컴퓨팅 사고력 과제

서울 5반 김영주

문제 12 (논리와 증명)

n^2이 3의 배수 $\longrightarrow$ n은 3의 배수

대우 n이 3의 배수 X $\longrightarrow$ n^2은 3의 배수 X

즉 $n = 3k+1$ 이거나 $n = 3k+2$ 이다

따라서 $n^2 = (3k+1)^2 = 9k^2 + 6k + 1 = 3(3k^2+2k) + 1$

$n^2 = (3k+2)^2 = 9k^2 + 12k + 4 = 3(3k^2+4k+1) + 1$

) 3x+1 의 형태

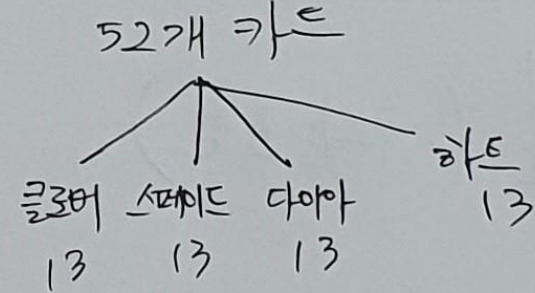
이므로 n^2도 3의 배수가 아니게 된다.

대우명제가 참이므로, 원명제인 "n^2이 3의 배수 $\longrightarrow$ n도 3의 배수"는 참이다.

문제4 (수와 표현)

$$x = \log_a yz = \log_a y + \log_a z = \frac{\log_2 y}{\log_2 a} + \frac{\log_2 z}{\log_2 a} = \frac{\log_2 y + \log_2 z}{\log_2 a}$$

문제 16 (집합과 조합론)



$_{13}C_5$ → 13개 숫자 중 5개 숫자 뽑기

$_4C_1$ → 4개 무늬 중 1개 뽑기

$$\therefore \underbrace{{}_{13}C_5 \times {}_4C_1 \times {}_4C_1 \times {}_4C_1 \times {}_4C_1 \times {}_4C_1}_{\text{5개의 조합}}$$

$$= \frac{13!}{(13-5)!\,5!} \times 4^5 = 1,317,888$$

문제 8 (기초수식)

$$T(n) = T(n-1) + \frac{1}{n}$$

$$= T(n-2) + \frac{1}{n-1} + \frac{1}{n}$$

$$= T(n-3) + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n}$$

$$\vdots$$

$$= T(n-n) + \frac{1}{n-(n-1)} + \frac{1}{n-(n-2)} + \cdots + \frac{1}{n}$$

$$= T(0) + 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

$$= T(0) + \sum_{i=1}^{n} \frac{1}{i}$$

$$\Rightarrow T(n) = O(\log n)$$

문제6 (재귀)

```
def recursion(node, ancestors, generations):
    if node in tree:
        for i, child in enumerate(tree[node], start=1):
            recursion(child, ancestors + [node], generations + [i])
    else:
        ancestors += [node]
        rtn = f'[{str(ancestors[0]).zfill(3)}]' if not printed else '     '
        for i, ancestor in enumerate(ancestors[1:], start=1):
            if ancestor in printed:
                rtn += '   ¦        '
            else:
                sibling_count = len(tree[ancestors[i - 1]])
                sibling_ranking = generations[i]
                number = str(ancestor).zfill(3)

                # 유일한 자식
                if sibling_count == 1:
                    rtn += f' ----- [{number}]'
                # 첫번째 자식
                elif sibling_ranking == 1:
                    rtn += f' --+-- [{number}]'
                # 마지막 자식
                elif sibling_count == sibling_ranking:
                    rtn += f'   L-- [{number}]'
                # 그외
                else:
                    rtn += f'   +-- [{number}]'

                printed.add(ancestor)
        print(rtn)

tree = {}
edges = list(map(int, input().split()))
for i in range(0, len(edges) - 1, 2):
    tree[edges[i]] = tree.get(edges[i], []) + [edges[i + 1]]

printed = set()
recursion(edges[0], [], [1])
          #node    #anc #gen
```

결과

```
30 54 30 2 30 45 54 1 54 3 45 123 1 101 1 102 3 103
[030] --+-- [054] --+-- [001] --+-- [101]
        ¦           ¦           L-- [102]
        ¦           L-- [003] ----- [103]
        +-- [002]
        L-- [045] ----- [123]
```