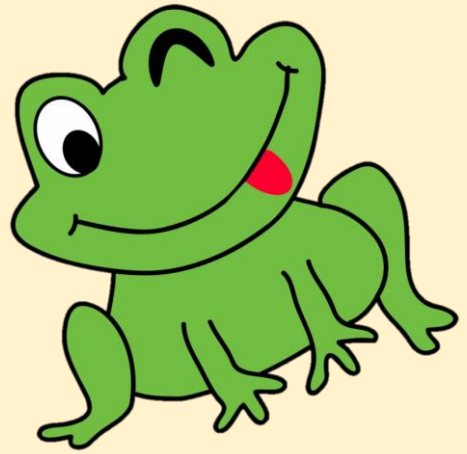
TMITalk: Too Much Information Talk

점프하는 개구리 문제:



백트래킹으로 풀어보는 점프하는 개구리 문제

백트래킹으로 풀어보는 점프하는 개구리 문제

- 점프하는 개구리: 흥미로운 수학 문제
  - 1부터 N까지의 숫자가 적힌 동전이 원 위에 놓여 있다.
  - 점프하는 개구리는 해당 동전 위에서
    - 동전에 적힌 숫자 크기에 따라 시계 방향으로 점프를 하며 여행을 한다.
  - 만약 개구리가 1이 적힌 동전 위에 있다면,
    - 개구리는 원 위에 놓인 모든 동전에 방문할 수 있을까?
  - 이런 방법으로 모든 동전을 방문할 수 있는 순환 배열을 가진
    - 숫자 N을 모두 찾아라. (단, $N \geq 2$)

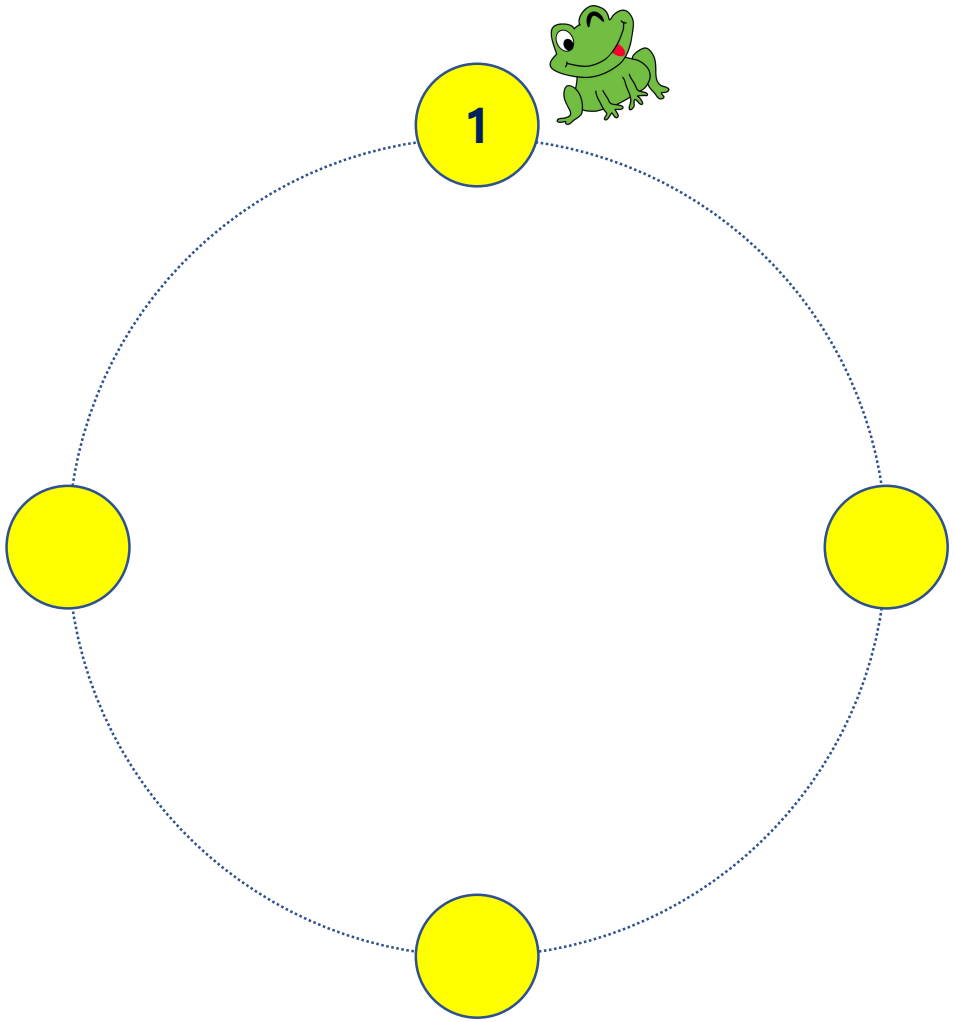
문제 출처: 2021 Pan-American Girl's Mathematical Olympiad

https://artofproblemsolving.com/community/c2499895_2021_panamerican_girls_mathematical_olympiad

