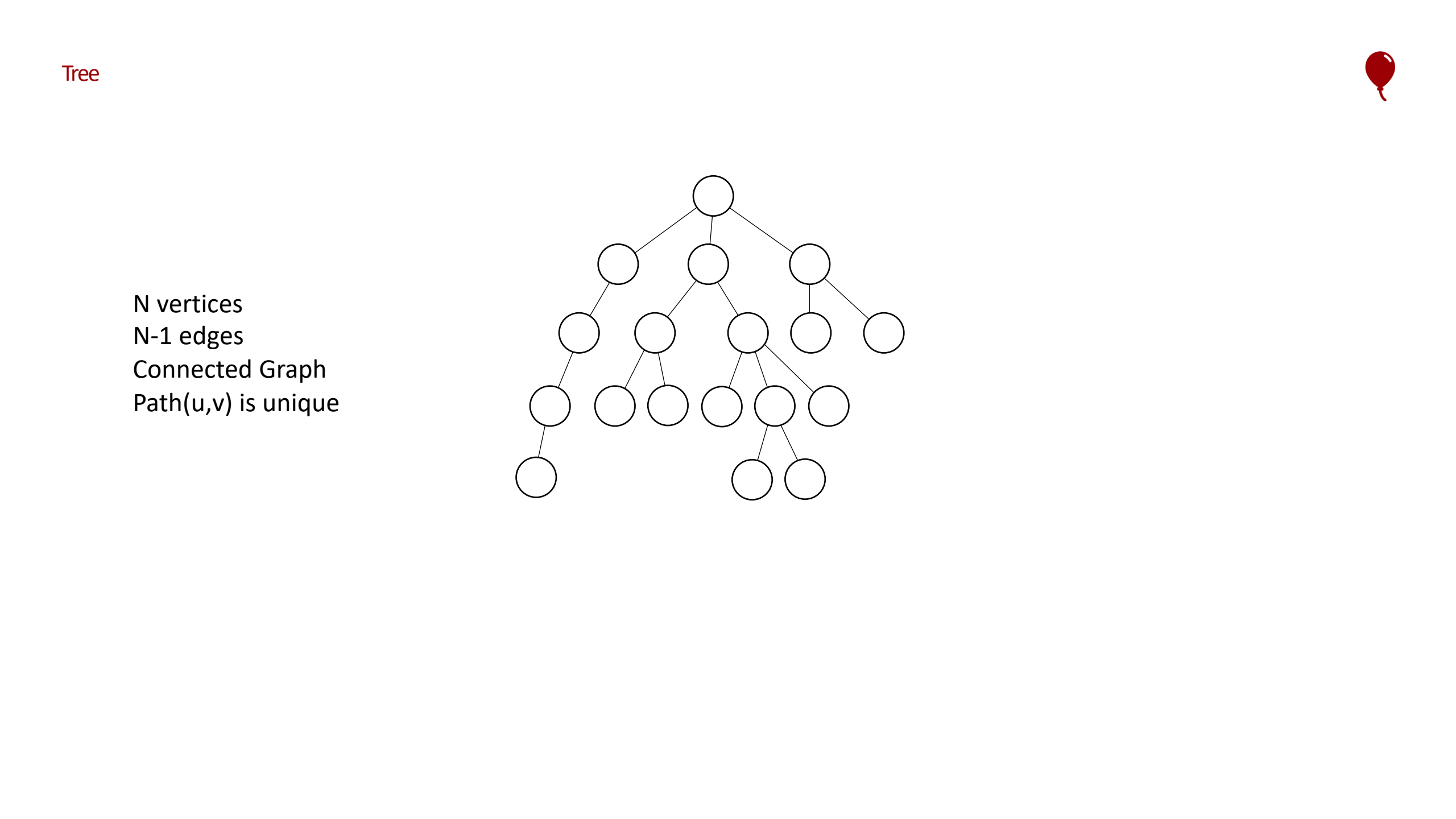
목차

- 선수지식
- Lowest Common Ancestor
- Dynamic Programming on Tree
- Complete Binary Tree
- Tree-like Graph

선수 지식

- Tree
- Sparse Table

Tree



N vertices
N-1 edges
Connected Graph
Path(u,v) is unique

- f: X -> X 로 정의된 function에서 $f^n(x)$의 값을 빠르게 구하기 위한 자료구조

- 아래의 표에서 $f^5(1)$은?

x	1	2	3	4	5	6
f(x)	3	2	4	6	1	5

- f: X -> X 로 정의된 function에서 $f^n(x)$의 값을 빠르게 구하기 위한 자료구조

- $f^5(1) = f(f^4(1))$
- $f^{11}(1) = f(f^2(f^8(1)))$
- …

x	1	2	3	4	5	6
f(x)	3	2	4	6	1	5

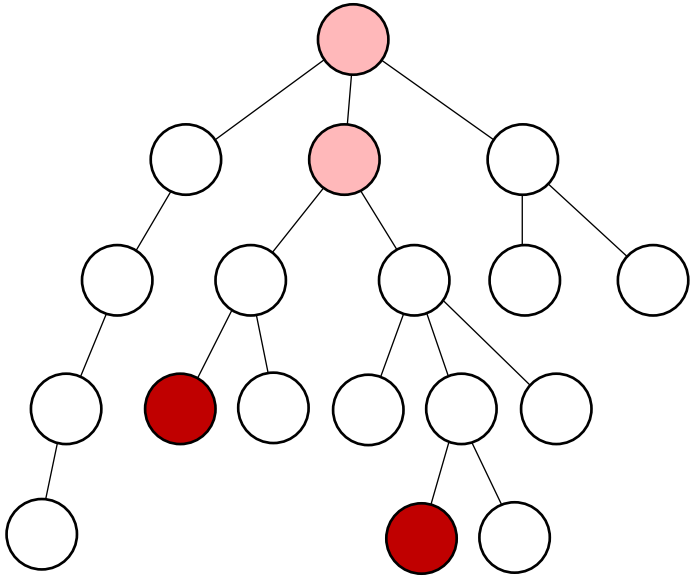
- f: X -> X 로 정의된 function에서 $f^n(x)$의 값을 빠르게 구하기 위한 자료구조
- $sparse[x][i]$: x에 함수f를 2^i번 가한 값
- $sparse[x][i] = sparse\big[sparse[x][i-1]\big][i-1]$

