제5회 전국 대학생 프로그래밍 동아리 연합 여름 대회 풀이

풀이입니다.

통계

- A - Dr. L's exam
  - 출제자: 정현환
  - First solve: DPneuDP @서울대 , 3분
  - 정답률: 68/87, 78.161%
- B – 뱀
  - 출제자: 이태현
  - First solve: Mola Mola @서울대 , 112분
  - 정답률: 12/141, 8.511%
- C – 사전 조사
  - 출제자: 박수찬
  - First solve: DPneuDP @서울대 , 226분
  - 정답률: 3/35 8.571%

통계

- D – Bulb
  - 출제자: 윤지학
  - First solve: Mola Mola @서울대 , 231분
  - 정답률: 3/8, 37.500%
- E - How many binary sequences?
  - 출제자: 정현환
  - First solve: gcc @서울대 , 94분
  - 정답률: 11/41, 26.829%
- F - Speak your mind!
  - 출제자: 조승현 (bryan)
  - First solve: NonsanWarrior @서울대 , 31분
  - 정답률: 22/103, 21.359%

통계

- G – 내가 어디를 거쳐갔더라?
  - 출제자: 조승현 (ainta)
  - 정답률: 0/0, 0.000%
- H - 프로도의 선물 포장
  - 출제자: 김선영
  - First solve: LifeIsNotThatEasy @서울대 , 24분
  - 정답률: 36/107, 33.645%
- I – International Meeting
  - 출제자: 김선영
  - First solve: Jungrae10002gija @한양대 , 7분
  - 정답률: 55/255, 21.569%

통계

- J – 분할
  - 출제자: 김경근
  - First solve: gcc @서울대 , 140분
  - 정답률: 7/61, 11.475%
- K - The festival must go on
  - 출제자: 조승현 (ainta)
  - First solve: Mola Mola @서울대 , 242분
  - 정답률: 2/8, 25.000%
- L - 수열의 장인
  - 출제자: 박수찬
  - First solve: Accepted @고려대 , 23분
  - 정답률: 25/317, 7.886%

A. Dr. L’s exam

j	1	2	3	4	5	6	7	8	9	10
$(j-1)\ \mathbf{mod}\ 5$	1	2	3	4	5	1	2	3	4	5

A. Dr. L's exam

j	1	2	3	4	5	6	7	8	9	10
$(j-1) \bmod 5$	1	2	3	4	5	1	2	3	4	5

- 제출한 답안이 "1 2 3 4 5 1 2 3 4 5"이면 재시험 대상

A. Dr. L's exam

j	1	2	3	4	5	6	7	8	9	10
$(j-1)\ \mathbf{mod}\ \mathbf{5}$	1	2	3	4	5	1	2	3	4	5

- 제출한 답안이 "1 2 3 4 5 1 2 3 4 5"이면 재시험 대상
- 한 줄을 통째로 받아서 문자열 비교하면 편합니다

B. 뱀

- 뱀의 경로는 이어지는 선분들이라 생각 가능

B. 뱀

- 뱀의 경로는 이어지는 선분들이라 생각 가능
- 수평 혹은 수직으로 이루어진 선분들 사이의 교차 문제

B. 뱀

- 뱀의 경로는 이어지는 선분들이라 생각 가능
- 수평 혹은 수직으로 이루어진 선분들 사이의 교차 문제
- 어느 교점까지 몇 초가 걸릴 지 계산하자

B. 뱀

- 뱀의 경로는 이어지는 선분들이라 생각 가능
- 수평 혹은 수직으로 이루어진 선분들 사이의 교차 문제
- 어느 교점까지 몇 초가 걸릴 지 계산하자
  - 그 선분 전까지의 $\sum t[i]$ + 교점을 구하고 선분의 시작점부터 거리

B. 뱀

- 뱀의 경로는 이어지는 선분들이라 생각 가능
- 수평 혹은 수직으로 이루어진 선분들 사이의 교차 문제
- 어느 교점까지 몇 초가 걸릴 지 계산하자
  - 그 선분 전까지의 $\sum t[i]$ + 교점을 구하고 선분의 시작점부터 거리
- $N \leq 1{,}000$이므로 $O(N^2)$에 해결 가능

B. 뱀

- 뱀의 경로는 이어지는 선분들이라 생각 가능
- 수평 혹은 수직으로 이루어진 선분들 사이의 교차 문제
- 어느 교점까지 몇 초가 걸릴 지 계산하자
  - 그 선분 전까지의 $\sum t[i]$ + 교점을 구하고 선분의 시작점부터 거리
- $N \leq 1{,}000$이므로 $O(N^2)$에 해결 가능
- $L \leq 10^8 \rightarrow$ 지나온 격자들을 다 저장하면 Memory Limit Exceeded

C. 사전 조사

- A는 언제 찾을 수 있을까?

C. 사전 조사

- A는 언제 찾을 수 있을까?

$$\cdots, A, \cdots$$

C. 사전 조사

- A는 언제 찾을 수 있을까?

$$\cdots, A, \cdots$$

A를 읽기 전에
얼마인지 알 수 없다

C. 사전 조사

- A는 언제 찾을 수 있을까?

$$\cdots, A, \cdots, \boxed{\cancel{(A-1)}}, \cdots$$

A를 읽기 전에
얼마인지 알 수 없다

$A-1$이 나와야 할 자리에
안 나왔다면, 알 수 있다

C. 사전 조사

- A는 언제 찾을 수 있을까?

$$\cdots, A, \cdots, (\cancel{A-1}), \cdots$$

A를 읽기 전에
얼마인지 알 수 없다

사전에서
A가 나타난 이후

$A-1$이 나와야 할 자리에
안 나왔다면, 알 수 있다

사전에서
$A-1$이 있어야 할 곳을
지난 이후

C. 사전 조사

- B는 언제 찾을 수 있을까?

C. 사전 조사

- B는 언제 찾을 수 있을까?

$$\cdots, B, \cdots, \boxed{\cancel{(B+1)}}, \cdots$$

B를 읽기 전에
얼마인지 알 수 없다

사전에서
B가 나타난 이후

$B+1$이 나와야 할 자리에
안 나왔다면, 알 수 있다

사전에서
$B+1$이 있어야 할 곳을
지난 이후

C. 사전 조사

- A는 언제 찾을 수 있을까?

$\cdots, A, \cdots, \cancel{(A-1)}, \cdots$

A를 읽기 전에 얼마인지 알 수 없다

사전에서 A가 나타난 이후

$A-1$이 나와야 할 자리에 안 나왔다면, 알 수 있다

사전에서 $A-1$이 있어야 할 곳을 지난 이후

- B는 언제 찾을 수 있을까?

$\cdots, B, \cdots, \cancel{(B+1)}, \cdots$

B를 읽기 전에 얼마인지 알 수 없다

사전에서 B가 나타난 이후

$B+1$이 나와야 할 자리에 안 나왔다면, 알 수 있다

사전에서 $B+1$이 있어야 할 곳을 지난 이후

C. 사전 조사

- A는 언제 찾을 수 있을까?

$$\cdots, A, \cdots, (\cancel{A-1}), \cdots$$

A를 읽기 전에
얼마인지 알 수 없다

<u>사전에서</u>
<u>A가 나타난 이후</u>

$A-1$이 나와야 할 자리에
안 나왔다면, 알 수 있다

<u>사전에서</u>
<u>$A-1$이 있어야 할 곳을</u>
<u>지난 이후</u>

- B는 언제 찾을 수 있을까?

$$\cdots, B, \cdots, (\cancel{B+1}), \cdots$$

B를 읽기 전에
얼마인지 알 수 없다

<u>사전에서</u>
<u>B가 나타난 이후</u>

$B+1$이 나와야 할 자리에
안 나왔다면, 알 수 있다

<u>사전에서</u>
<u>$B+1$이 있어야 할 곳을</u>
<u>지난 이후</u>

C. 사전 조사

- 결론: 사전에서 $A, A-1, B, B+1$이 있어야 할 위치를 모두 지난 순간 수찬이는 A, B를 알아낼 수 있다.

C. 사전 조사

- 결론: 사전에서 $A, A-1, B, B+1$이 있어야 할 위치를 모두 지난 순간 수찬이는 A, B를 알아낼 수 있다.
- 바뀐 문제: A 이상 B 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?

C. 사전 조사

- 결론: 사전에서 $A, A-1, B, B+1$이 있어야 할 위치를 모두 지난 순간 수찬이는 A, B를 알아낼 수 있다.
- 바뀐 문제: A 이상 B 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.
- $X = 5167$이라면?

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.
- $X = 5167$이라면?
  - 1, 2, 3, 4로 시작하는 모든 수: X보다 사전에서 앞에 있다.

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.
- $X = 5167$이라면?
  - 1, 2, 3, 4로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 50으로 시작하는 모든 수: X보다 사전에서 앞에 있다.

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.
- $X = 5167$이라면?
  - 1, 2, 3, 4로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 50으로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 510, 511, …, 515로 시작하는 모든 수: X보다 사전에서 앞에 있다.

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.
- $X = 5167$이라면?
  - 1, 2, 3, 4로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 50으로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 510, 511, …, 515로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 5160, 5161, …, 5166로 시작하는 모든 수: X보다 사전에서 앞에 있다.

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 설명하기 어려우니 예를 들어봅시다.
- $X = 5167$이라면?
  - 1, 2, 3, 4로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 50으로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 510, 511, …, 515로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 5160, 5161, …, 5166로 시작하는 모든 수: X보다 사전에서 앞에 있다.
  - 5167: X이다.

C. 사전 조사

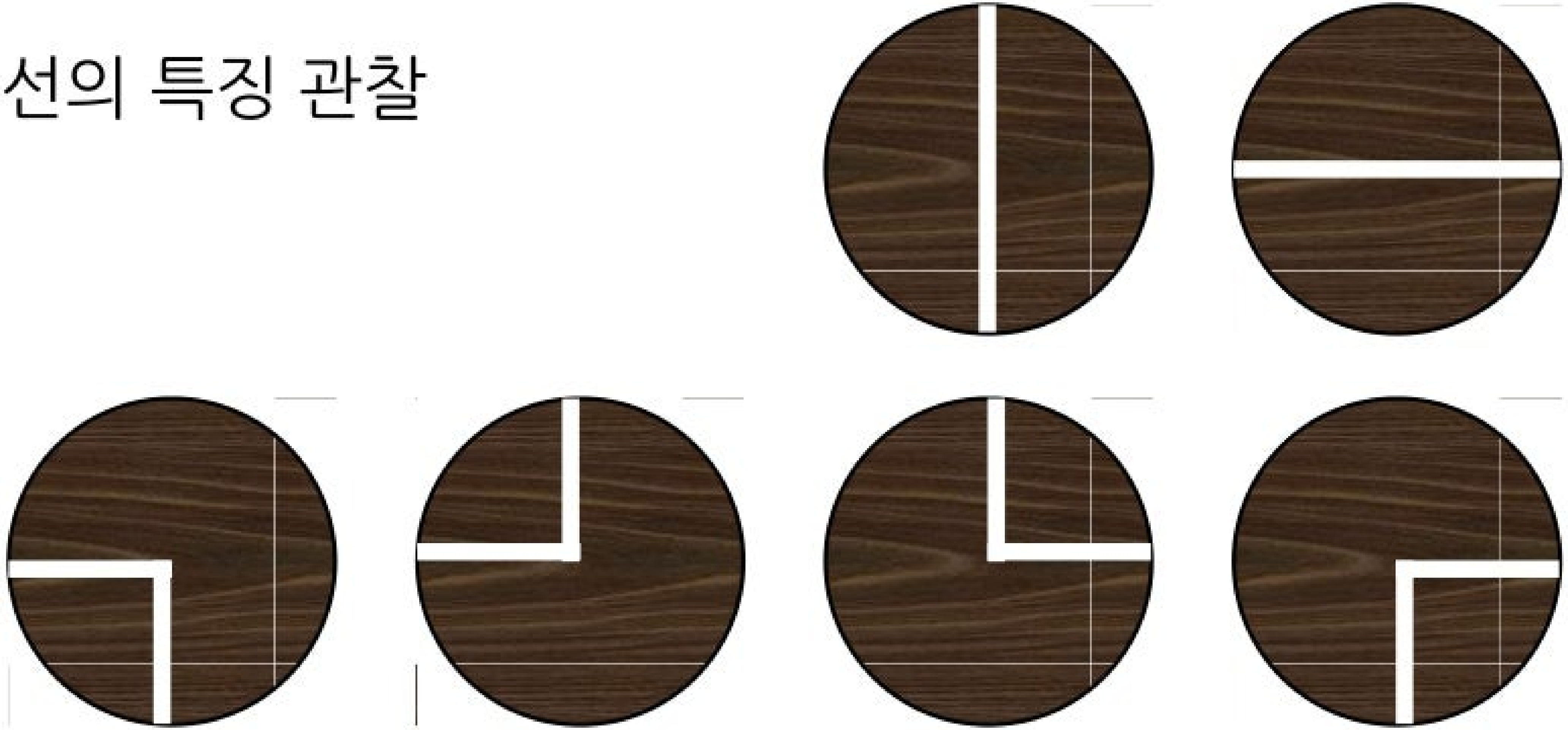
- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 이런 식으로 접두사가 p인 자연수들 중 1 이상 n 이하인 것의 개수를 세어 주면 됩니다.

C. 사전 조사

- 더 간단히: 1 이상 n 이하의 자연수 중 사전순으로 X보다 작거나 같은 것의 개수는?
- 이런 식으로 접두사가 p인 자연수들 중 1 이상 n 이하인 것의 개수를 세어 주면 됩니다.
- 한 테스트 케이스당 $O(\log_{10}^2 n)$에 해결 가능합니다.