백트래킹으로 풀어보는 점프하는 개구리 문제

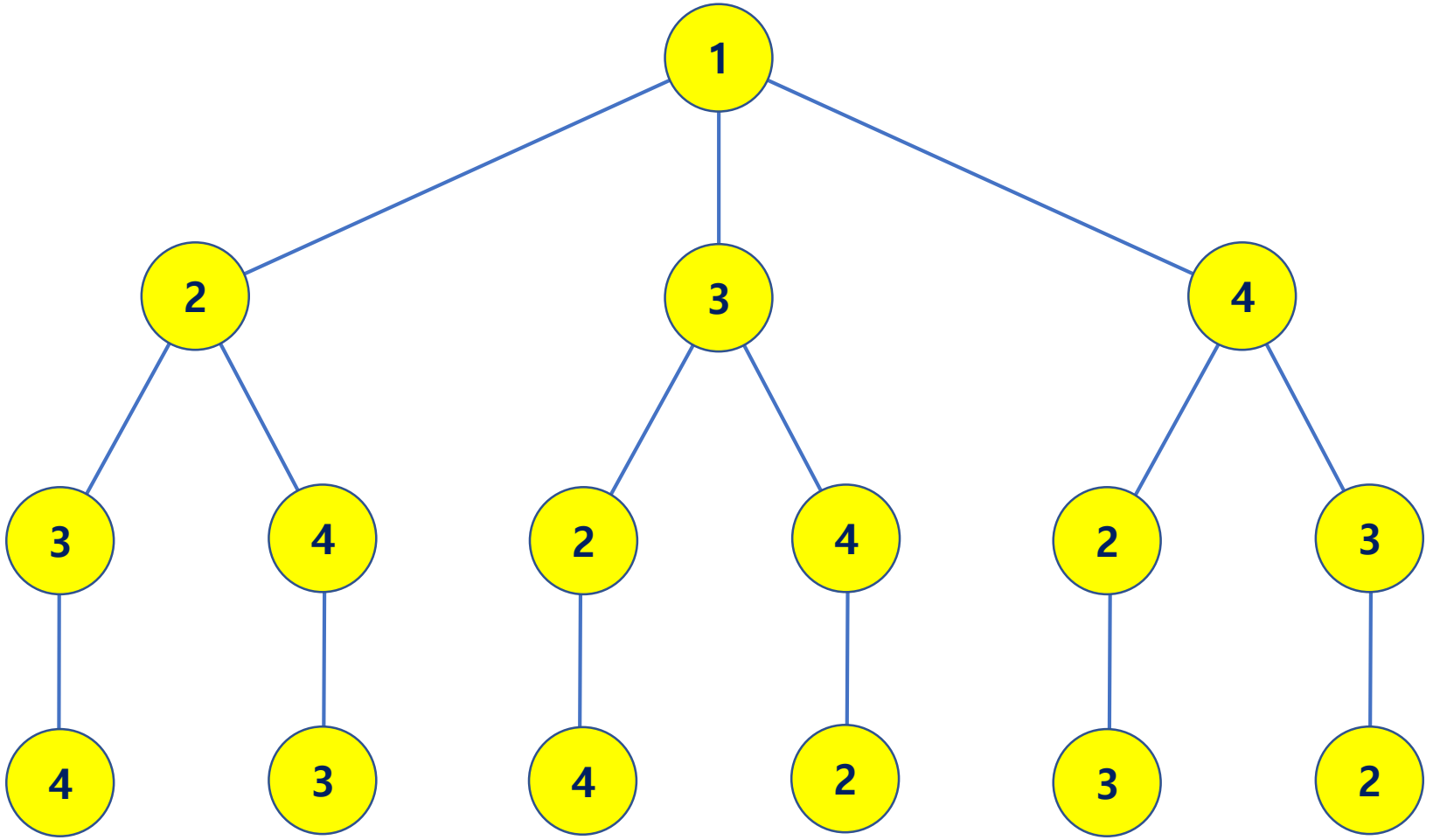
$N = 4$

```
[1, 2, 3, 4]
[1, 2, 4, 3]
[1, 3, 2, 4]
[1, 3, 4, 2]
[1, 4, 2, 3]
[1, 4, 3, 2]
```