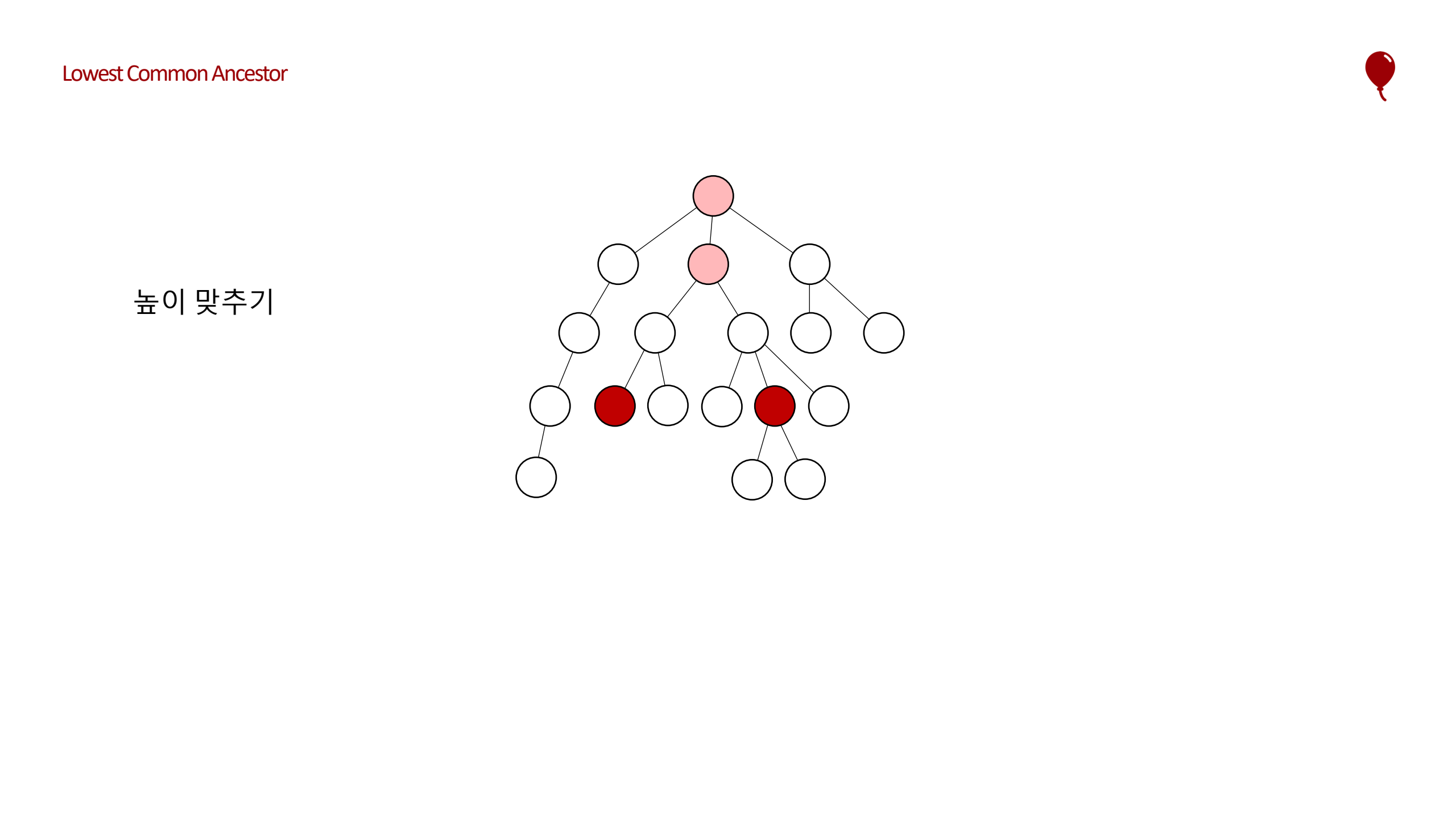
x	1	2	3	4	5	6
f(x)	3	2	4	6	1	5
$f^2(x)$	4	2	6	5	3	1
$f^4(x)$	5	2	1	3	6	**4**