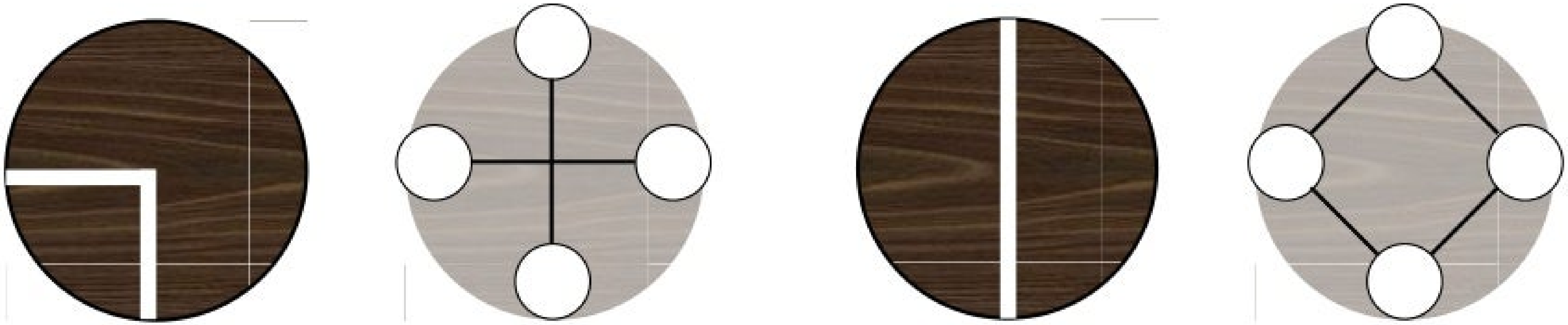
D. Bulb

- 전선의 특징 관찰



D. Bulb

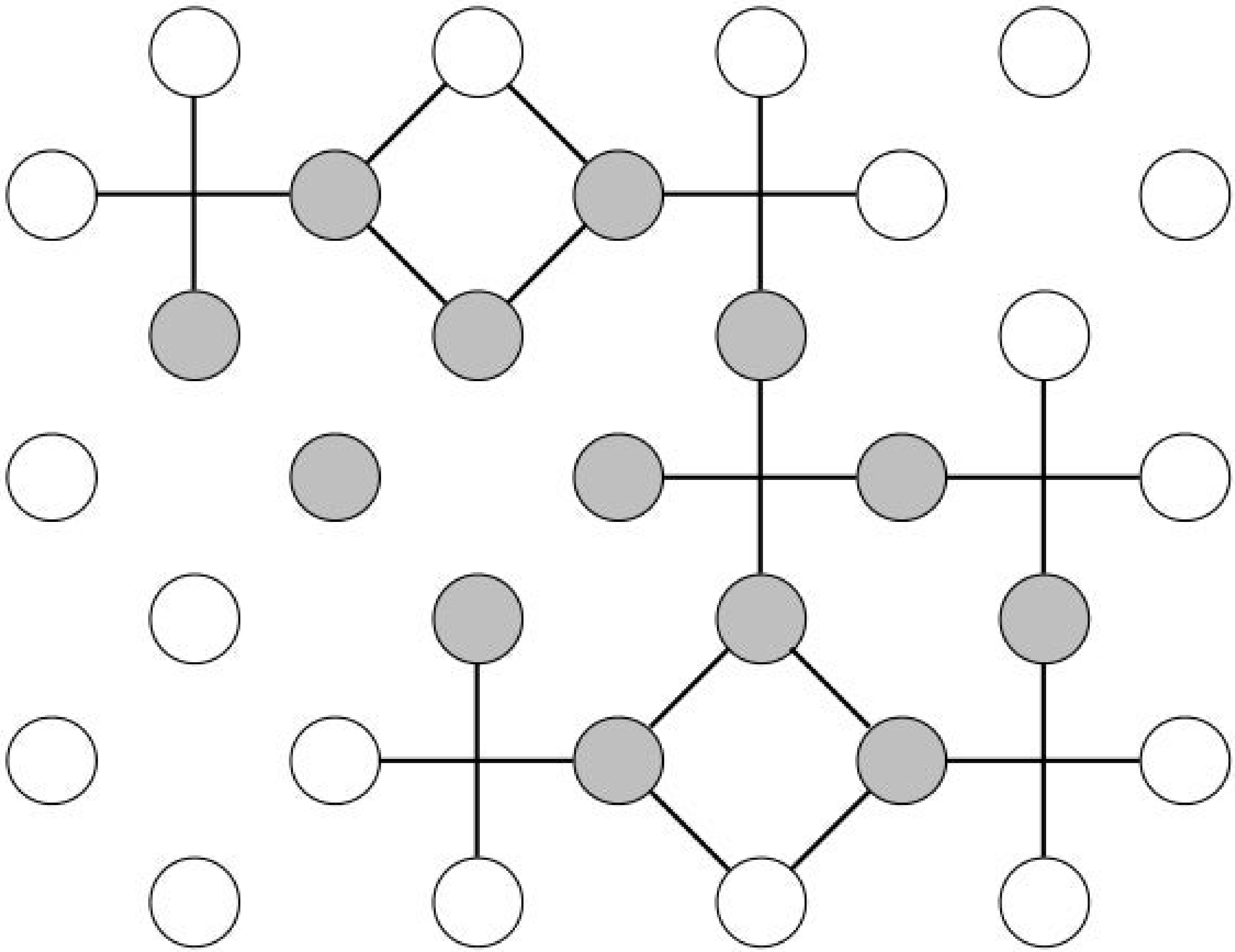
- 격자선마다 정점을 만들고 간선을 이어준다
- 색을 칠해준다



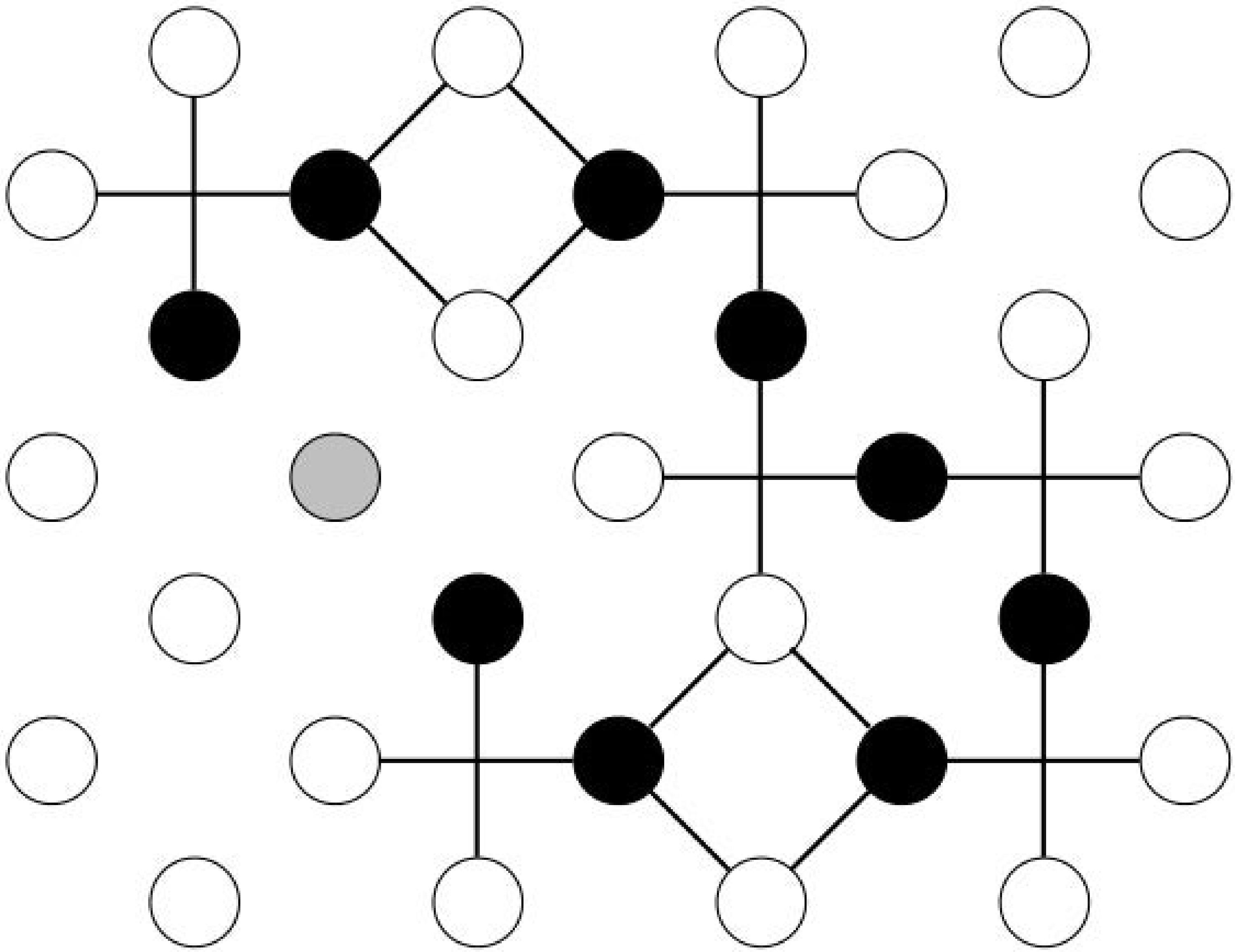
D. Bulb

q	l	q	*
O	O	q	q
*	q	l	q

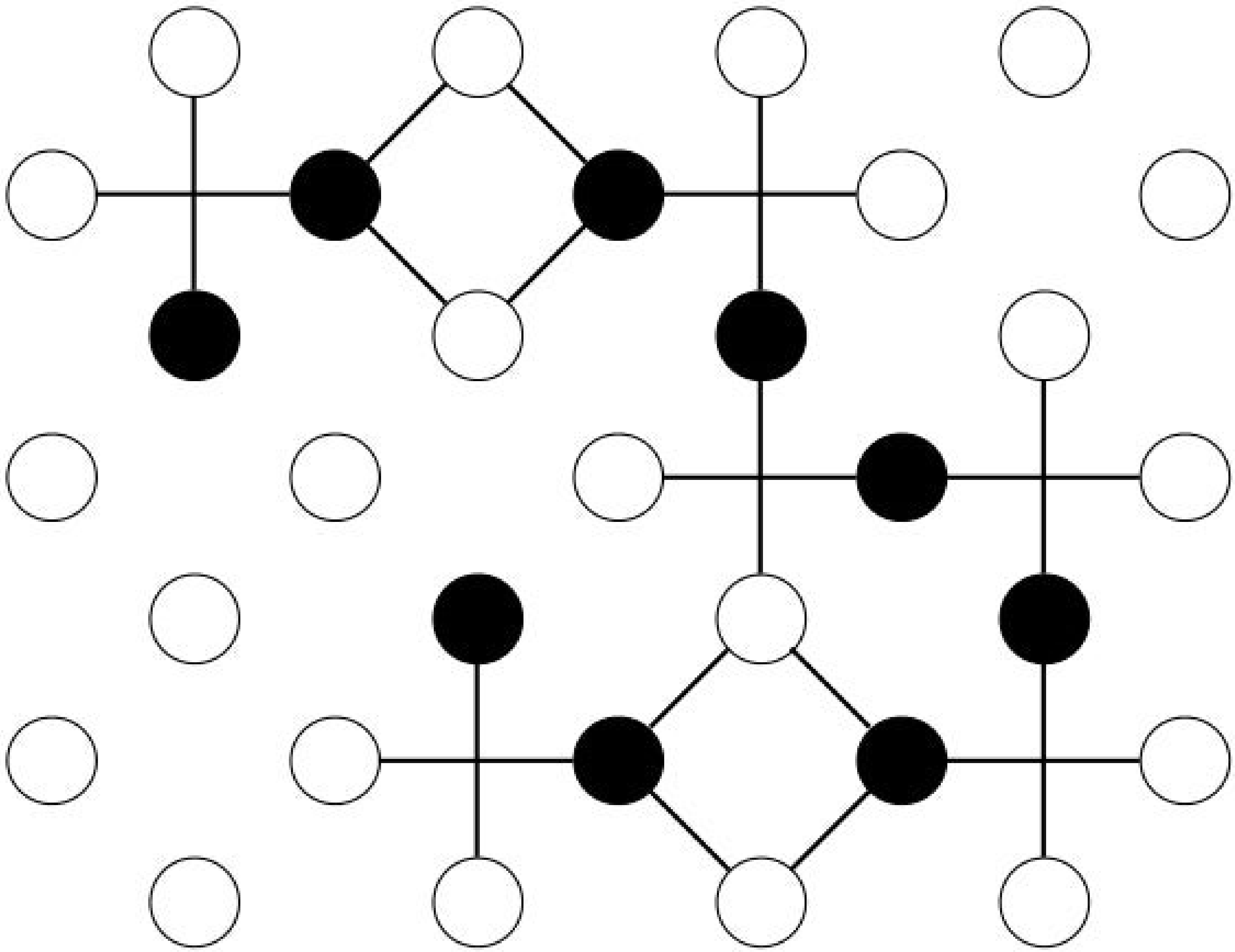
D. Bulb



D. Bulb



D. Bulb



E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

Z	2	2	1	0	2	1	2
X	1	1	0/1	0	1	0/1	1
Y	1	1	1/0	0	1	1/0	1

E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

$\boldsymbol{Z}$	2	2	1	0	2	1	2
$\boldsymbol{X}$	1	1	0/1	0	1	0/1	1
$\boldsymbol{Y}$	1	1	1/0	0	1	1/0	1

- 각 Z_i마다 X, Y의 후보가 4개 생긴다

E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

$\boldsymbol{Z}$	2	2	1	0	2	1	2
$\boldsymbol{X}$	1	1	0/1	0	1	0/1	1
$\boldsymbol{Y}$	1	1	1/0	0	1	1/0	1

- 각 Z_i마다 X, Y의 후보가 ~~4개~~ 생긴다
  - 사실 2개

E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

Z	2	2	1	0	2	1	2
X	1	1	0/1	0	1	0/1	1
Y	1	1	1/0	0	1	1/0	1

- 각 Z_i마다 X, Y의 후보가 ~~4개~~ 생긴다
  - 사실 2개
  - X, Y는 순서가 바뀌어도 상관 없으므로

E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

$\boldsymbol{Z}$	2	2	1	0	2	1	2
$\boldsymbol{X}$	1	1	0/1	0	1	0/1	1
$\boldsymbol{Y}$	1	1	1/0	0	1	1/0	1

- 각 Z_i마다 X, Y의 후보가 ~~4개~~ 생긴다
  - 사실 2개
  - X, Y는 순서가 바뀌어도 상관 없으므로
- 그래서 모든 2^N개의 경우를 순회하며 답을 찾으면 된다.

E. How many binary sequences?

- 참고: 원래 문제는 굉장히 어려웠습니다.

Z	2	2	1	0	2	1	2
X	1	1	0/1	0	1	0/1	1
Y	1	1	1/0	0	1	1/0	1

- 각 Z_i마다 X, Y의 후보가 ~~4개~~ 생긴다
  - 사실 2개
  - X, Y는 순서가 바뀌어도 상관 없으므로
- 그래서 모든 2^N개의 경우를 순회하며 답을 찾으면 된다.
  - 유의: 문자열 등으로 비교하여 $O(2^N K)$로 풀면 제한 시간 초과

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다
  - $\pm 2^k$의 경우 답은 $k + 1$

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다
  - $\pm 2^k$의 경우 답은 $k + 1$
  - m을 p개의 단어로 표현할 수 있다면 $m \pm 2^l$의 답은 $p + l + 5$

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다
  - $\pm 2^k$의 경우 답은 $k+1$
  - m을 p개의 단어로 표현할 수 있다면 $m \pm 2^l$의 답은 $p+l+5$
  - Dijkstra 비슷하게 하면 답을 구할 수 있음. 근데 다들 DP로 풀더라고요..

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다
  - $\pm 2^k$의 경우 답은 $k+1$
  - m을 p개의 단어로 표현할 수 있다면 $m \pm 2^l$의 답은 $p+l+5$
  - Dijkstra 비슷하게 하면 답을 구할 수 있음. 근데 다들 DP로 풀더라고요..
  - 0을 표현하는 데 필요한 단어 수는 6입니다.