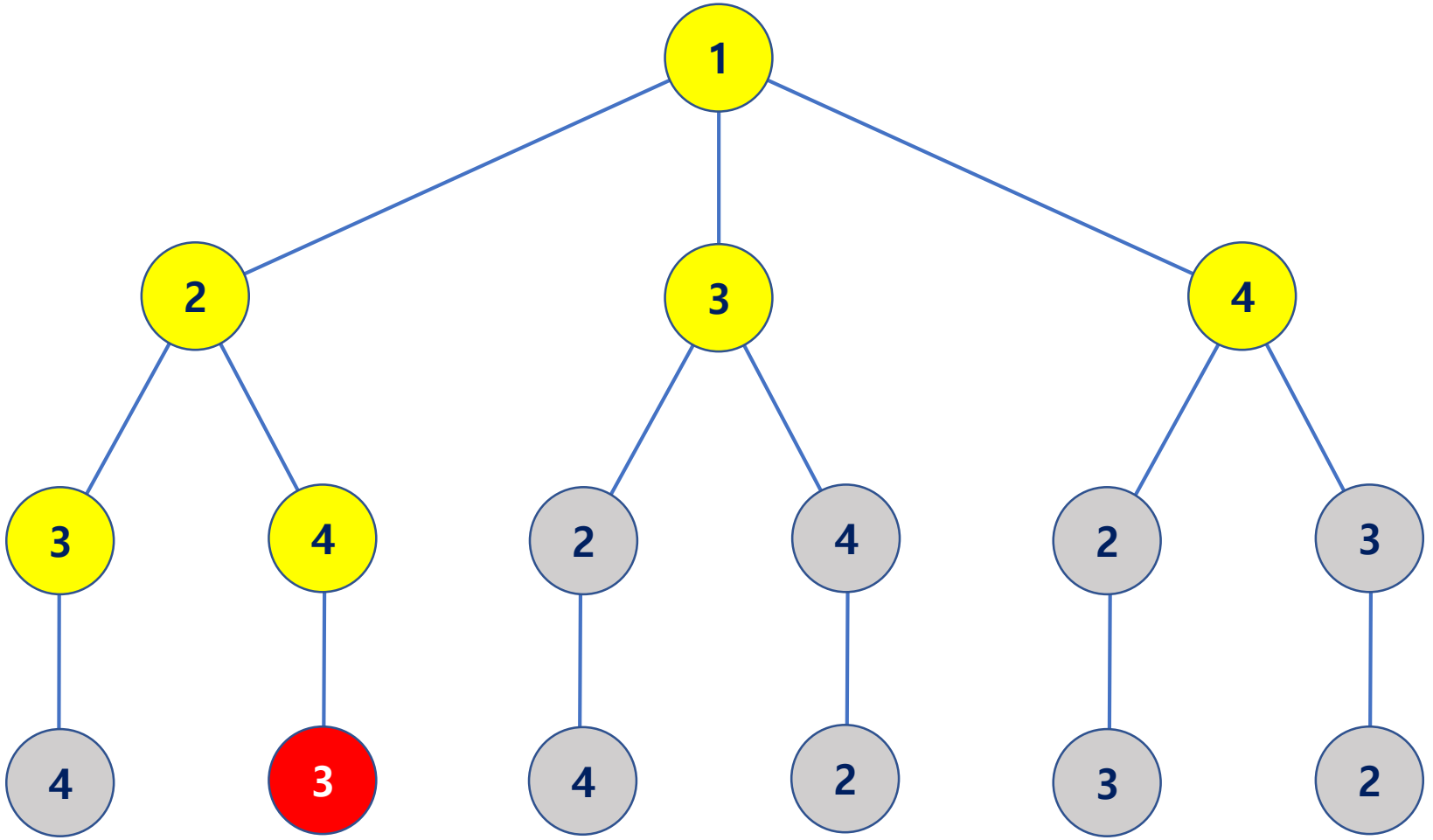
백트래킹으로 풀어보는 점프하는 개구리 문제

- **상태 공간 트리**: State Space Tree



백트래킹으로 풀어보는 점프하는 개구리 문제

- **가지치기**: Pruning



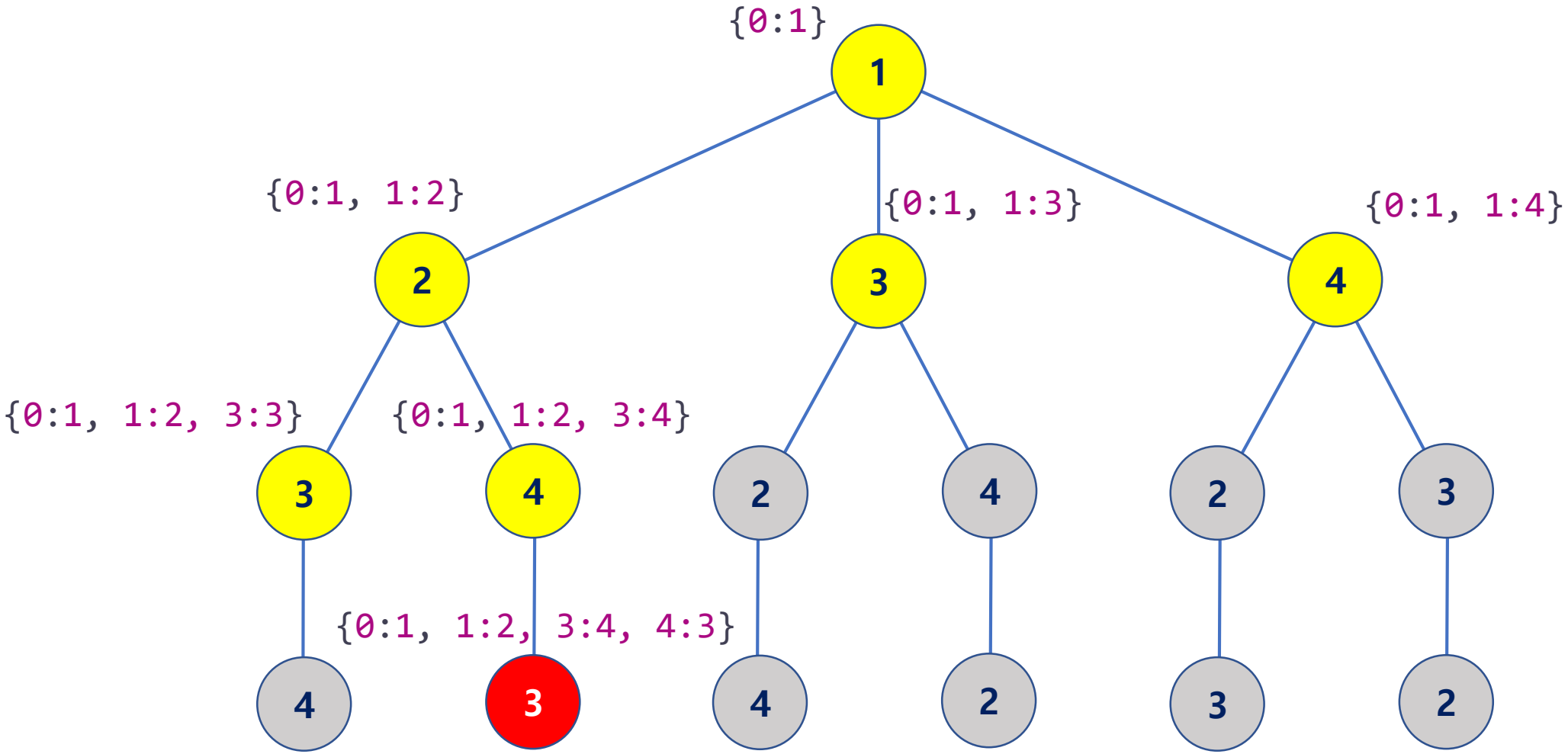
백트래킹으로 풀어보는 점프하는 개구리 문제

- **문제해결 테크닉:**
  - 순환 배열을 딕셔너리(또는 해시맵)로 선언
    - key: 배열의 인덱스, value: 배열의 원소
    - 모두 방문했는가?: 딕셔너리의 길이
    - 방문 가능한가?: 다음 방문 위치의 key가 존재하는가?
    - 이미 방문했는가?: 다음 방문 원소의 value가 존재하는가?

```
n = 4
circle = {0:1}
jump(n, circle, 0)
```

백트래킹으로 풀어보는 점프하는 개구리 문제

- 딕셔너리를 활용한 백트래킹



백트래킹으로 풀어보는 점프하는 개구리 문제

- Backtracking: DFS with Pruning

```
def jump(n, circle, pos):
    if len(circle) == n:
        print("solution:", circle)
    else:
        next_pos = ((pos + circle[pos]) % n)
        if next_pos not in circle:
            for k in range(1, n + 1):
                if k not in circle.values():
                    circle[next_pos] = k
                    jump(n, circle, next_pos)
                    circle.pop(next_pos)
```

점프하는 개구리는 순환할 수 있을까?

- 가능한 경우의 수 세기:

```
def jump(n, circle, pos):
    global count
    if len(circle) == n:
        count += 1
    # 이하 생략
```

```
for n in range(1, 15):
    circle = {0:1}
    count = 0
    jump(n, circle, 0)
    print(n, ":", count)
```

Any Questions?