Lowest Common Ancestor

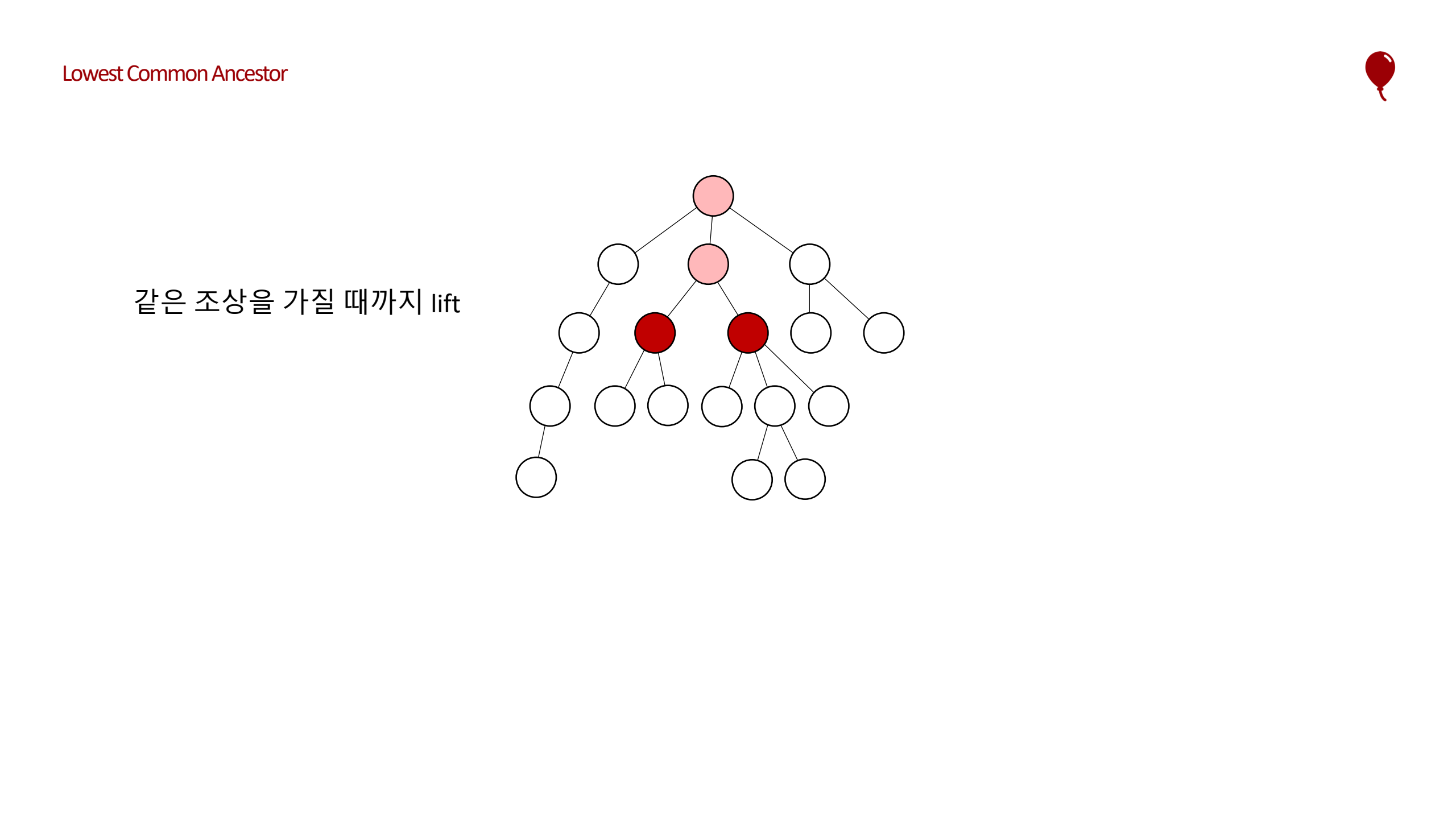
- k-th ancestor
- lca

Lowest Common Ancestor



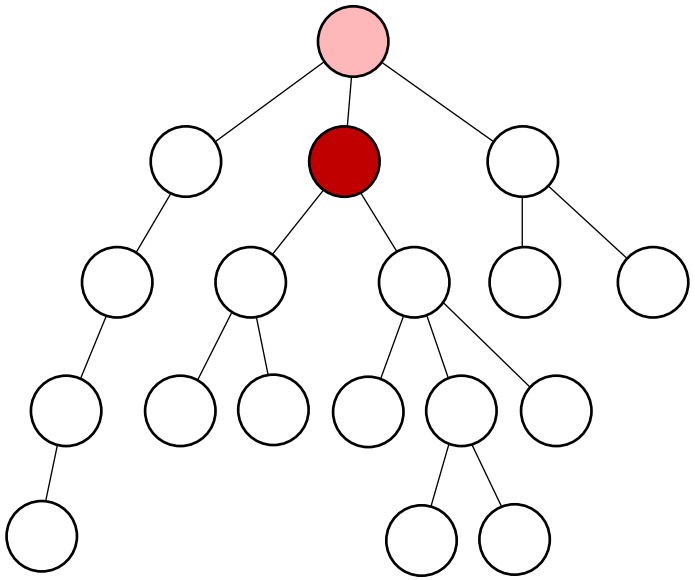


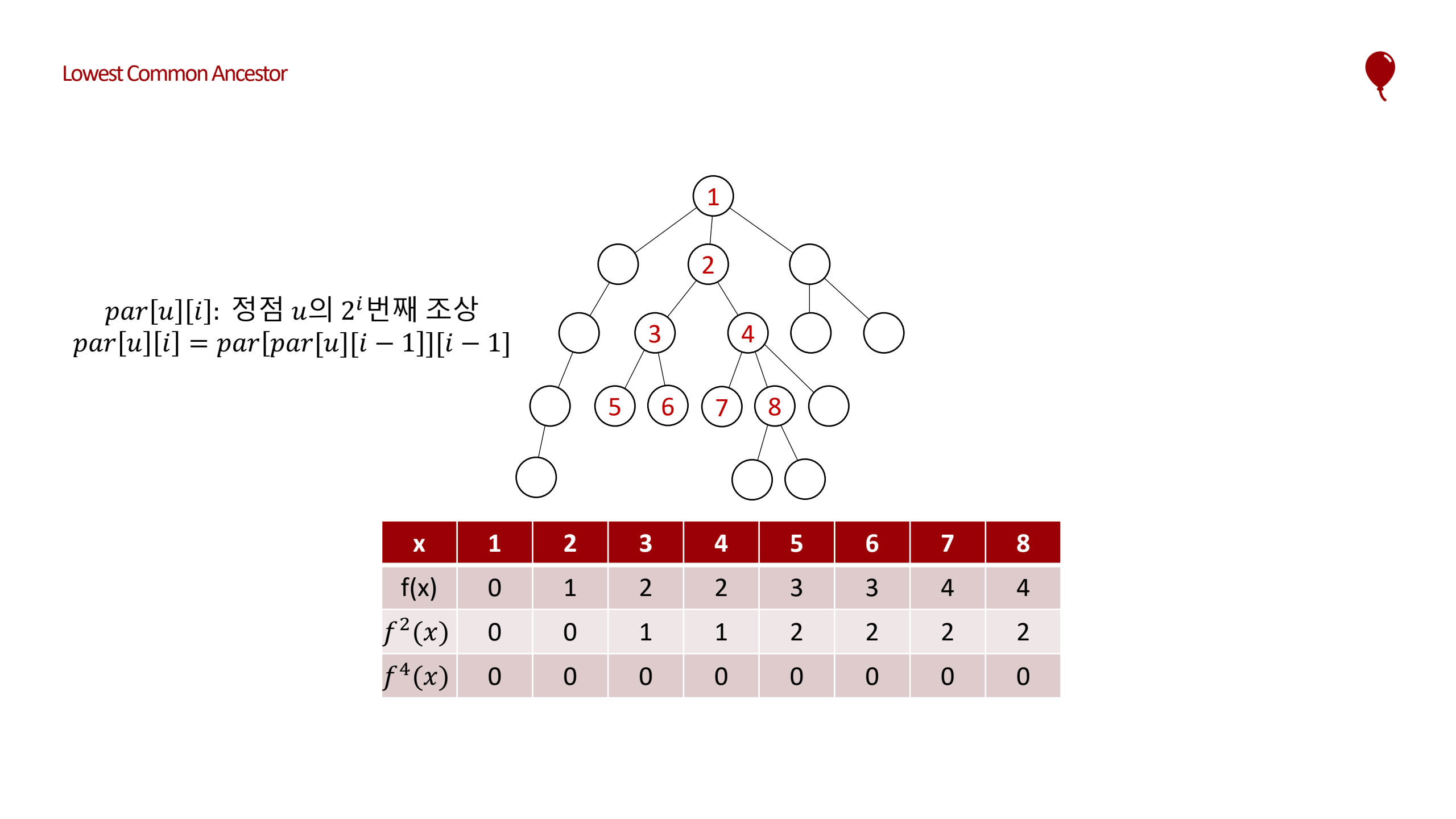
높이 맞추기



같은 조상을 가질 때까지 lift

찾았다!

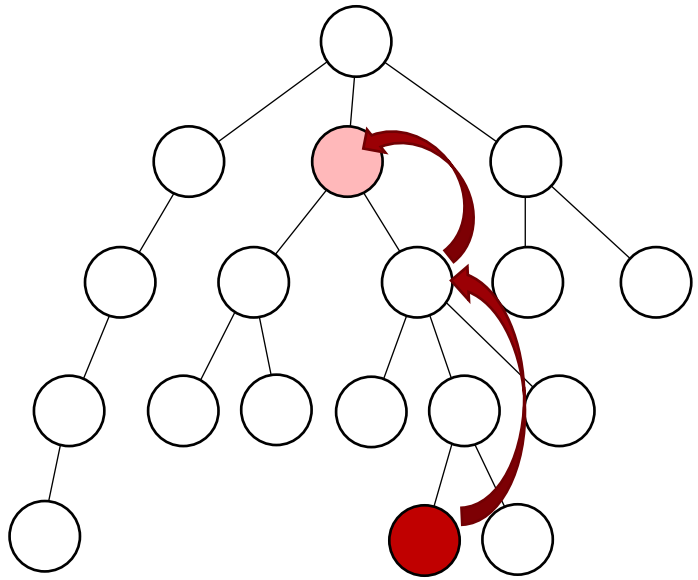




$par[u][i]$: 정점 u의 2^i번째 조상

$par[u][i] = par[par[u][i-1]][i-1]$

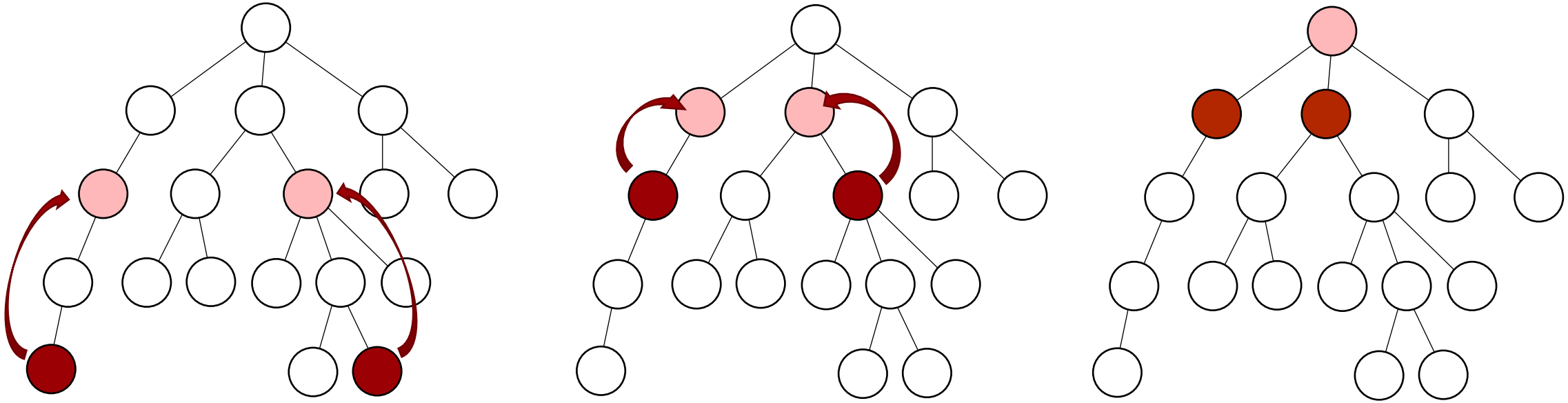
x	1	2	3	4	5	6	7	8
f(x)	0	1	2	2	3	3	4	4
$f^2(x)$	0	0	1	1	2	2	2	2
$f^4(x)$	0	0	0	0	0	0	0	0