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다
  - $\pm 2^k$의 경우 답은 $k + 1$
  - m을 p개의 단어로 표현할 수 있다면 $m \pm 2^l$의 답은 $p + l + 5$
  - Dijkstra 비슷하게 하면 답을 구할 수 있음. 근데 다들 DP로 풀더라고요..
  - 0을 표현하는 데 필요한 단어 수는 6입니다.
- 입력받은 숫자마다 그에 대한 답을 출력합니다.

F. Speak your mind!

- 먼저 입력 범위의 모든 정수에 대해 답을 구합니다
  - $\pm 2^k$의 경우 답은 $k + 1$
  - m을 p개의 단어로 표현할 수 있다면 $m \pm 2^l$의 답은 $p + l + 5$
  - Dijkstra 비슷하게 하면 답을 구할 수 있음. 근데 다들 DP로 풀더라고요..
  - 0을 표현하는 데 필요한 단어 수는 6입니다.
- 입력받은 숫자마다 그에 대한 답을 출력합니다.
- Shakespeare Programming Language는 좋은 프로그래밍 언어입니다.

G. 내가 어디를 거쳐갔더라?

- DFS traversal을 통해 간선을 tree edge와 back edge로 구분 (DFS tree 생성)

G. 내가 어디를 거쳐갔더라?

- DFS traversal을 통해 간선을 tree edge와 back edge로 구분 (DFS tree 생성)

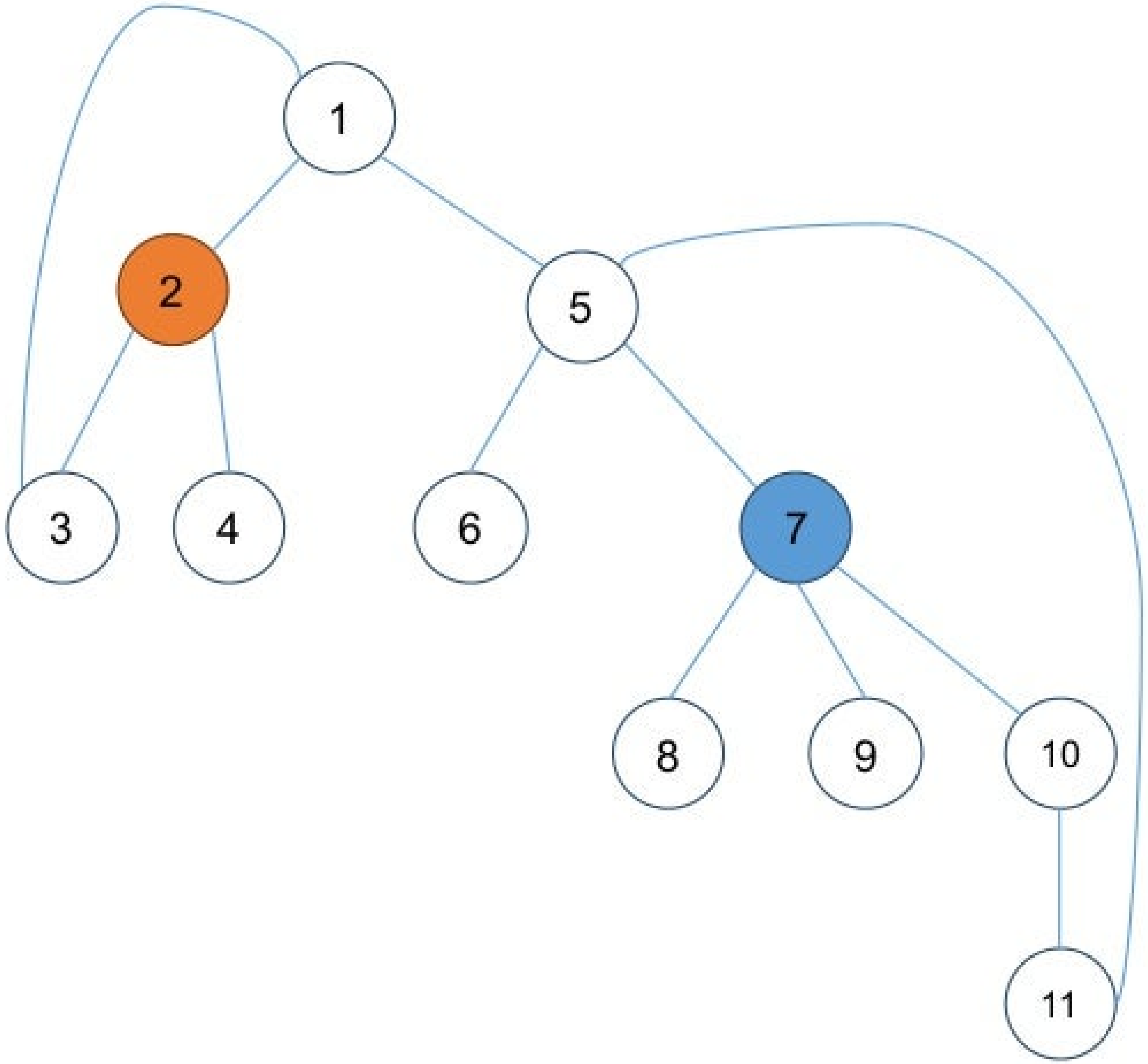
- 각 노드를 root로 하는 subtree를 구간으로 표현 가능

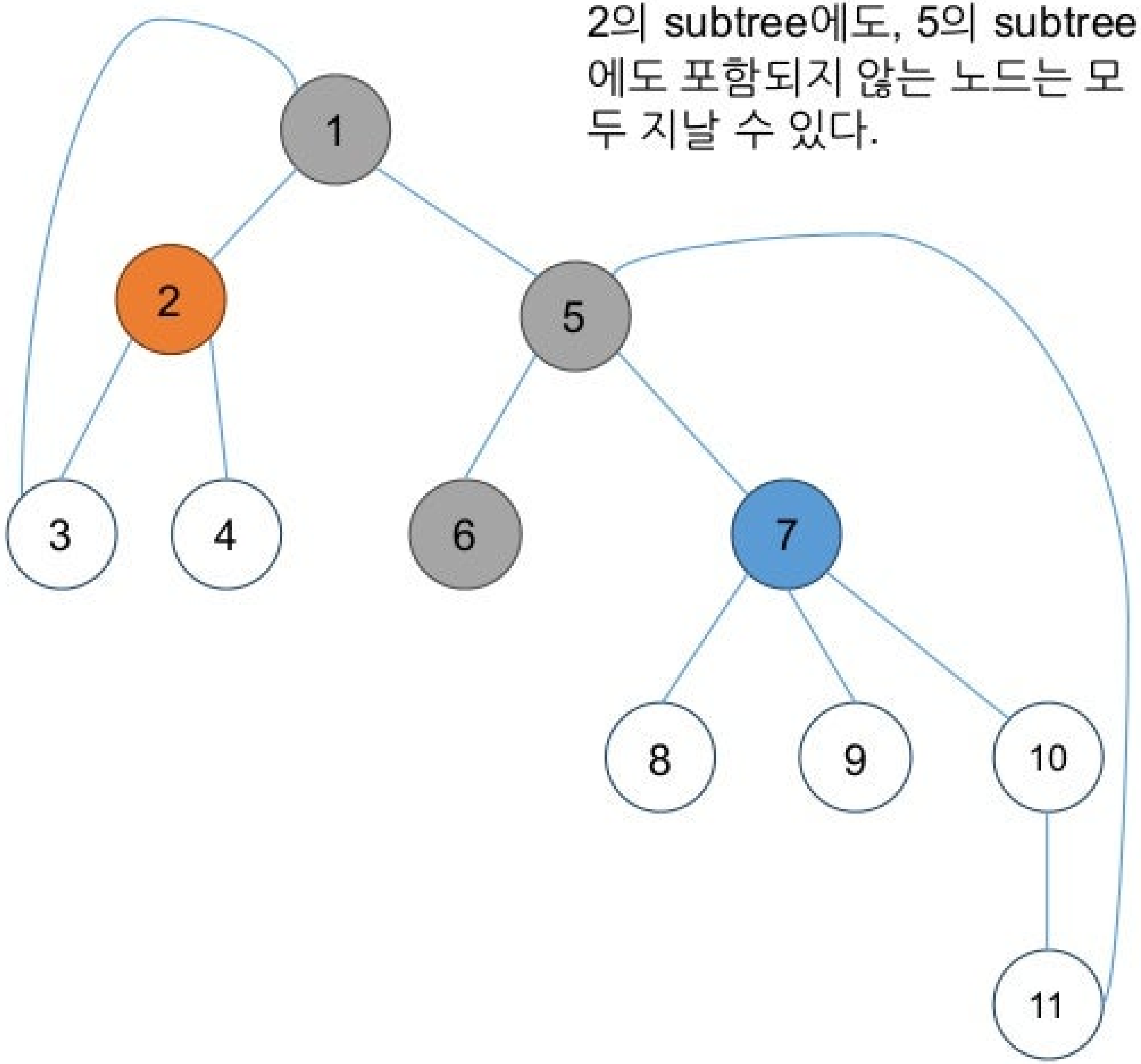
G. 내가 어디를 거쳐갔더라?

- DFS traversal을 통해 간선을 tree edge와 back edge로 구분 (DFS tree 생성)

- 각 노드를 root로 하는 subtree를 구간으로 표현 가능

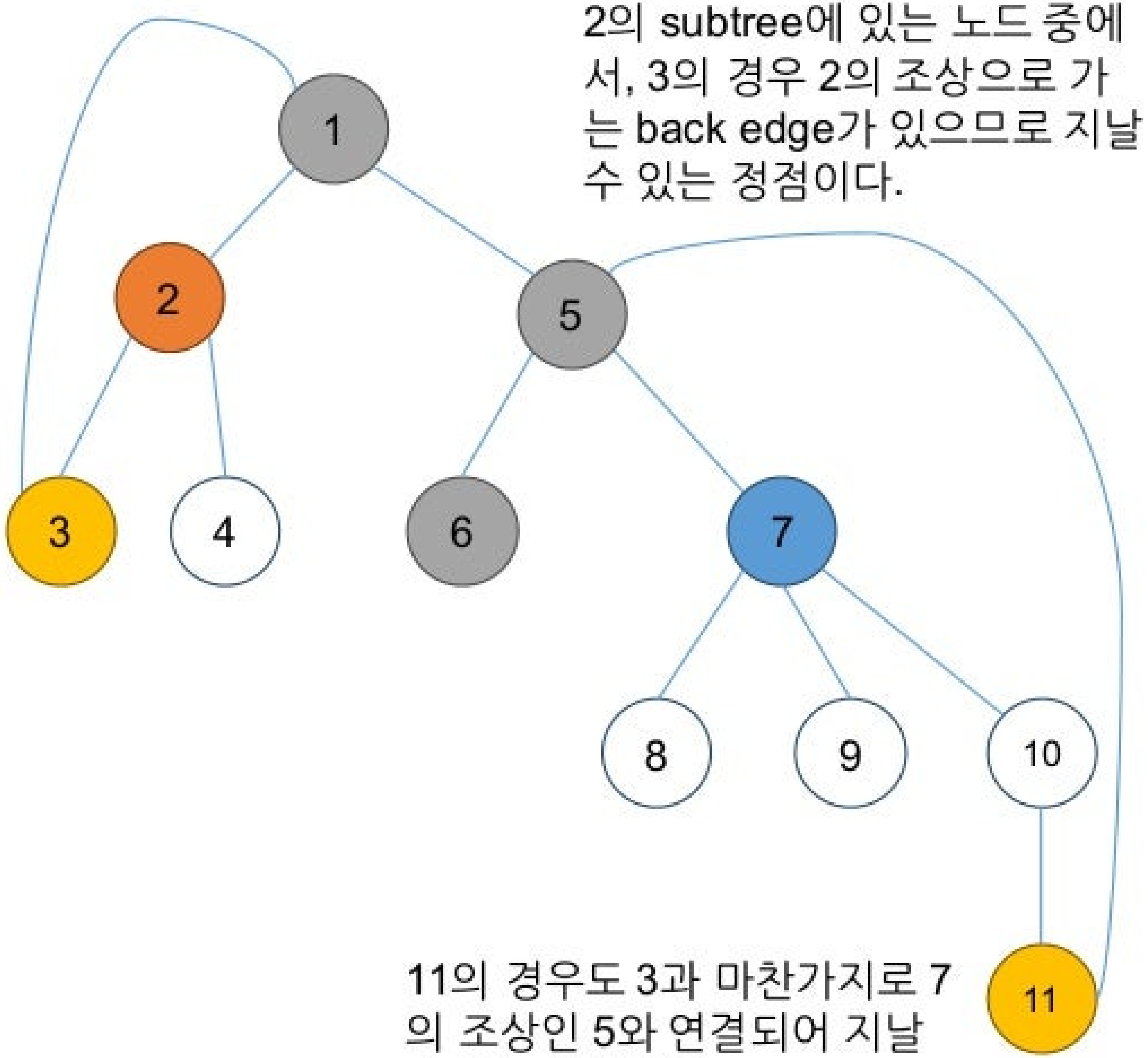
- 이를 통해 승현이네 집 u와 민수네 집 v가 주어졌을 때, u와 v가 DFS tree에서 서로 조상-자손 관계인지 아닌지 판단 가능


