04/07실습문제

2021/4/7 (수) (논리와 증명)

문제1

항진명제 증명.

$(\sim p \vee q) \vee (p \wedge \sim q)$

i) $p=T$ $q=T$.

$(F \vee T) \vee (T \wedge F)$

$= T \vee F = T$.

ii) $p=T$ $q=F$.

$(F \vee F) \vee (T \wedge T)$

$F \vee T = T$

iii) $p=F$, $q=T$

$(T \vee T) \vee (F \wedge F)$

$= T \vee F = T$

iv) $p=F$, $q=F$

$(T \vee F) \vee (F \wedge T)$.

$= T \vee F = T$.

문제2

모순명제, 증명.

$(p \wedge q) \wedge (p \wedge \sim q)$

i) $p=T$ $q=T$

$T \wedge F = F$

ii) $p=T$ $q=F$.

$F \wedge T = F$

iii) $p=F$ $q=T$

$F \wedge F = F$

iv) $p=F$ $q=F$.

$F \wedge F = F$

~~T∧F=F.~~ ~~F∨T~~ ~~F∨F.~~

문제4

명제 간소화.

$(p \vee \sim q) \wedge (\sim p \vee \sim q)$

$= (p \wedge \sim p) \vee \sim q = F \vee \sim q. = \sim q$

문제3

$\sim p \vee q$ 와 $\sim(p \vee q)$ 동등한지

~~q (p∨q) p∧(p∨~~

i) $p=T$ $q=T$.

$\sim p \vee q = T$ / $\sim(p \vee q) = F$

ii) $p=T$ $q=F$.

$\sim p \vee q = F$ / $\sim(p \vee q) = F$.

iii) $p=F$, $q=T$

$\sim p \vee q = T$ / $\sim(p \vee q) = F$.

iv) $p=F$, $q=F$.

$\sim p \vee q = T$ / $\sim(p \vee q) = T$.

문제5

②

$^{\forall}x \in Z$, $x^2 \geq x$. 참.

$x^2 - x \geq 0$.

$x^2 - x + \frac{1}{4} \geq \frac{1}{4}$

$(x - \frac{1}{2})^2 \geq \frac{1}{4}$

④

$\exists x \in Z$, $x^2 < x$. 거짓.

$x^2 - x < 0$

$(x - \frac{1}{2})^2 < \frac{1}{4}$

문제17

n이 홀수면 n^2+n은 짝수 증명.

sol) $n = 2k+1$

$n^2+n = (2k+1)^2 + (2k+1)$

$= 4k^2 + 4k + 1 + 2k + 1$

$= 4k^2 + 6k + 2$.

$= 2(2k^2 + 3k + 1)$ ∴ 짝수.

04/08 실습문제

2021. 04. 08. 실습문제.

논리와 증명 $\boxed{9}$ (대우를 증명)

자연수 n 에 대해, n^2+5 가 홀수 → n은 짝수 증명.

Sol) n이 홀수이면 n^2+5 는 짝수를 증명. (대우)

if) $n = 2k+1$.

$n^2+5 = 4k^2+4k+1+5 = 4k^2+4k+6 = 2(2k^2+2k+3)$

$\boxed{10}$ 자연수 n, n^2+5n+3 은 항상 홀수 증명.

Sol) Ⅰ) $n = 2k$.

$n^2+5n+3 = 4k^2+10k+3 = 2(2k^2+5k+1)+1$. → 홀수.

Ⅱ) $n = 2k+1$

$n^2+5n+3 = 4k^2+4k+1+10k+5+3 = 4k^2+14k+9 = 2(2k^2+7k+4)+1$ → 홀수.

수와 표현. $\boxed{2}$ 스무고개. 맞출 수 있는 답의 종류는?

Sol) 2^{20}

$\boxed{3-2}$

$2^{\frac{n}{2}}$ (<) $\sqrt{3^n}$

Sol) $\Rightarrow 2^{\frac{n}{2}} < 3^{\frac{n}{2}}$

$\boxed{3-4}$

$\log 2^{2n}$ (<) $n\sqrt{n}$.

$\Rightarrow 2n < n\sqrt{n}$.

$\because$ (n이 충분히 큰수이므로 $\sqrt{n} > 2$)

$\boxed{5-2}$ $f(x) = 3\log(x+3)+1$. 의 역함수.

Sol) $3\log(x+3) = y-1$

$\log(x+3) = \frac{y-1}{3}$

$x+\cancel{3} = 2^{\frac{y-1}{3}} - 3$.

$\cancel{f^{-1}(x) = 2^{\frac{f(x)-1}{3}} - 3}$.