K=3

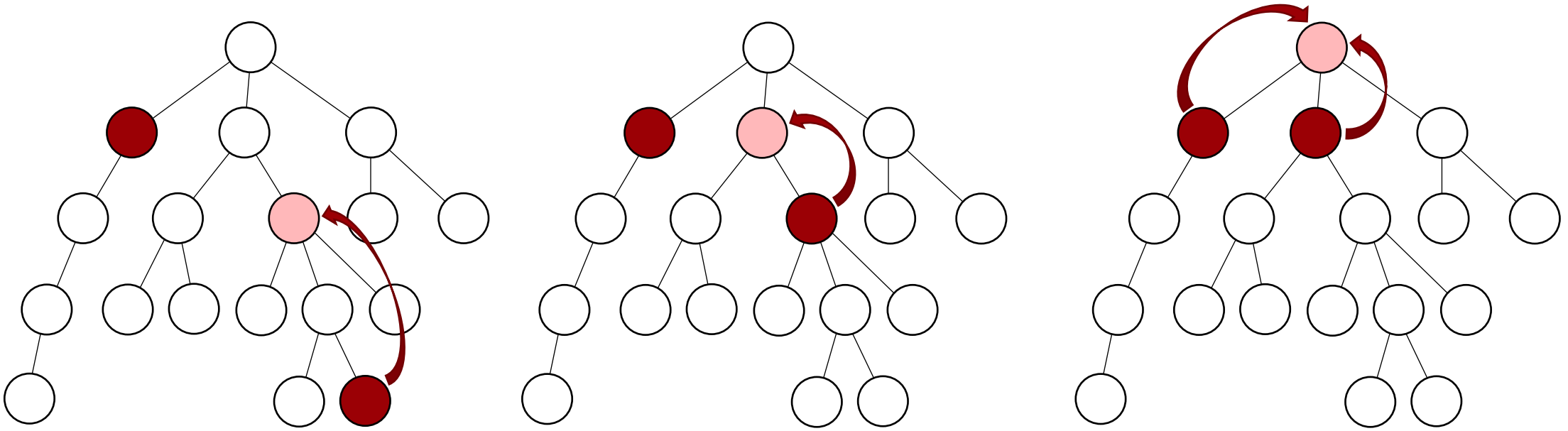
$$i = 20 \sim 0: if\big(k \geq (1 \ll i)\big) u = par[u][i], k \ -= (1 \ll i)$$

Lowest Common Ancestor



$$i = 20 \sim 0:\ if\,(par[u][i] == par[v][i])\ continue$$
$$else\ u = par[u][i], v = par[v][i]$$

Lowest Common Ancestor

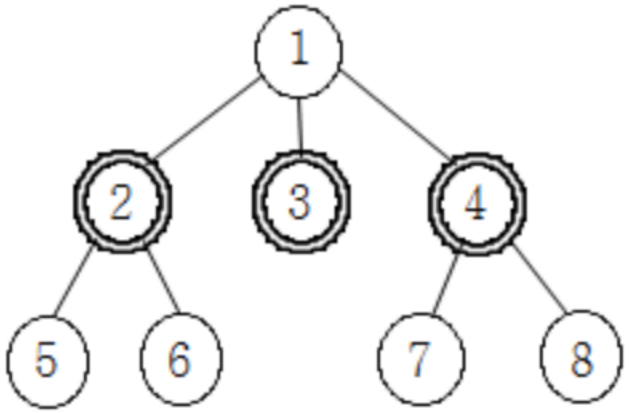


- Sparse Table을 이용해 LCA를 다음 과정으로 구할 수 있다.
  - 높이 맞추기
  - 두 노드를 공통 조상이 아니면 lift
  - 최종적으로 return $par[u][0]$
- 여러 쿼리에 적용 가능
  - 두 노드의 거리: $dist(u,v) = dist(u,lca) + dist(v,lca)$
  - 두 노드 경로상의 최대값: $max(u,v) = \max(max(u,lca), max(v,lca))$
  - 등등...
- 샘플코드
  - https://github.com/Weaasel/Algorithm-ProblemSolving/blob/master/BOJ/LCA/11438.cpp

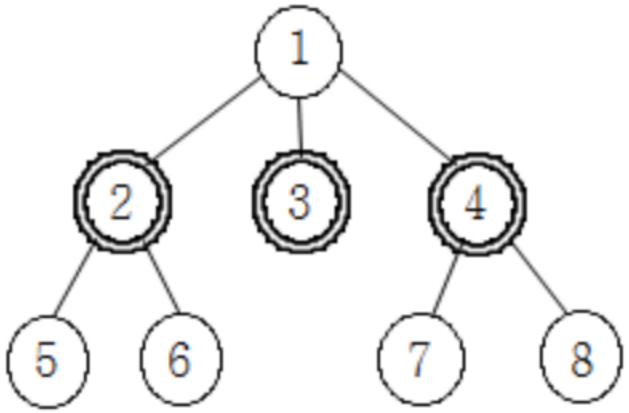
Dynamic Programming on Tree

- 0/1 State
- Preprocess / dfs

- 각 정점은 얼리어답터 or Not
- 얼리어답터가 아닌 정점은 인접한 정점들이 모두 얼리어답터여야 한다.
- 조건을 만족시키기 위한 최소 얼리어답터 수

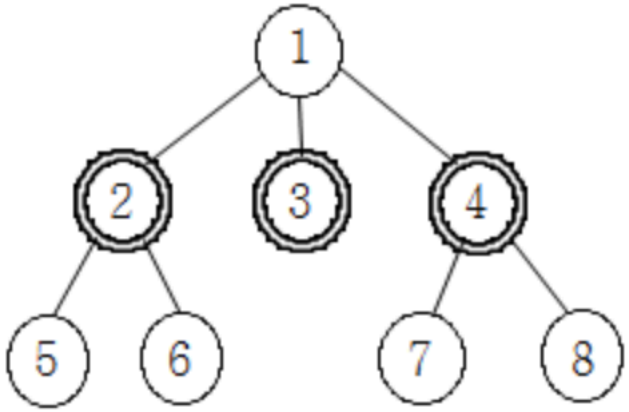


- 각 정점은 얼리어답터 or Not
- 얼리어답터가 아닌 정점은 인접한 정점들이 모두 얼리어답터여야 한다.
- 조건을 만족시키기 위한 최소 얼리어답터 수

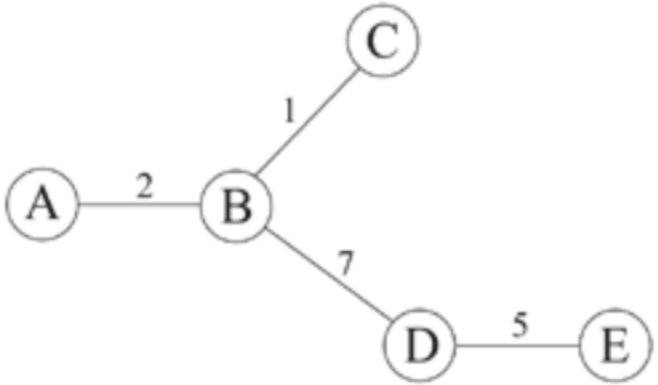


- State를 나눠서 생각해 보자
  - $Dp[u][type]$: 정점u가 $type$일때 $subtree\ u$의 답

- *Type 0: Early / 1: Not*
- $Dp[u][0] = 1 + \sum_{u \to v} \min(dp[v][0], dp[v][1])$
- $Dp[u][1] = \sum_{u \to v} dp[v][0]$
- 샘플코드
  - https://github.com/Weaasel/Algorithm-ProblemSolving/blob/master/BOJ/DP/2533.cpp