1
2
5
3
4
6
7
8
9
10
11

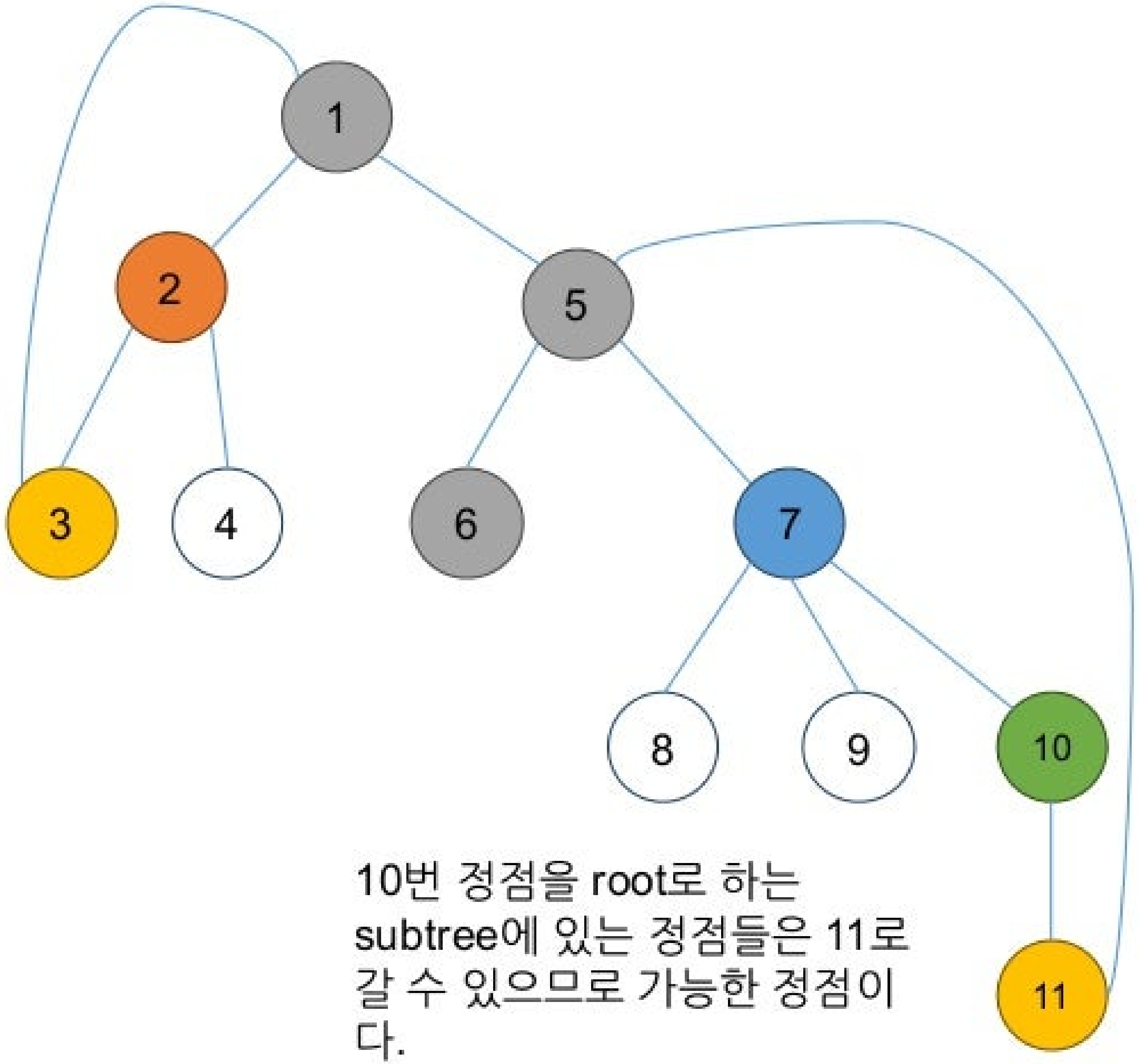


2의 subtree에도, 5의 subtree
에도 포함되지 않는 노드는 모
두 지날 수 있다.

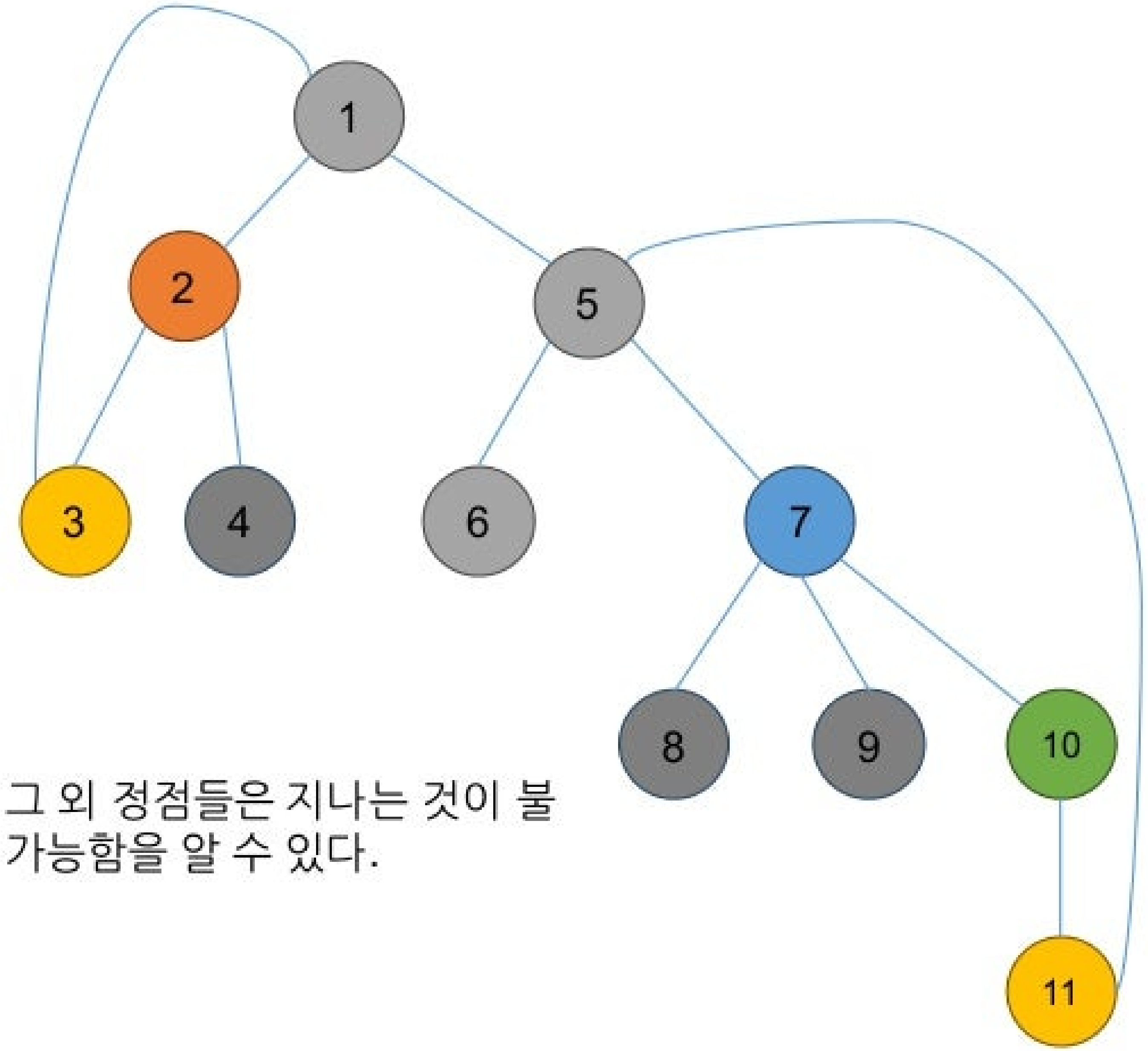
2의 subtree에 있는 노드 중에서, 3의 경우 2의 조상으로 가는 back edge가 있으므로 지날 수 있는 정점이다.



11의 경우도 3과 마찬가지로 7의 조상인 5와 연결되어 지날 수 있는 정점이다.



10번 정점을 root로 하는
subtree에 있는 정점들은 11로
갈 수 있으므로 가능한 정점이
다.



그 외 정점들은 지나는 것이 불
가능함을 알 수 있다.

G. 내가 어디를 거쳐갔더라?

- 각 정점에 대해, 자신을 거치지 않고 자신의 조상 노드로 갈 수 없는 정점 개수를 저장한다. (u번 정점에 대한 이 값 : Cnt[u])

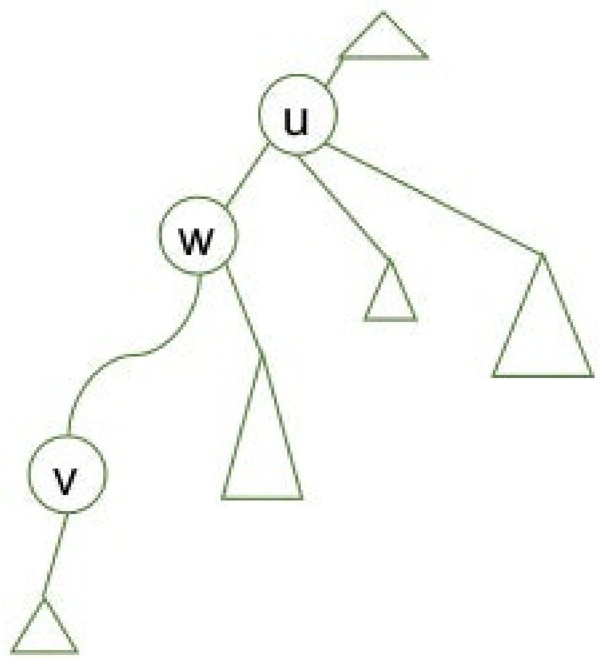
- 이는 DFS traversal을 하면서 같이 구할 수 있다. (절선 구하기)

G. 내가 어디를 거쳐갔더라?

- 1. u와 v가 서로 조상 - 자손 관계가 아닐 경우
  - u의 subtree에도 포함되지 않고, v의 subtree에도 포함되지 않는 정점은 방문 가능
  - N - Cnt[u] + Cnt[v]가 답

G. 내가 어디를 거쳐갔더라?

- 2. u가 v의 조상인 경우
  - v의 subtree 중에서는 Cnt[v]개만큼 방문 불가능

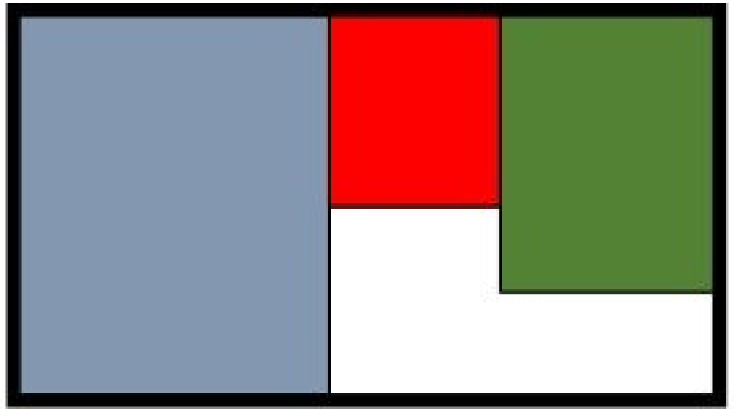


H. 프로도의 선물 포장

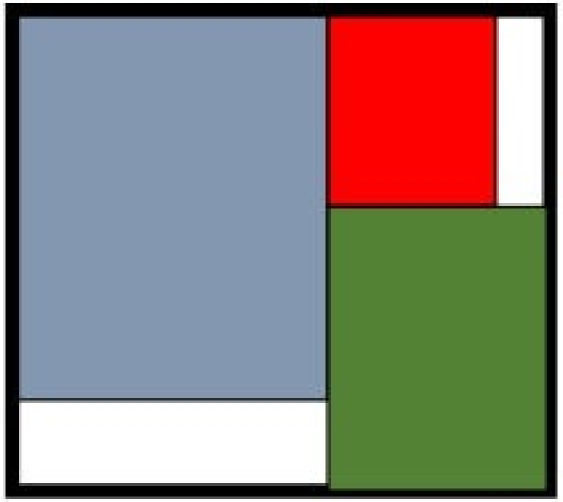
- 선물의 배치 방법은 크게 두 가지

H. 프로도의 선물 포장

- 선물의 배치 방법은 크게 두 가지



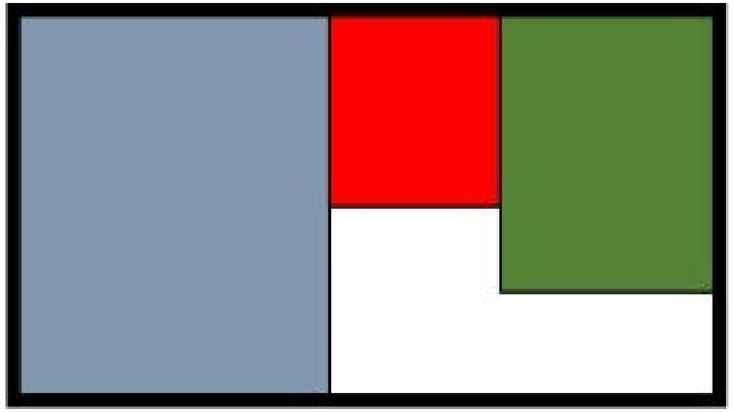
1. 세 선물 상자를 일렬로 놓기



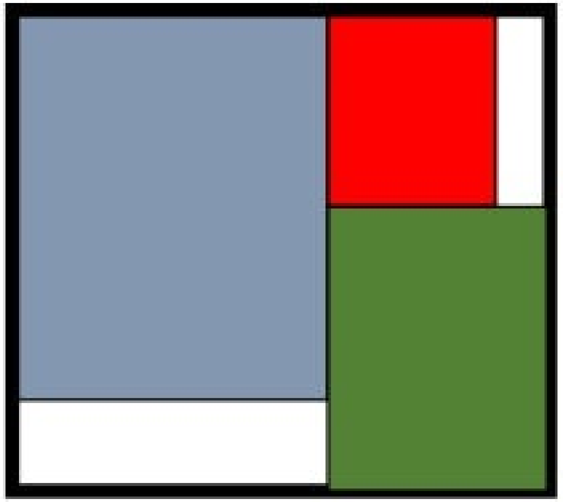
2. 한 선물상자 | 다른 두 선물 상자

H. 프로도의 선물 포장

- 선물의 배치 방법은 크게 두 가지



1. 세 선물 상자를 일렬로 놓기

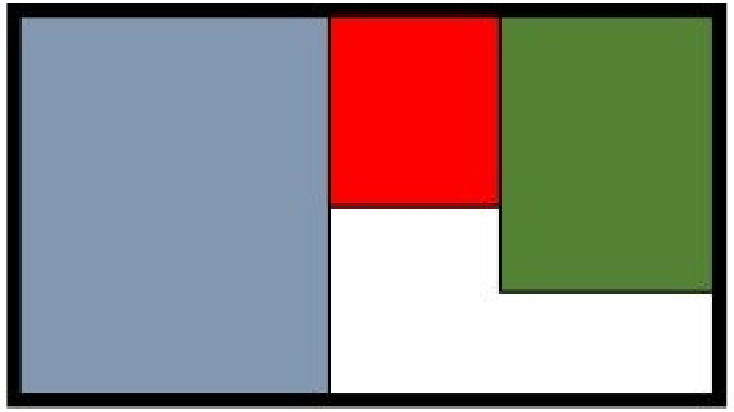


2. 한 선물상자 | 다른 두 선물 상자

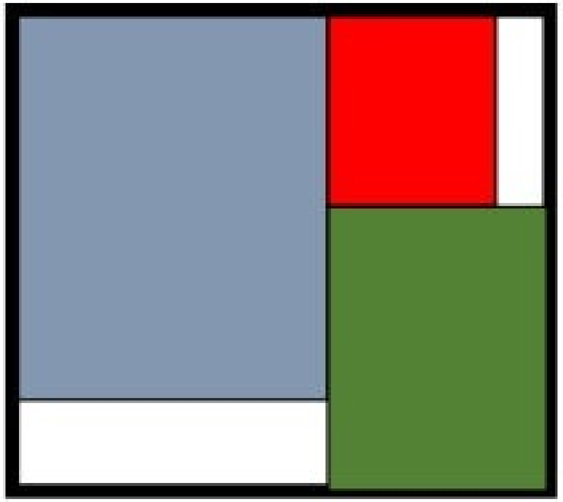
- 가능한 모든 방법을 다 돌려봅시다.