$\cancel{f^{-1}(x) = 2^{\frac{y-1}{3}} - 3}$.

$\therefore f^{-1}(x) = 2^{\frac{f(x)-1}{3}} - 3$.

집합과조합론.

$\boxed{3}$ 문제2 결과 이용해서 n개의 원소를 가진 집합의 가능한 부분집합의 종류는 2^n개 임을 증명.

sol) 문제 2. → $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$.

$x, y = 1$. 일때

n개 중 원소개수 0개 인 부분집합수 $= {}_nC_0$

〃 1 〃 $= {}_nC_1$

〃 2 〃 $= {}_nC_2$.

〃 3 〃 $= {}_nC_3$.

⋮ ⋮

〃 n 〃 $= {}_nC_n$.

다 더하면 $\sum_{k=0}^{n} {}_nC_k$

∴ 2^n = n개 원소 중 가능한 부분집합 종류.

$\boxed{10}$ 비밀번호 0~9 까지 숫자로 만들기. 각 숫자는 최대 한번 사용가능. 4개 ~ 6개 숫자 사용가능. 가능한 비밀번호 가지수?

sol) $\cancel{{}_9C_4}$ ${}_9P_4 + {}_9P_5 + {}_9P_6 = 78624$

(4자리 + 5자리 + 6자리) 일때

$\boxed{13}$ 52개 카드 이용해 만들수 있는 5개 카드 조합 중 같은 무늬 카드가 정확히 3개인 경우는?

sol) $\underbrace{{}_4C_1}_{\text{같은무늬일때 4개중 1개 택}} \times \underbrace{{}_{13}C_3}_{\text{그 무늬중 3장 택.}} \times \underbrace{{}_{39}C_2}_{\text{남은무늬들 카드중 2개 택.}} = 847704$.

∵

기초수식

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

sol) $T(n) = T(n-2) + T(n-1) + n-1 + n.$

$\vdots$

$\leq T(1) + \underbrace{\cdots + (n-1) + n}_{n개}.$

n이 n개 → $n \times n$

$\therefore O(n^2)$

$\boxed{4}$ $T(n) = T\left(\frac{n}{2}\right) + 1$

sol) $\cancel{T(n) = T\left(\frac{n-1}{2}\right) + T\left(\frac{n}{2}\right) + 2}$

$T(n) = T\left(\frac{n}{2^2}\right) + 1 + 1$

$= T\left(\frac{n}{2^3}\right) + 1 + 1 + 1$

$\cancel{= \sum_{k=0}^{n} T\left(\frac{n}{2^k}\right) + k}$

$\frac{n}{2^k} \to 1 \to n = 2^k \to k = \log n$

$= T\left(\frac{n}{2^k}\right) + k.$

$= T(1) + \log n.$

$\therefore O(\log n)$

$\boxed{6}$ $T(n) = 2T\left(\frac{n}{2}\right) + n.$

sol) $T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{2^2}\right) + \frac{n}{2}.$

$T(n) = 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n. = 2^2 \cdot T\left(\frac{n}{2^2}\right) + 2n.$

$= 2^k T\left(\frac{n}{2^k}\right) + k \cdot n.$ $\therefore$ $n = 2^k$, $k = \log n.$

$\frac{n}{2^k} \to 1$

$= n \cdot T(1) + n \cdot \log n.$

$\therefore O(n + n\log n)$

재귀

```python
lst = [69, 10, 30, 2, 16, 8, 31, 22]

def merge_sort(lst):
    if len(lst) == 1:
        return lst
    mid = len(lst)//2
    left = lst[:mid]
    right = lst[mid:]
    left = merge_sort(left)
    right = merge_sort(right)

    return merge(left, right)

# 이건 잘나옴
def merge(left, right):
    result = []
    while len(left) > 0 or len(right) > 0:
        if len(left)> 0 and len(right) > 0:
            if left[0] <= right[0]:
                result.append(left[0])
                left = left[1:]
            else:
                result.append(right[0])
                right = right[1:]
        elif len(left) > 0:
            result.append(left[0])
            left = left[1:]
        elif len(right) > 0:
            result.append(right[0])
            right = right[1:]
    return result

print(merge_sort(lst))
```

동적프로그래밍

memo_fibo.py

```python
def fibo_memo(n, memo_lst):
    if n < 2:
        return n
    if memo_lst[n] == 0:
        memo_lst[n] = fibo_memo(n-2, memo_lst) + fibo_memo(n-1, memo_lst)
    return memo_lst[n]

print(fibo_memo(10, [0]*11)) # 영부터 시작이니까 인덱스 하나 추가해야함
# => 55

# 계산되는 값이 n 가지니까 시간 복잡도 O(n)
```