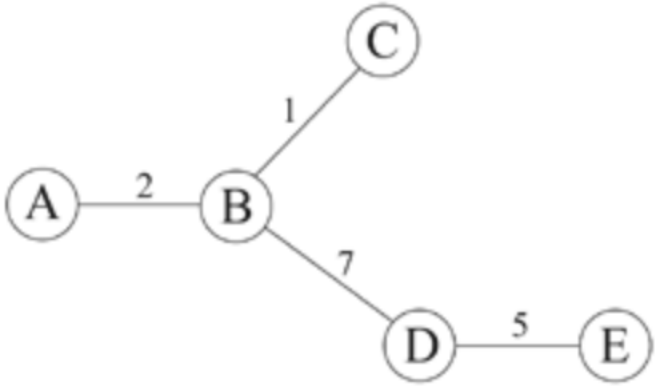


- 모든 정점에 대해 다른 정점들과 거리의 합 구하기
- $sum[B] = 2 + 1 + 7 + 12 = 22$

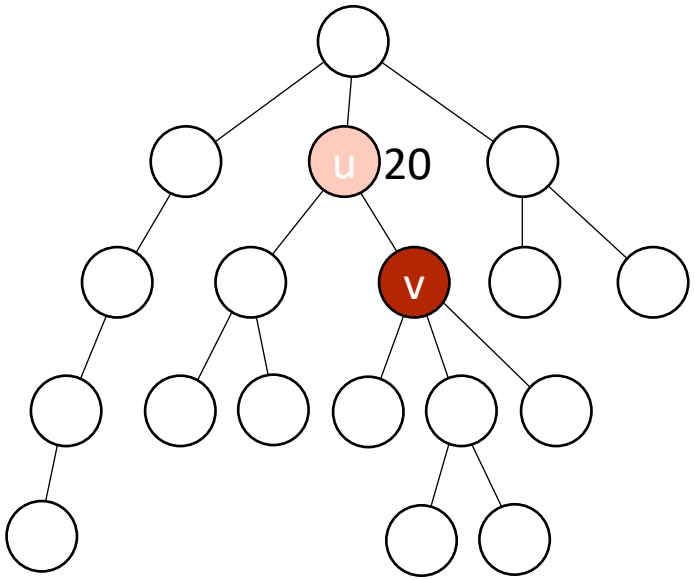


- 모든 정점에 대해 다른 정점들과 거리의 합 구하기
- $sum[B] = 2 + 1 + 7 + 12 = 22$

- $sum[u]$: u를 root로 하는 tree에서 dfs 한번으로 구할 수 있다.
- O(n^2)