H. 프로도의 선물 포장

- 선물의 배치 방법은 크게 두 가지



1. 세 선물 상자를 일렬로 놓기



2. 한 선물상자 | 다른 두 선물 상자

- 가능한 모든 방법을 다 돌려봅시다.
  - 90도 회전 2^3가지, 선물 순서 3!가지를 고려해야 합니다

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?
  - UTC+0에서 $n - x$시

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?
  - UTC+0에서 $n - x$시
  - UTC+y에서 $n - x + y$시

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?
  - UTC+0에서 $n - x$시
  - UTC+y에서 $n - x + y$시
- 따라서 $n - x + y$를 구해서 잘 출력합니다.

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?
  - UTC+0에서 $n - x$시
  - UTC+y에서 $n - x + y$시
- 따라서 $n - x + y$를 구해서 잘 출력합니다.
- 하지만 우리에게는 0.5가 남았다!

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?
  - UTC+0에서 $n - x$시
  - UTC+y에서 $n - x + y$시
- 따라서 $n - x + y$를 구해서 잘 출력합니다.
- 하지만 우리에게는 0.5가 남았다!
  - 0.5시 = 30분

I. International meeting

- UTC+x에서 현재 n시라면, UTC+y에서는 몇 시일까?
  - UTC+0에서 $n - x$시
  - UTC+y에서 $n - x + y$시
- 따라서 $n - x + y$를 구해서 잘 출력합니다.
- 하지만 우리에게는 0.5가 남았다!
  - 0.5시 = 30분
  - 역시 잘 출력합니다.

J. 분할

- 한 차원만 생각하면

Depth 0 : 14

J. 분할

- 한 차원만 생각하면

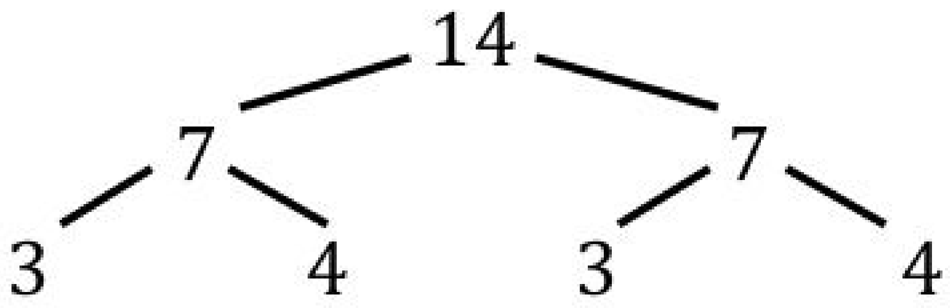
Depth 0 : 14

Depth 1 : 7 7

J. 분할

- 한 차원만 생각하면

Depth 0 :
Depth 1 :
Depth 2 :