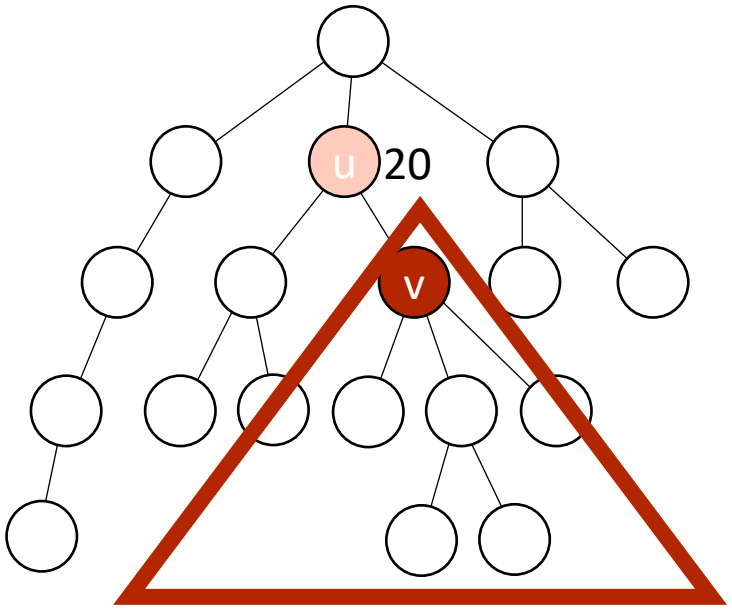


중앙 트리

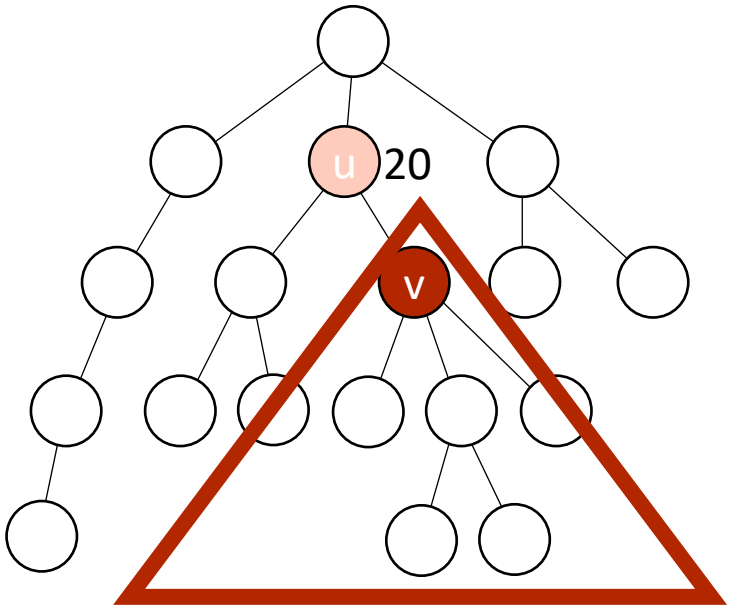


u –> v 로 이동하면, 누군가와는 멀어지고 누군가에게는 가까워진다.
$edge(u, v)$ 만큼

중앙 트리

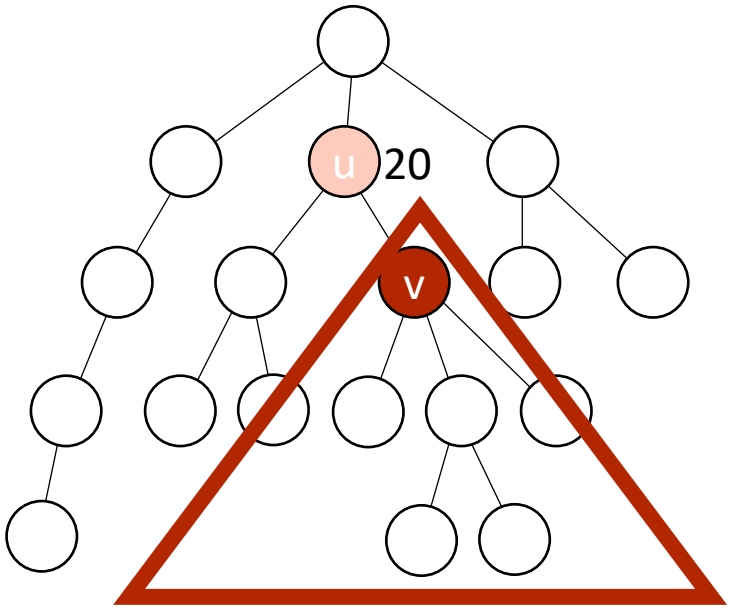


중앙 트리



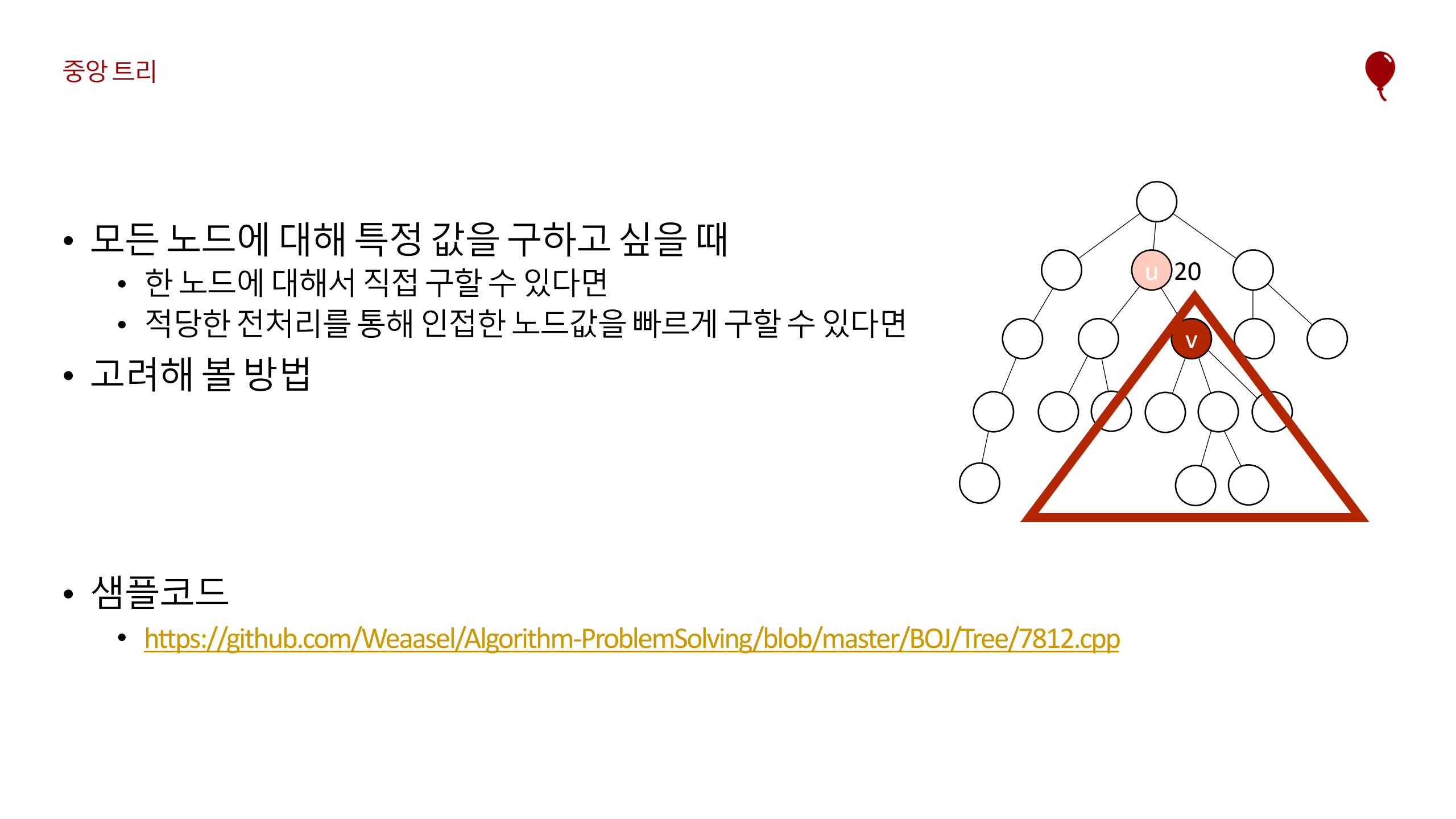
$$res[v] = res[u] + edge(u,v) * (n - subs[v]) - edge(u,v) * subs[v]$$

중앙 트리



$$res[v] = res[u] + edge(u,v) * (n - subs[v]) - edge(u,v) * subs[v]$$

: subs 전처리해두면 O(1), res[root]는 직접 구하기



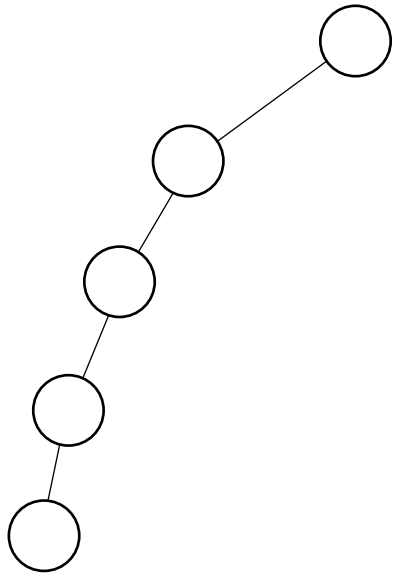
- 모든 노드에 대해 특정 값을 구하고 싶을 때
  - 한 노드에 대해서 직접 구할 수 있다면
  - 적당한 전처리를 통해 인접한 노드값을 빠르게 구할 수 있다면
- 고려해 볼 방법

- 샘플코드
  - https://github.com/Weaasel/Algorithm-ProblemSolving/blob/master/BOJ/Tree/7812.cpp

Complete Binary Tree

- 트리문제의 Worst Case – Skewed Tree
- 몇 가지 풀이를 막는 제약

- Complete Binary Tree 라면 어떤 제약이 풀리는 지 염두하자

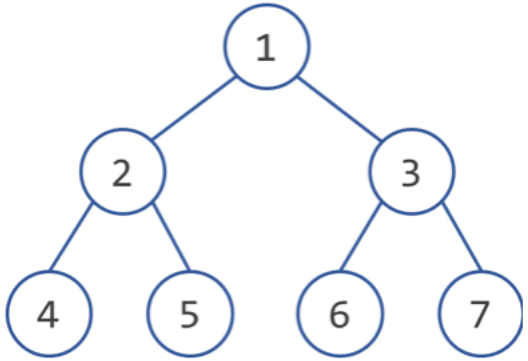


- 높이가 logn임을 이용한 풀이들

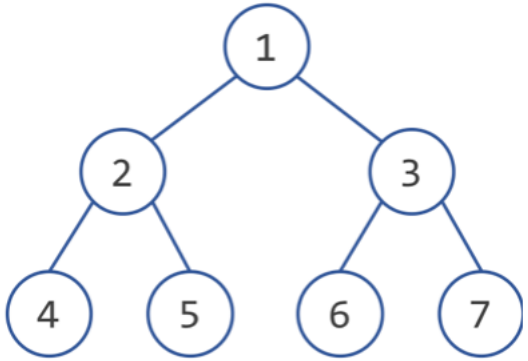
1. (u, v)에 대한 쿼리를 할 때, 직적 간선을 타고 다니면서 할 수 있다.
    - 각 쿼리마다 최대 간선 N개 -> logN개

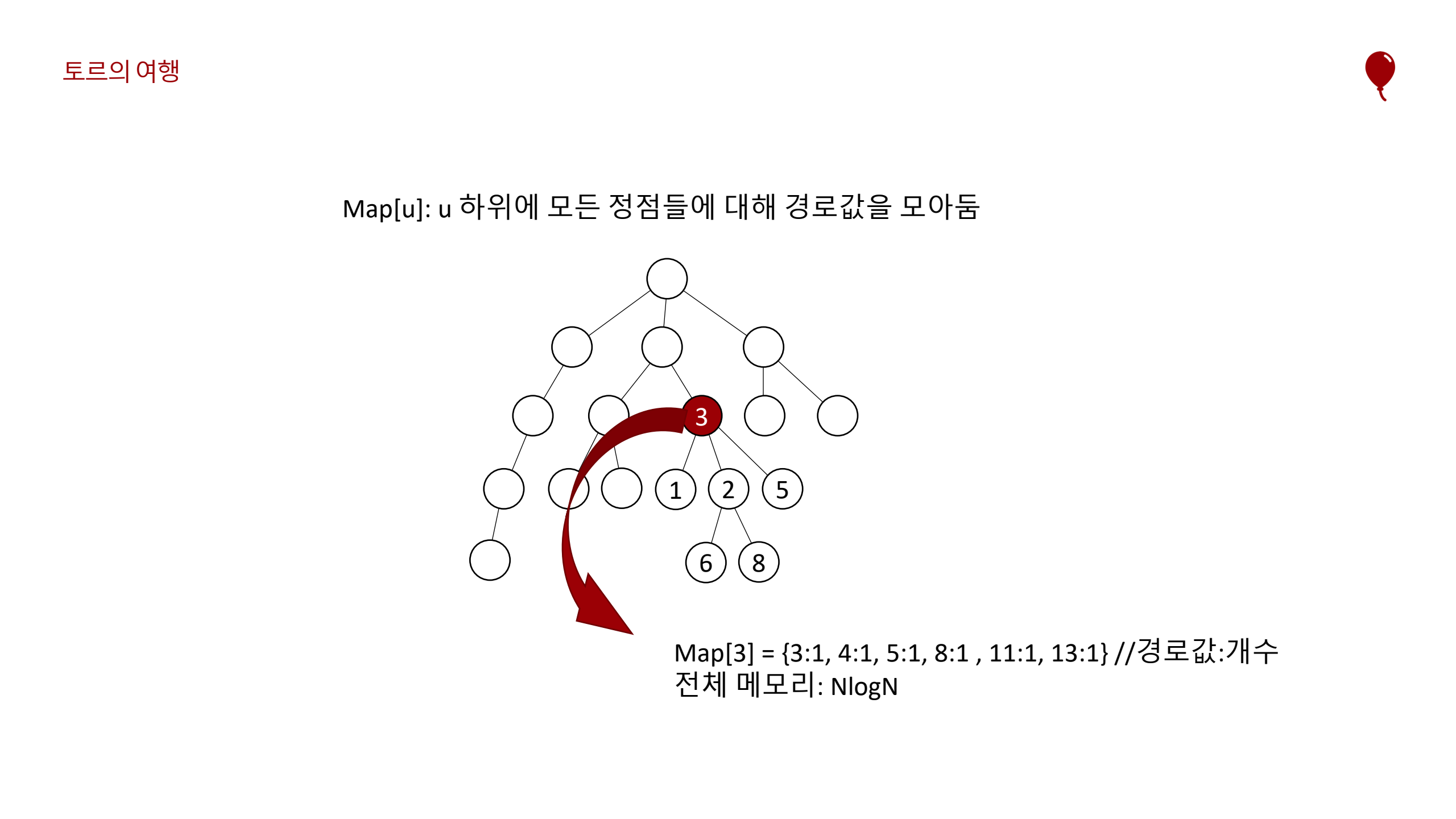
2. 하위 subtree에 대한 정보(map, set, vector…) 를 저장할 수 있다.
    - 각 level마다 최대 n개의 메모리. N^2 -> NlogN