J. 분할

- 한 차원만 생각하면

Depth 0 :				14			
Depth 1 :		7				7	
Depth 2 :	3		3		4		4

J. 분할

- 한 차원만 생각하면

Depth 0 : 14

Depth 1 : $7 \quad 7$

Depth 2 : $3 \quad 3 \quad 4 \quad 4$

Depth 3 : $1 \quad 2 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2$

J. 분할

- 한 차원만 생각하면

Depth 0 : 14

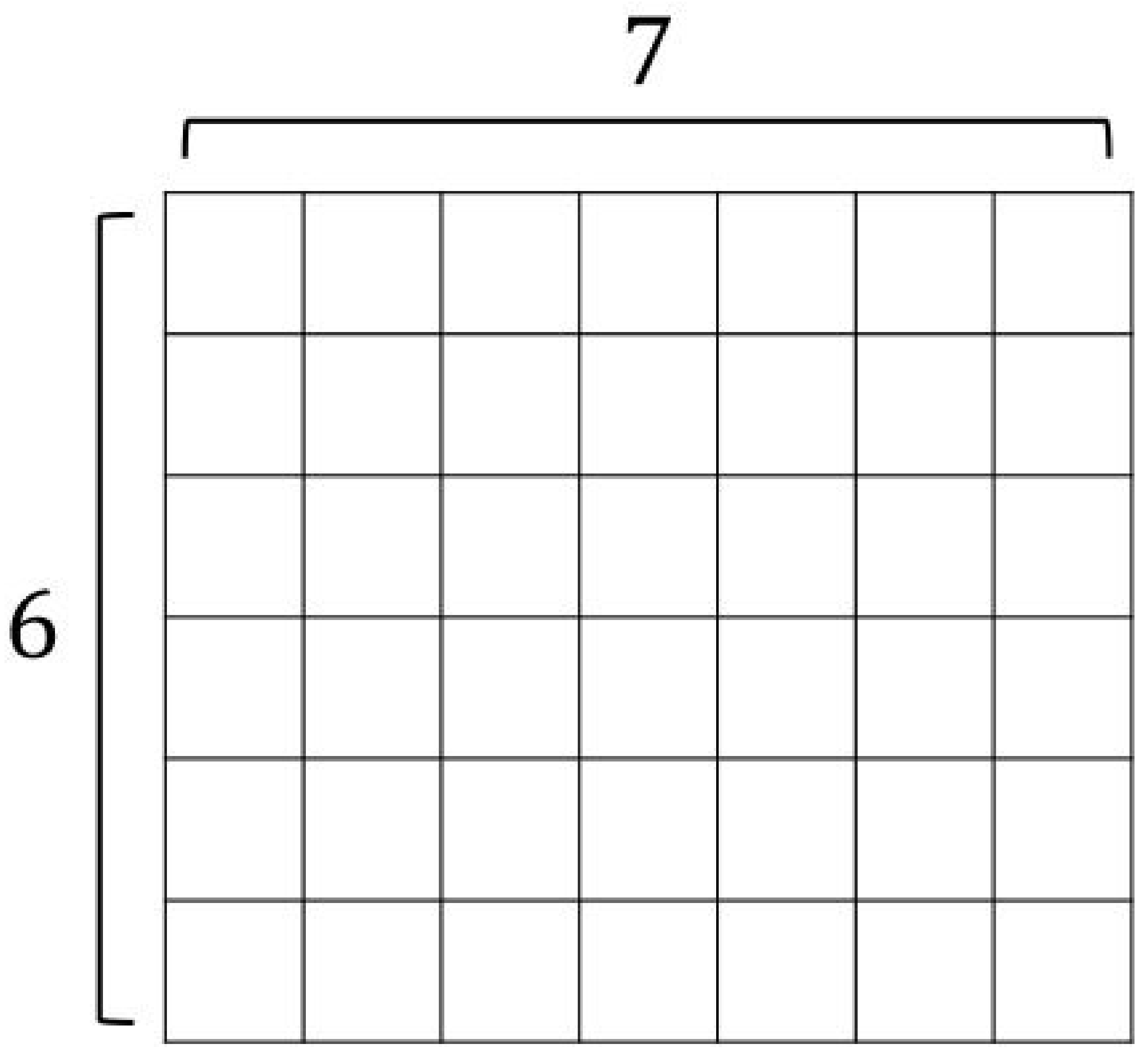
Depth 1 : 7 7

Depth 2 : 3 3 4 4

Depth 3 : 1 1 2 2 2 2 2 2

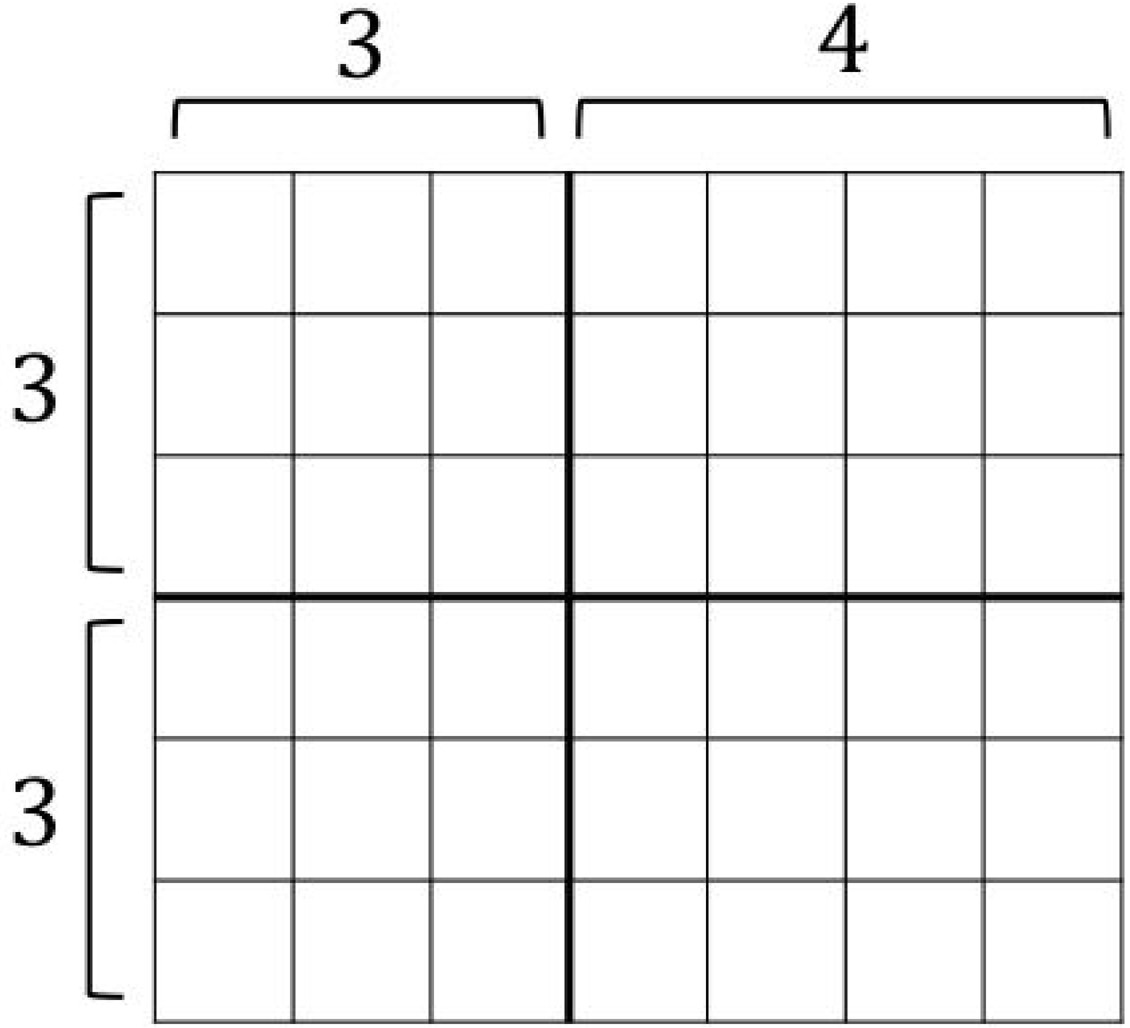
J. 분할

Depth 0



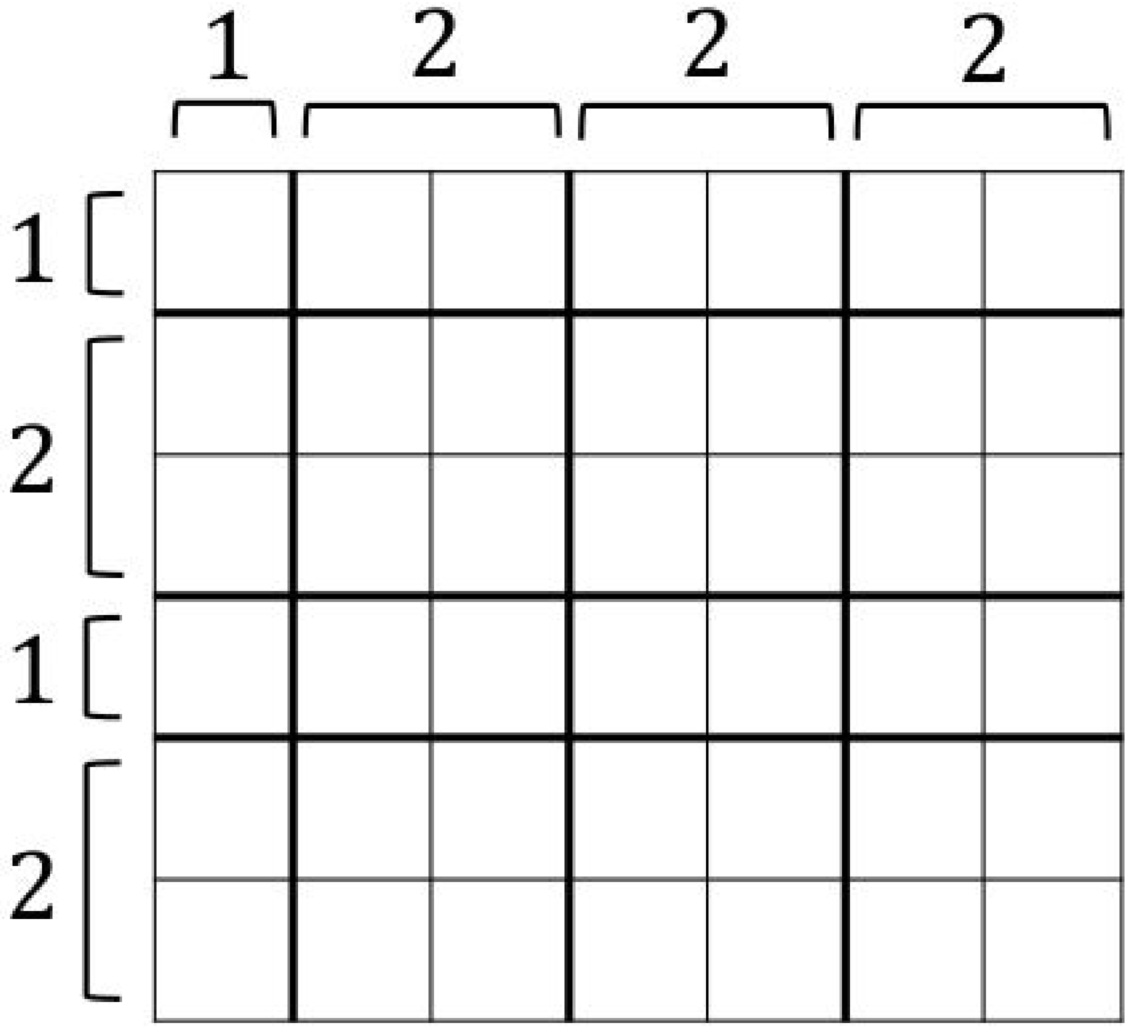
J. 분할

Depth 1



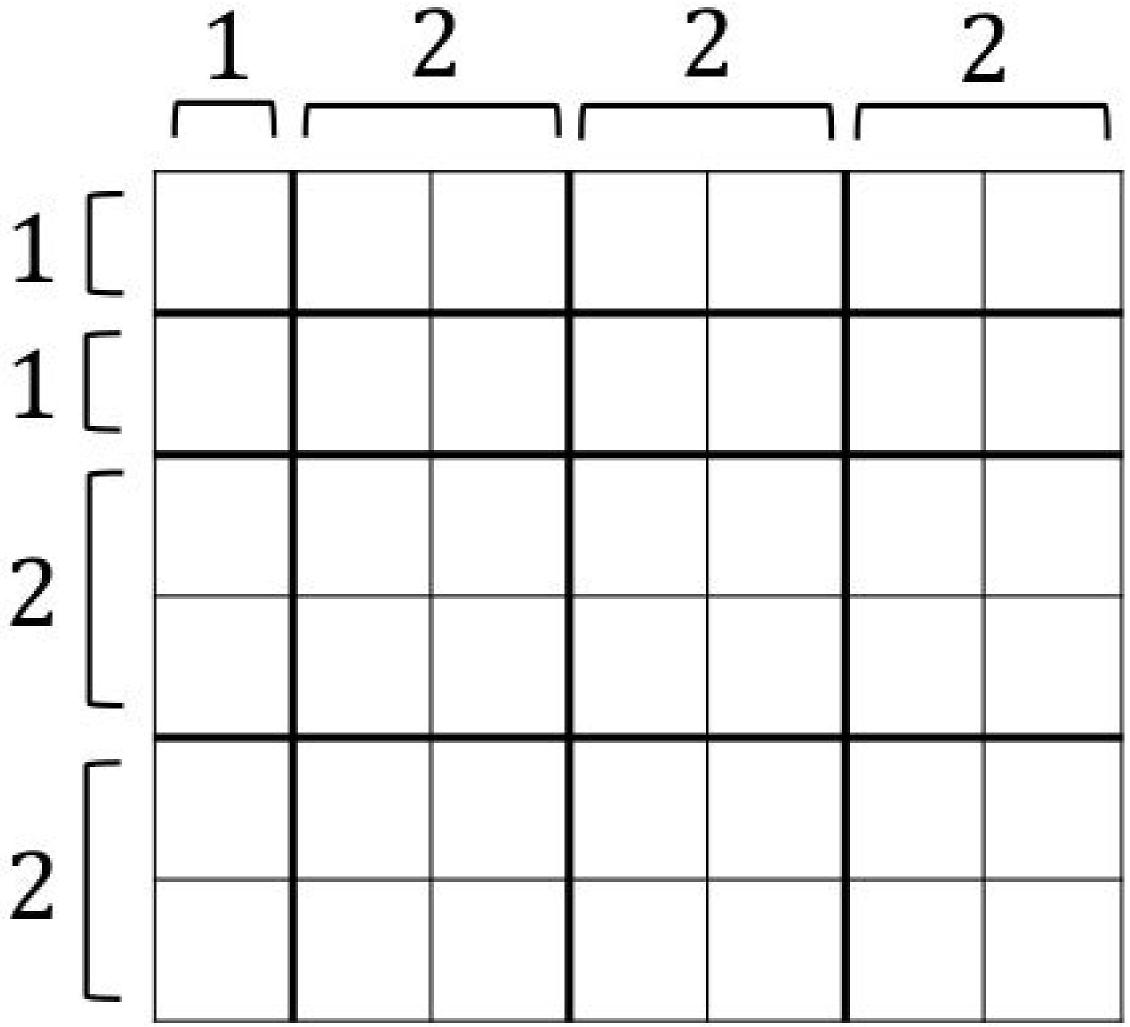
J. 분할

Depth 2



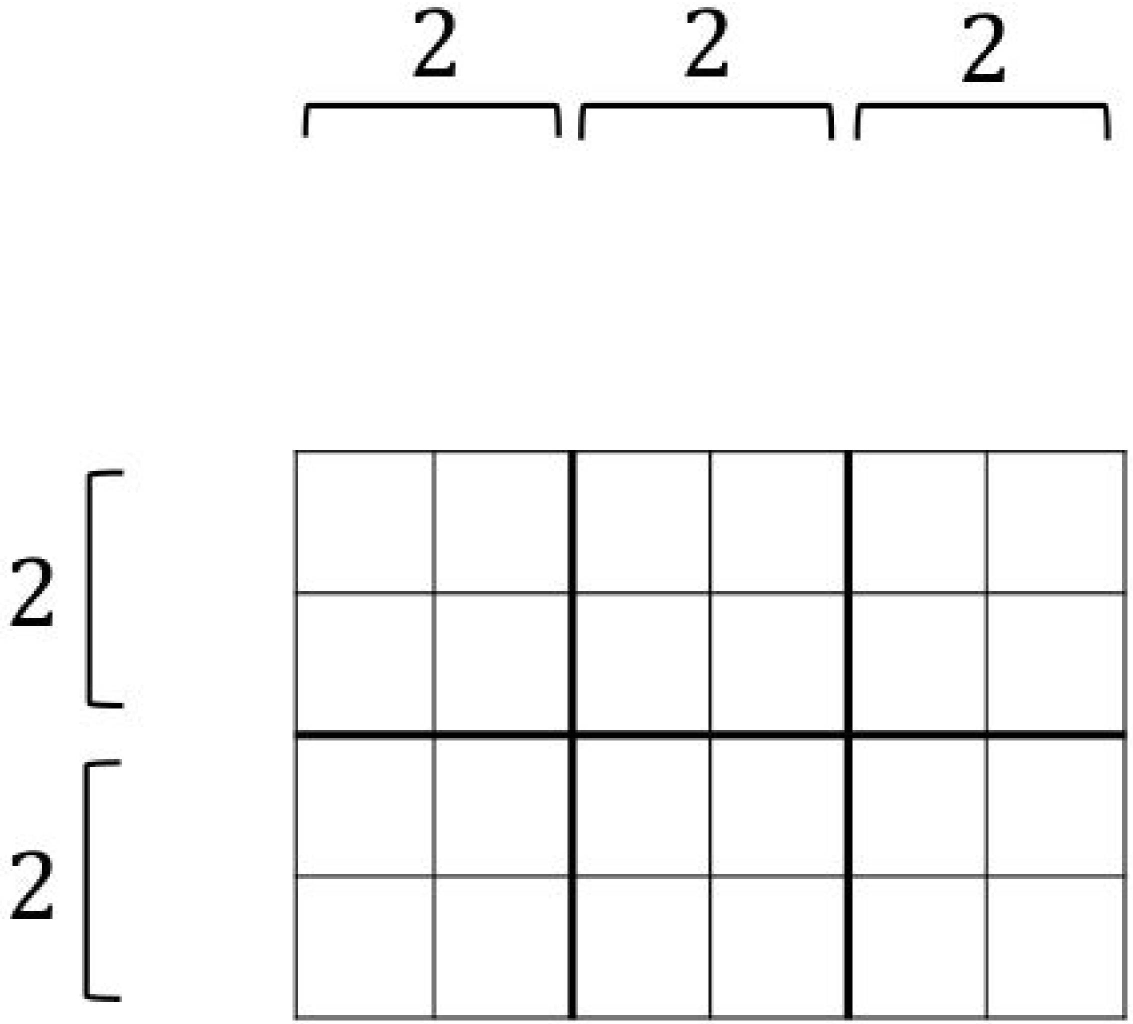
J. 분할

Depth 2



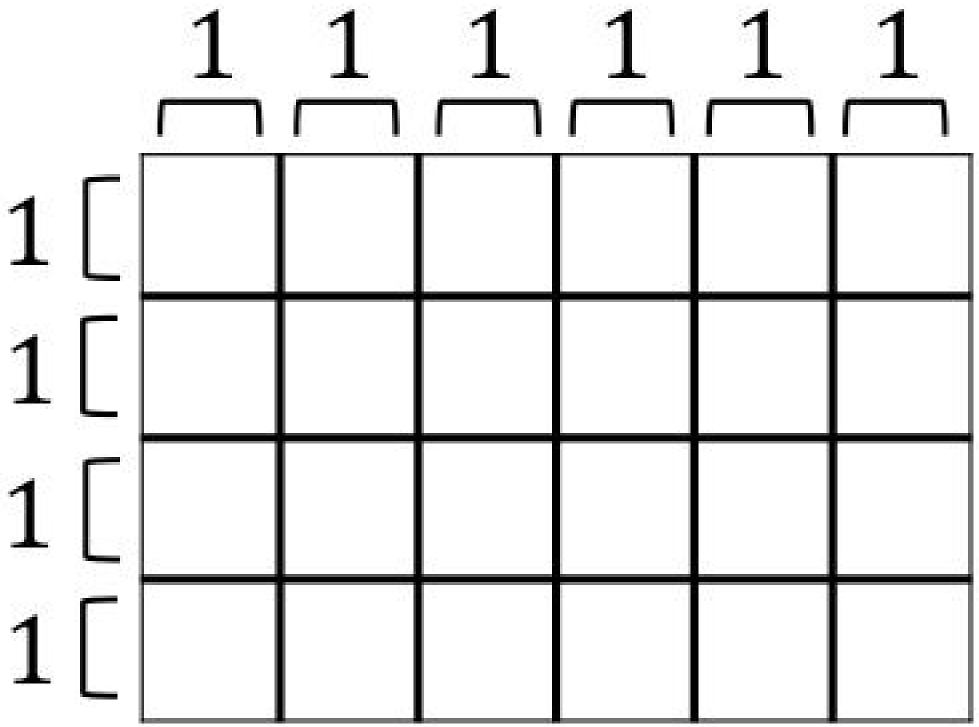
J. 분할

Depth 2



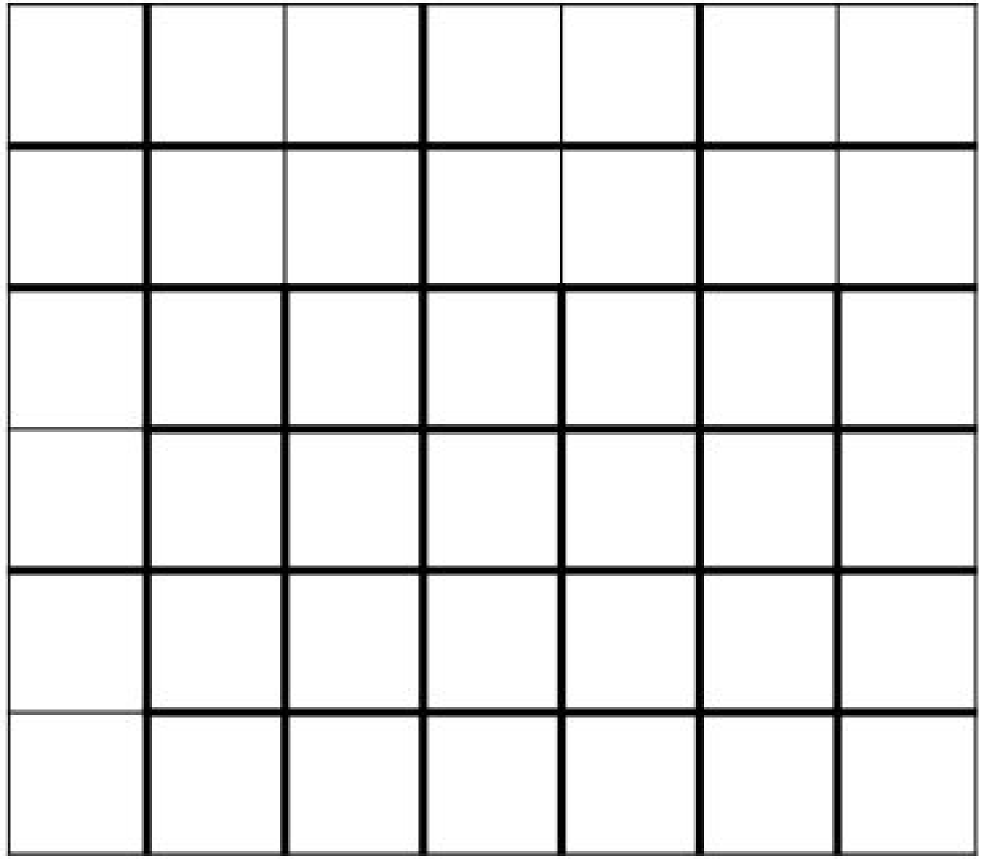
J. 분할

Depth 3



J. 분할

결과



J. 분할

- 실제로 할 때는 배열의 크기가 너무 크기 때문에, 같은 depth에서 1차원 크기의 종류가 두 개 이하, 2차원 배열의 종류는 네 개 이하라는 사실을 이용하여 각 크기에 대해 개수가 몇 개인지 저장하여 이전 depth에서 다음 depth에 있는 배열 종류와 그 개수를 저장하여 문제를 해결할 수 있습니다.

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 있다면? – 최대 길이가 l 이상이어도 선택 가능

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 있다면? – 최대 길이가 l 이상이어도 선택 가능
  - 없다면? – 최대 길이가 l 이하여도 선택 불가능

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 있다면? – 최대 길이가 l 이상이어도 선택 가능
  - 없다면? – 최대 길이가 l 이하여도 선택 불가능
  - 따라서 결정 문제로 바뀐다

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 있다면? – 최대 길이가 l 이상이어도 선택 가능
  - 없다면? – 최대 길이가 l 이하여도 선택 불가능
  - 따라서 결정 문제로 바뀐다
  - 이분 탐색(Parametric Search)을 통해 l의 값을 정할 수 있다

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - u를 루트로 하는 각 서브트리마다,

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - u를 루트로 하는 각 서브트리마다,
    - 최대 길이가 l 이하이면서

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - u를 루트로 하는 각 서브트리마다,
    - 최대 길이가 l 이하이면서
    - 선택하는 간선의 수를 최대화하면서

K. The festival must go on

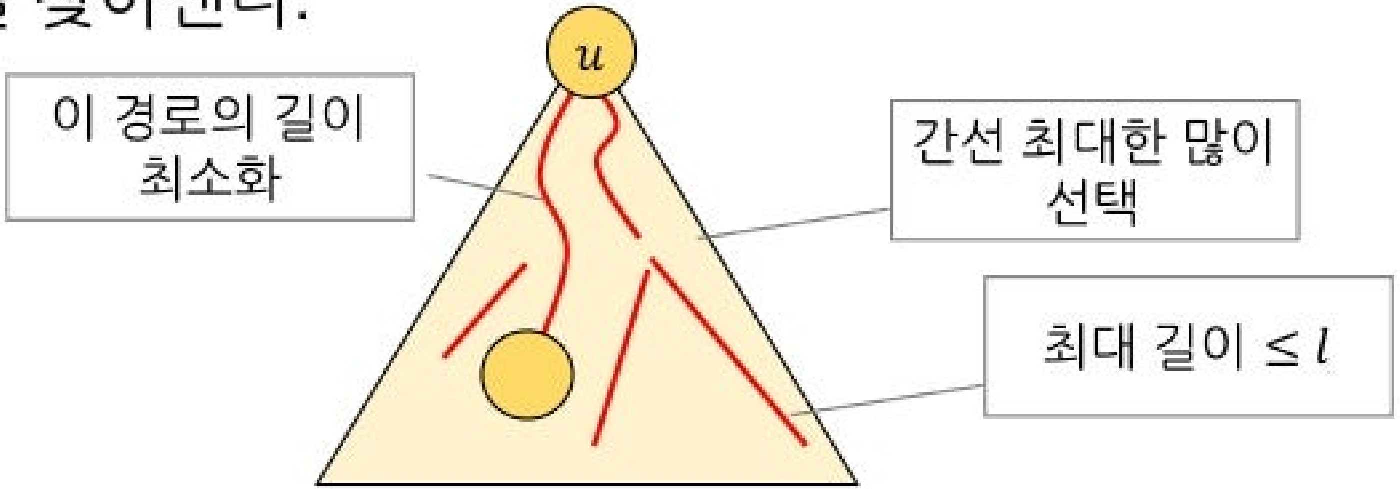
- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - u를 루트로 하는 각 서브트리마다,
    - 최대 길이가 l 이하이면서
    - 선택하는 간선의 수를 최대화하면서
    - 루트에서 끝나는 선택된 간선들로만 이루어진 경로의 최대 길이를 최소화하는

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - u를 루트로 하는 각 서브트리마다,
    - 최대 길이가 l 이하이면서
    - 선택하는 간선의 수를 최대화하면서
    - 루트에서 끝나는 선택된 간선들로만 이루어진 경로의 최대 길이를 최소화하는
  - 선택 방법을 찾아낸다.