- 주어진 시작점 u로부터 경로값이 d인 점의 개수 찾기



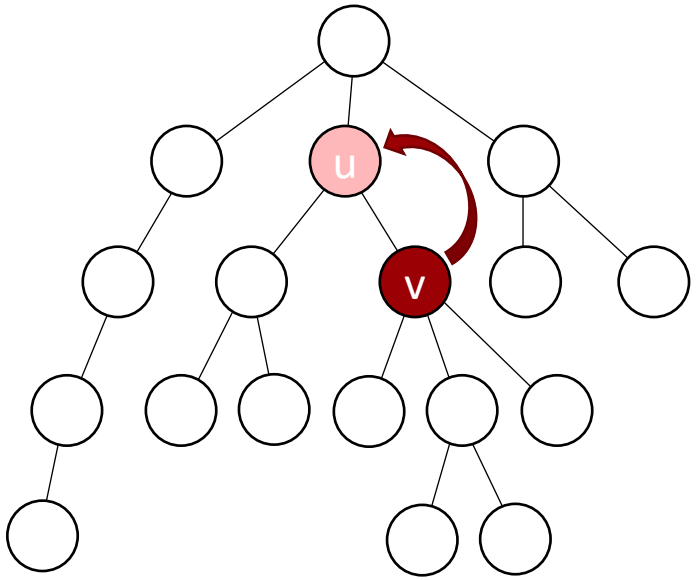
- 주어진 시작점 u로부터 경로값이 d인 점의 개수 찾기
- 하위 점들은 정보를 미리 저장할 수 있다.
- 외부 점들은 미뤄두고 하위의 개수부터 세자.



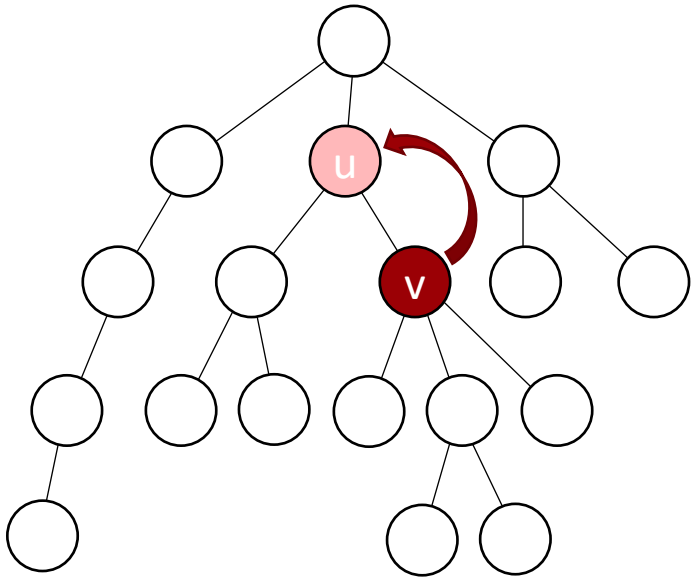


Map[u]: u 하위에 모든 정점들에 대해 경로값을 모아둠

Map[3] = {3:1, 4:1, 5:1, 8:1 , 11:1, 13:1} //경로값:개수
전체 메모리: NlogN



외부 점들과의 경로는 무조건 u를 거쳐야 함
즉, u에서부터 경로값이 $(d - val[v])$인 정점을 찾는 문제
-> root까지 반복

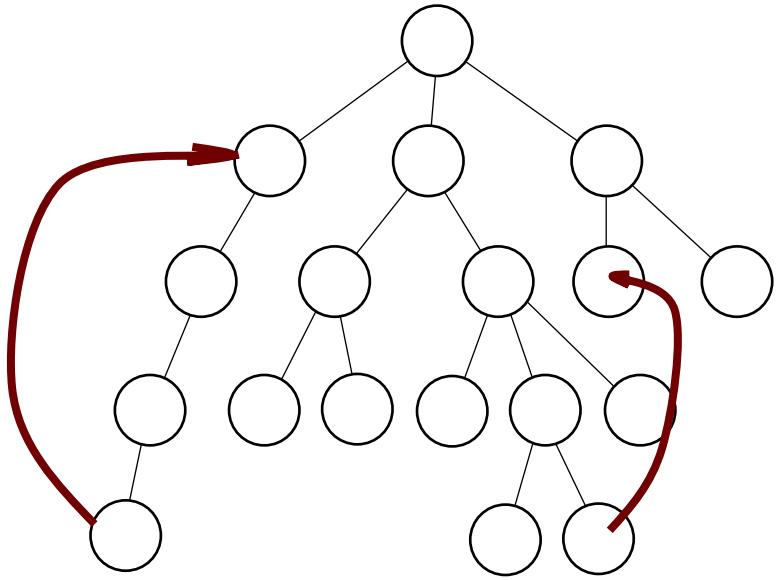


총 반복수: logN(높이)

1, 2번 특징 모두 활용.
Complete가 아니었다면 불가능

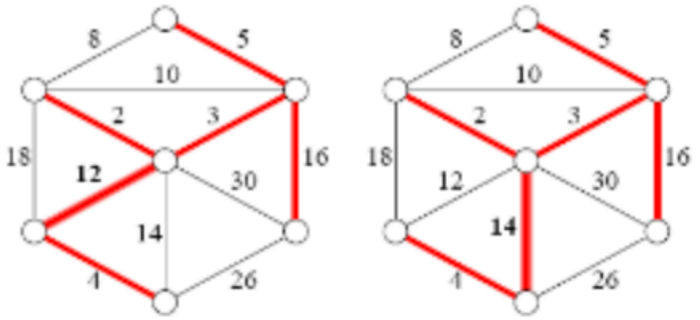
Tree-Like Graph

- 그래프가 트리와 비슷한 경우
  - 적은 수의 간선: n개, n+k개
  - 간선이 많지만 사실상 MST를 만들어서 사용
- 트리/DAG를 만들어 두고 남은 간선들에 대해 따로 고려
  - LCA, Topological Order, …