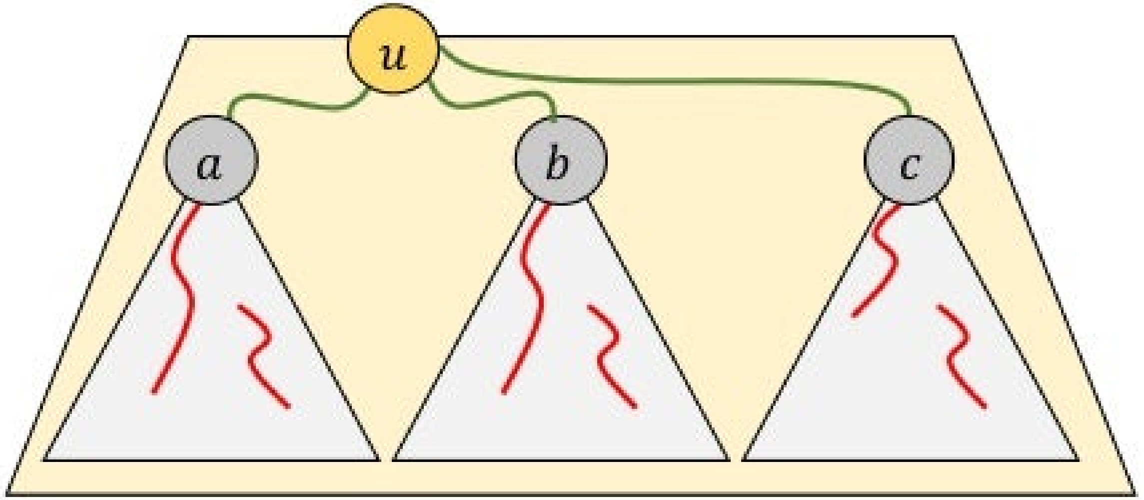
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - u를 루트로 하는 각 서브트리마다,
    - 최대 길이가 l 이하이면서
    - 선택하는 간선의 수를 최대화하면서
    - 루트에서 끝나는 선택된 간선들로만 이루어진 경로의 최대 길이를 최소화하는
  - 선택 방법을 찾아낸다.



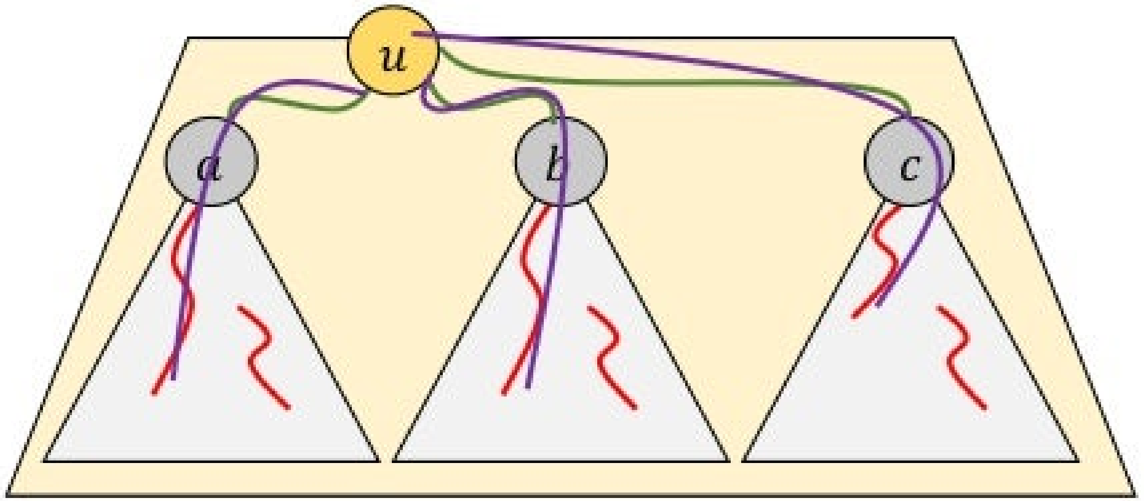
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



K. The festival must go on

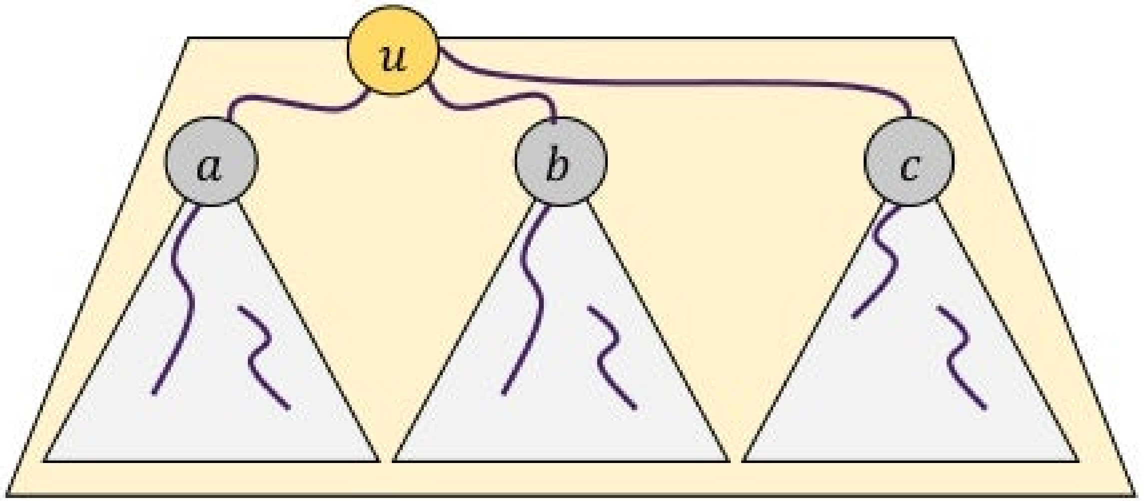
- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



루트가 끝인 경로의 길이가 긴 순으로 정렬

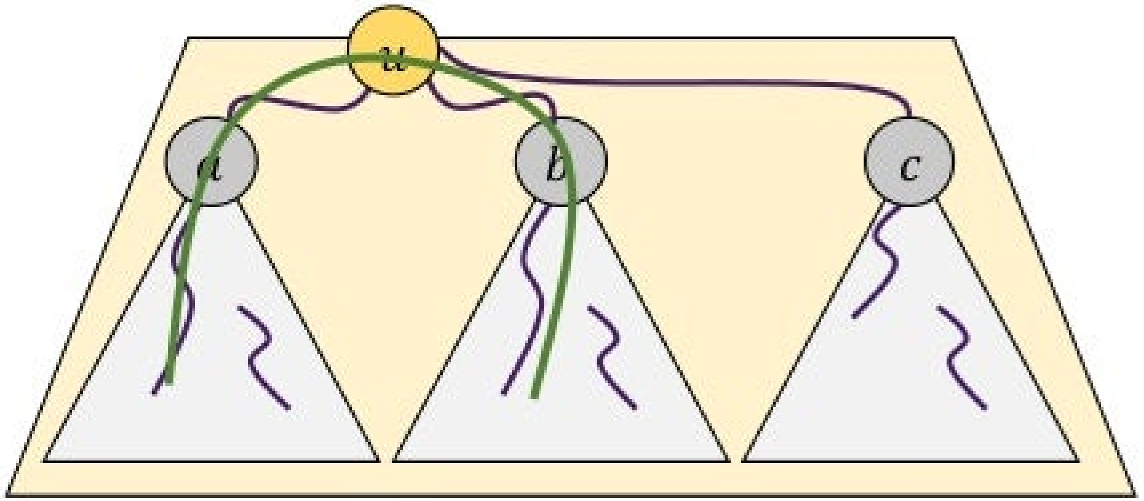
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



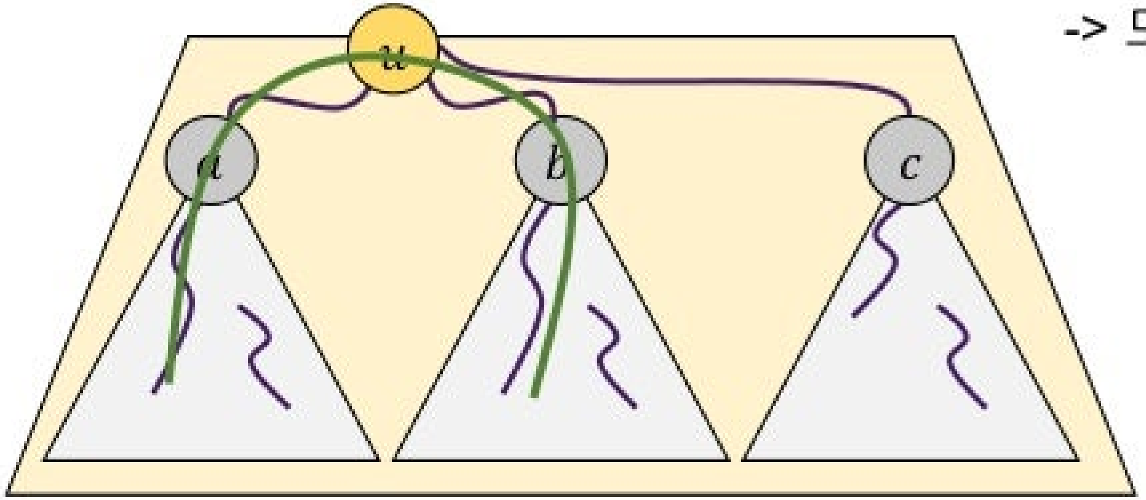
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



K. The festival must go on

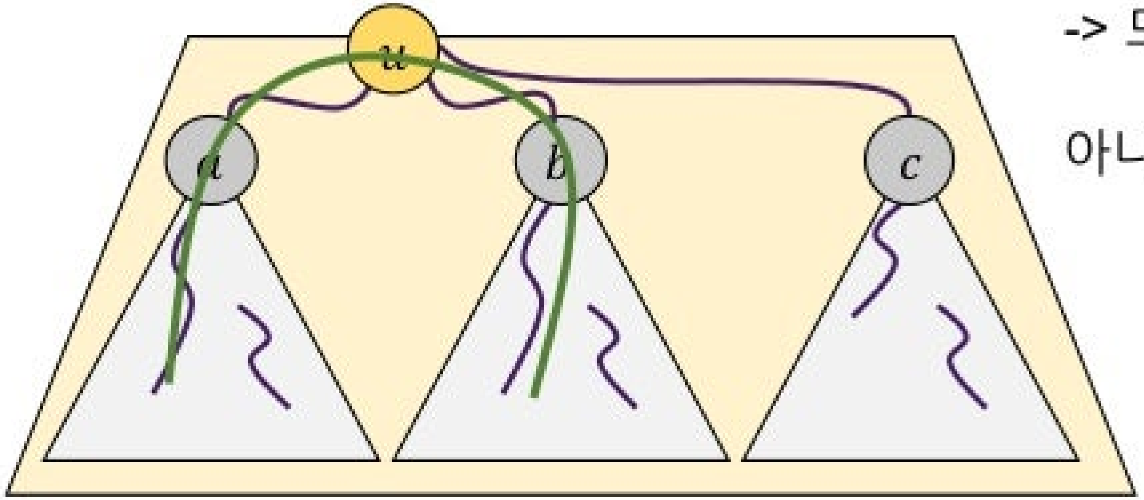
- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



초록색 경로의 길이 ≤ l이면?
-> 모든 보라색 간선 선택 가능

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘

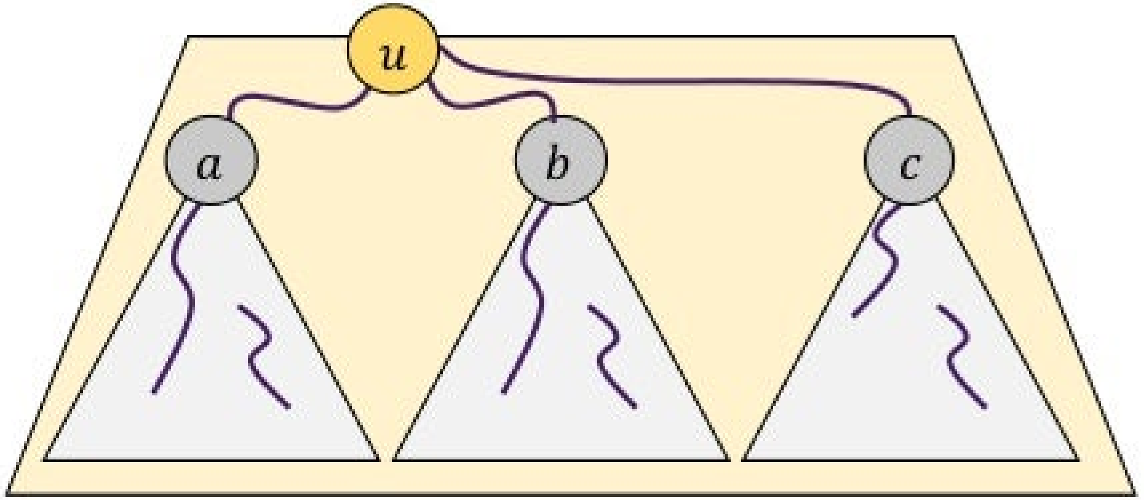


초록색 경로의 길이 $\leq l$이면?
-> 모든 보라색 간선 선택 가능

아니면?

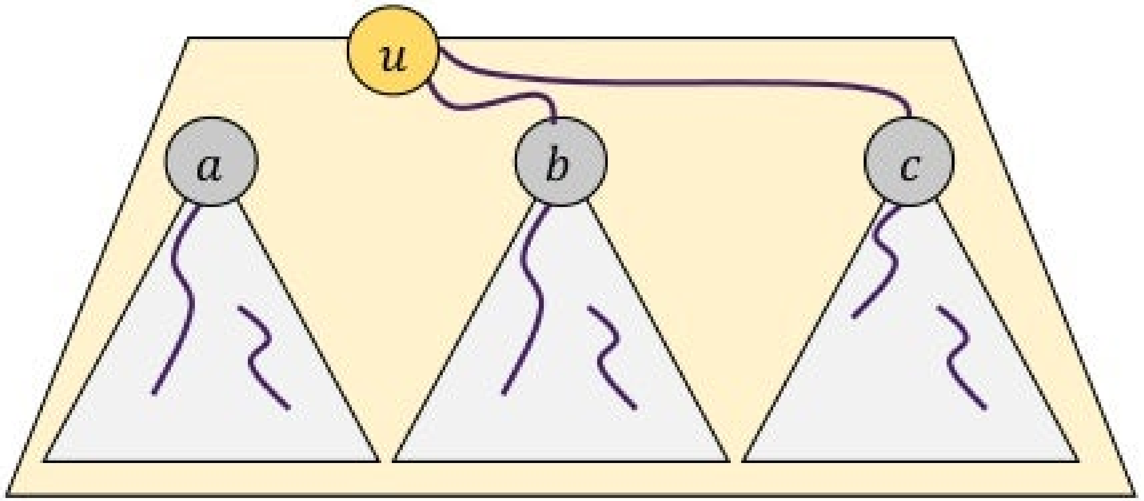
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



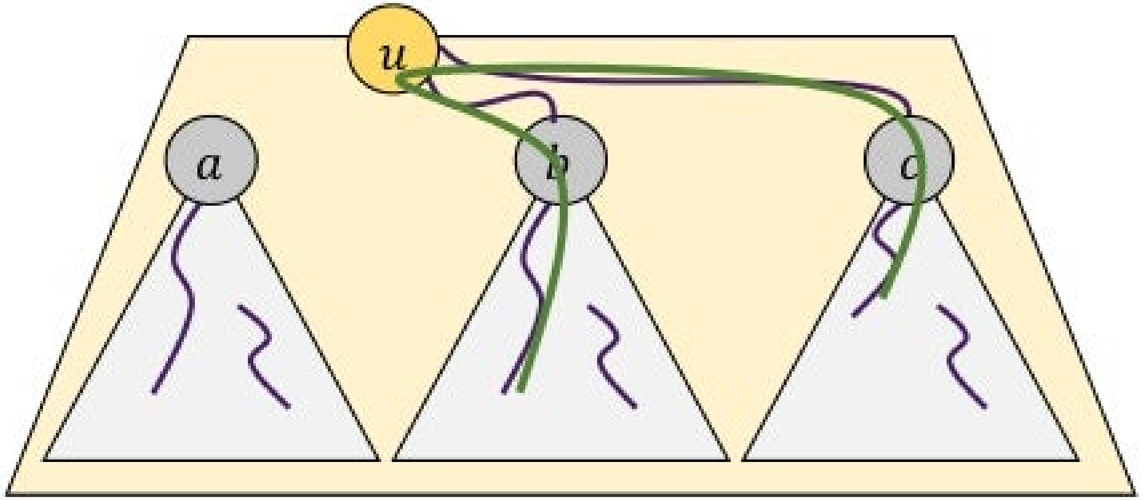
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



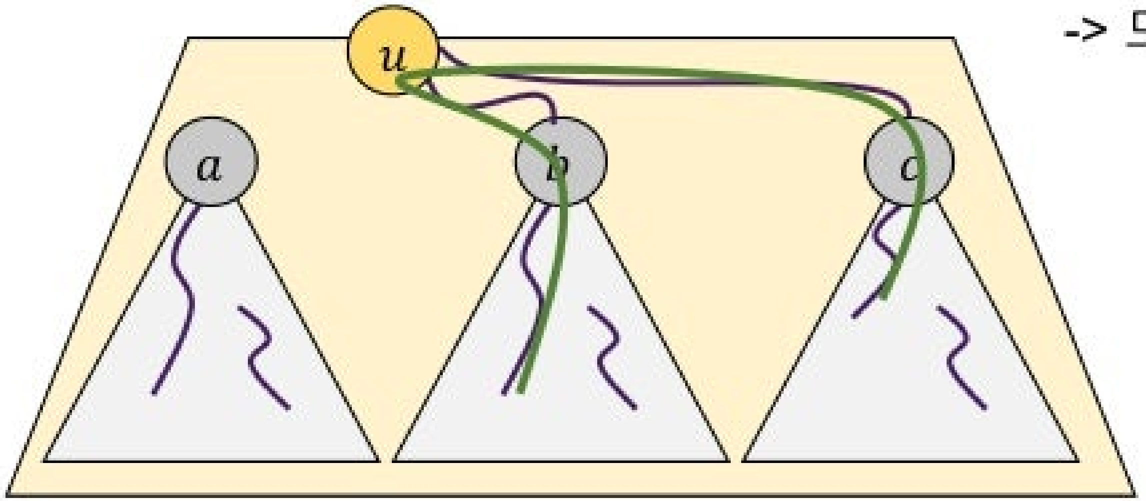
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



K. The festival must go on

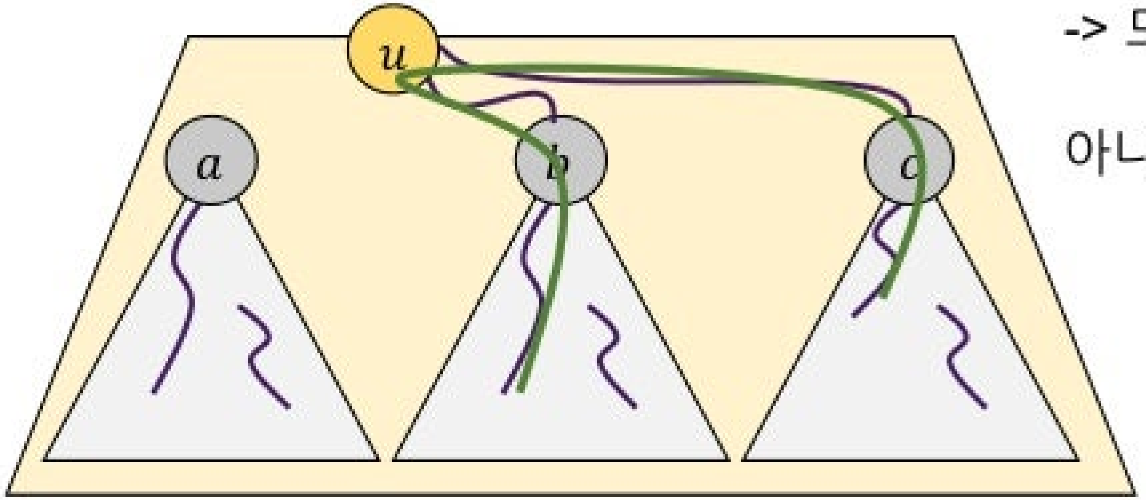
- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



초록색 경로의 길이 $\leq l$이면?
-> 모든 보라색 간선 선택 가능

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘

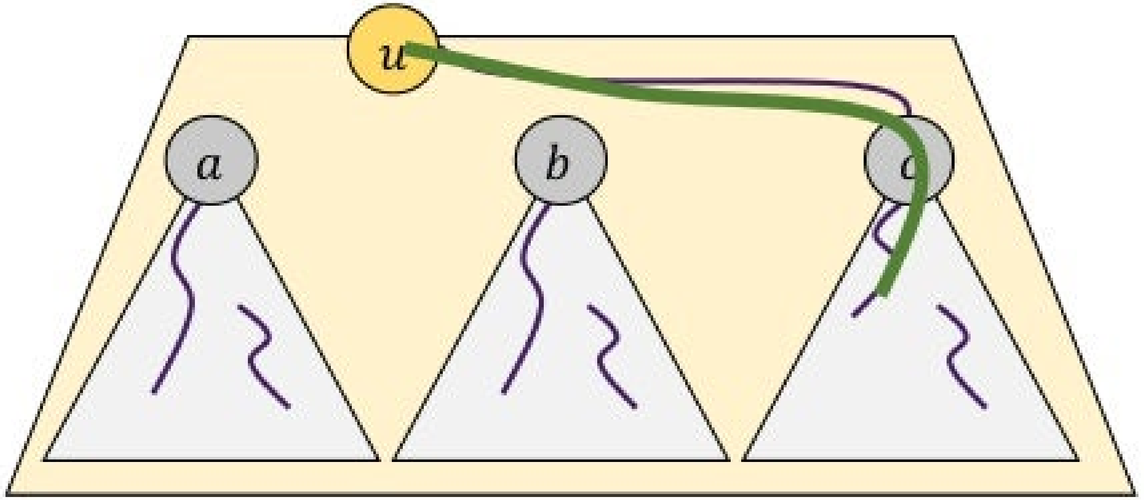


초록색 경로의 길이 $\leq l$이면?
-> 모든 보라색 간선 선택 가능

아니면?

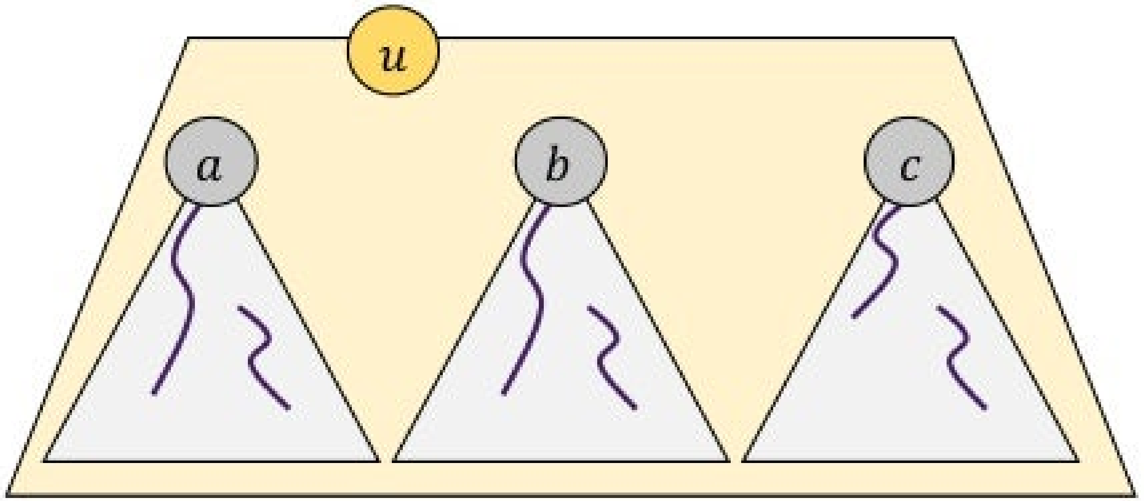
K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘



K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - 이런 식으로 재귀적으로 문제를 해결할 수 있습니다.

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - 이런 식으로 재귀적으로 문제를 해결할 수 있습니다.
  - 이게 왜 되는지 적고 싶은데 시간이 없네요.

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - 이런 식으로 재귀적으로 문제를 해결할 수 있습니다.
  - 이게 왜 되는지 적고 싶은데 시간이 없네요.
- 시간복잡도: $O(\log \text{길이} \times N \log N)$
  - 이분탐색 x 정점 수 x 정렬

K. The festival must go on

- 최대 길이가 l일 때 M개의 간선을 선택할 수 있는가?
  - 요약: 탐욕 알고리즘
  - 이런 식으로 재귀적으로 문제를 해결할 수 있습니다.
  - 이게 왜 되는지 적고 싶은데 시간이 없네요.
- 시간복잡도: $O(\log 길이 \times N \log N)$
  - 이분탐색 x 정점 수 x 정렬
- 처음 만든 풀이: DP를 활용한 $O(N^2 \log 길이)$
  - 이분탐색의 아이디어는 같고, 냅색 같은 걸 합니다.
  - 역시 설명하고 싶으나 시간이 없습니다.

L. 수열의 장인

- 답은 0 이상
  - $n \geq 2$이기 때문

L. 수열의 장인

- 답은 0 이상
  - $n \geq 2$이기 때문
- 곱의 최댓값이 아니라 합의 최댓값이었다면?
  - $T[i] = \max\{T[i-1], 0\} + a_i$
  - (답) = $\max\{T[i]\}$

L. 수열의 장인

- 답은 0 이상
  - $n \geq 2$이기 때문
- 곱의 최댓값이 아니라 합의 최댓값이었다면?
  - $T[i] = \max\{T[i-1], 0\} + a_i$
  - (답) = $\max\{T[i]\}$
- 곱의 최댓값도 저렇게?
  - $T[i] = \max\{T[i-1], 1\} \times a_i$
  - (답) = $\max\{T[i]\}$

L. 수열의 장인

- 답은 0 이상
  - $n \geq 2$이기 때문
- 곱의 최댓값이 아니라 합의 최댓값이었다면?
  - $T[i] = \max\{T[i-1], 0\} + a_i$
  - (답) = $\max\{T[i]\}$
- 곱의 최댓값도 저렇게?
  - $T[i] = \max\{T[i-1], 1\} \times a_i$
  - (답) = $\max\{T[i]\}$
  - 문제점: $(-\mathbf{2^2}) \times (-2) > \mathbf{2} \times (-2)$

L. 수열의 장인

- 답은 0 이상
  - $n \geq 2$이기 때문
- 곱의 최댓값이 아니라 합의 최댓값이었다면?
  - $T[i] = \max\{T[i-1], 0\} + a_i$
  - (답) = $\max\{T[i]\}$
- 곱의 최댓값도 저렇게?
  - $T[i] = \max\{T[i-1], 1\} \times a_i$
  - (답) = $\max\{T[i]\}$
  - 문제점: $(-\mathbf{2}^{\mathbf{2}}) \times (-2) > \mathbf{2} \times (-2)$
    - 음수와 음수를 곱하면 더 큰 양수가 만들어질 수 있다!

L. 수열의 장인

- 그러니 최소, 최대를 저장하자!

L. 수열의 장인

- 그러니 최소, 최대를 저장하자!
  - 최소 – 음수 중 절댓값이 가장 큰 것, 최대 – 양수 중 최댓값이 가장 큰 것 ..일 가능성 있음

L. 수열의 장인

- 그러니 최소, 최대를 저장하자!
  - 최소 – 음수 중 절댓값이 가장 큰 것, 최대 – 양수 중 최댓값이 가장 큰 것 ..일 가능성 있음
  - 이후의 풀이는 다 틀렸으므로 생략하고 언젠가 다시 올릴게요!