두 번째로 작은 스패닝 트리

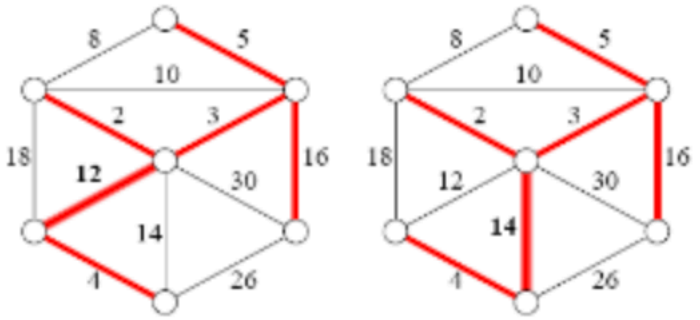
- 주어진 그래프에서
- Secondary Minimum Spanning Tree의 사이즈 구하기
- Secondary MST: MST에서 간선



MST와 second MST의 모습

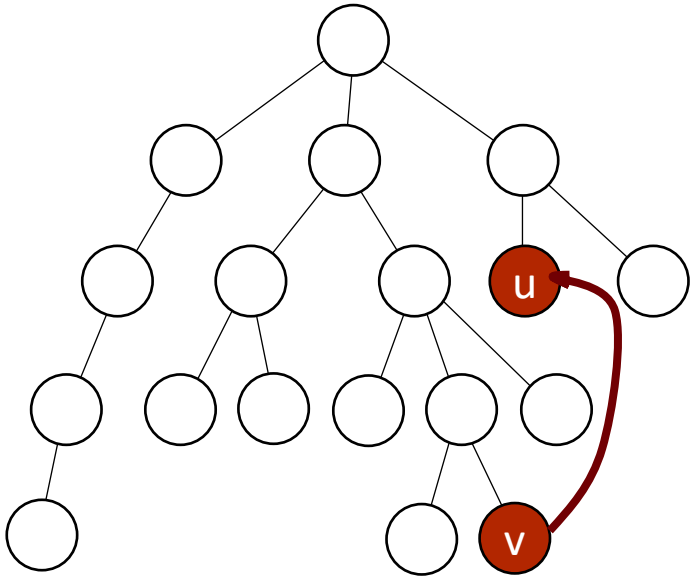
두 번째로 작은 스패닝 트리

- 주어진 그래프에서
- Secondary Minimum Spanning Tree의 사이즈 구하기
- MST에서 간선 하나만 교체
  - 최적임이 보장됨
  - 어떻게 교체?



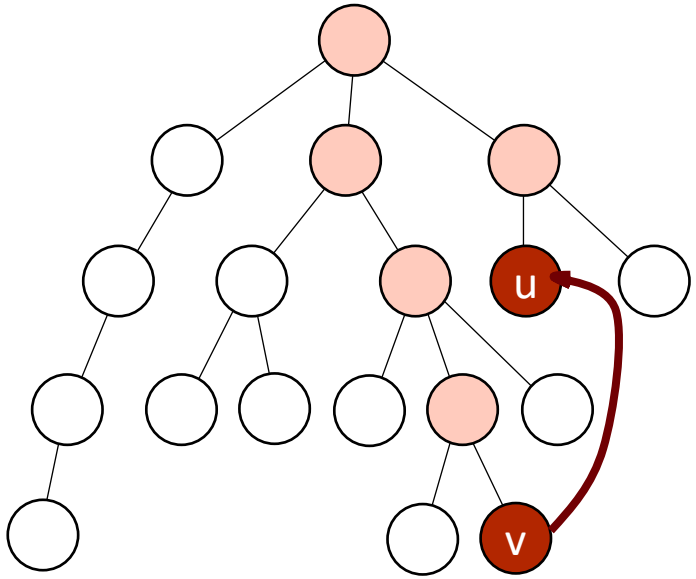
MST와 second MST의 모습

두 번째로 작은 스패닝 트리



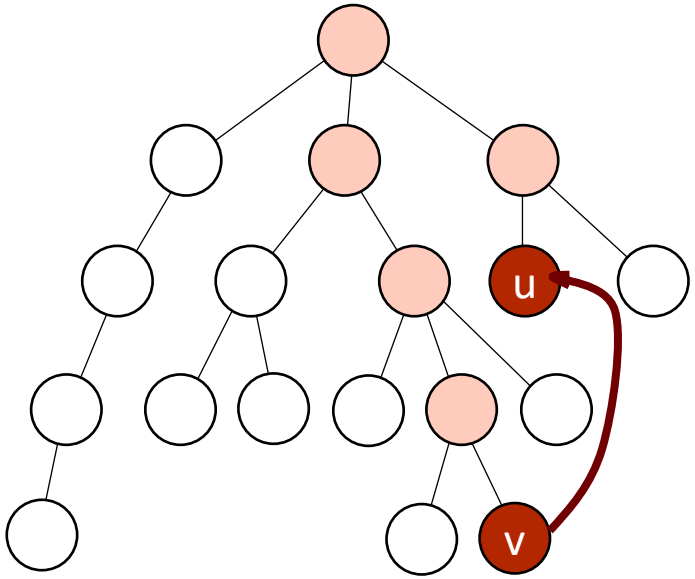
이 간선이 추가된다면?

두 번째로 작은 스패닝 트리



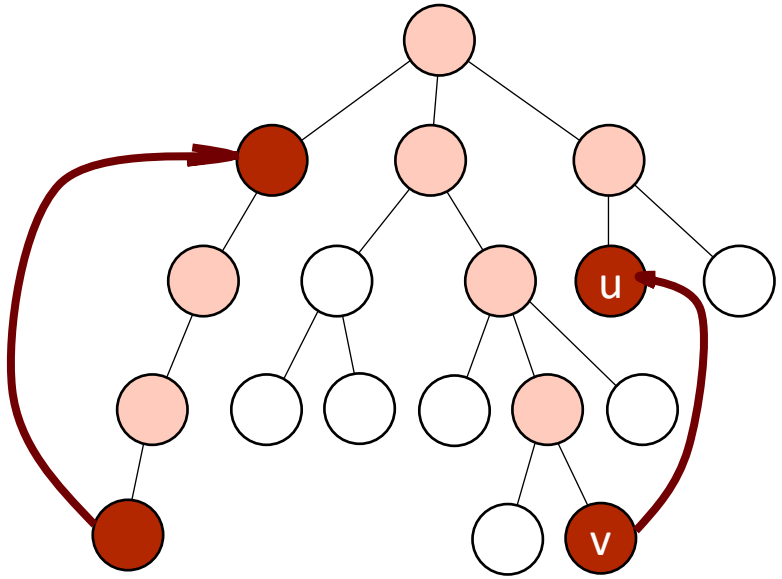
사이클 형성

두 번째로 작은 스패닝 트리



MST와의 격차 최소화하기 위해
$path(u, v)$의 최대값과 교체

두 번째로 작은 스패닝 트리



MST를 만들고 남은 간선 $m - N + 1$개에 대해
각 간선을 추가하면서 만든 MST를 $logN$에 연산 가능
$->$ 전체 $O(mlogN)$

마무리

- 트리구조에서 해결할 수 있는 여러 유형들을 살펴봤습니다.
  - LCA query
  - DP on Tree
  - Complete Binary Tree
  - Tree-Like Graph

- HLD, Centroid, Splay, ...

연습문제

- 11438 LCA2
- 3176 도로 네트워크
- 20295 사탕 배달
- 20931 혹 떼러 갔다 혹 붙여 온다
- 2533 사회망 서비스
- 1949 우수 마을
- 7812 중앙 트리
- 18152 Radio Prize
- 15909 토르의 여행
- 17261 석유가 넘쳐흘러
- 20498 C = 15
- 2861 원섭동 사람들
- 17790 Inquiry II
- 1626 두 번째로 작은 스패닝트리
- http://codeforces.com/contest/1454/problem/E

- 스스로 못 풀어내도 괜찮습니다.
- 어떻게든 전부 AC를 받고 **이해하는 것**이 목표

Q&A