5조

- 남정현 (조장)
- 이상원
- 이성렬
- 정현정

[1. 논리와 증명]

1-2번. $(\sim p \vee q) \vee (p \wedge \sim q)$

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

2-2번. $(p \wedge q) \wedge (p \wedge \sim q)$

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

3-2번. $\sim p \vee \sim q$ 와 $\sim (p \vee q)$

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

정답. Not 동등

4-2번. 명제 간소화

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

$$\underbrace{(p \wedge \sim p)}_{F} \vee \sim q$$

정답: $\sim q$

5-2번. 참 확인

$$\forall x \in \underset{\text{정수}}{\mathbb{Z}},\ x^2 \ge x$$

① $x = 0$

$0^2 \ge 0$ (T)

풀이 2)

$x(x-1) \ge 0$

② $x > 0$

$\underbrace{x^2 \ge x}$ 양변 $\div x$

$x \ge 1$ (T)

③ $x < 0$

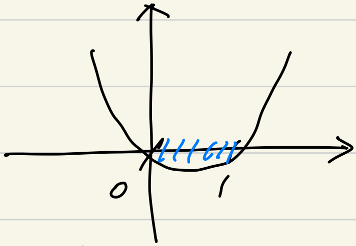
$x<0$

$x \le 1$ (T)

5-4번. $\exists x \in \mathbb{Z}$, $x^2 < x$

$x^2 < x$

$x(x-1) < 0$



$0 < x < 1$ 구간에 정수 X

정답 : F

7번. n이 홀수 → $n^2 + n$은 짝수

$n = 2k+1$ $\quad (2k+1)^2 + (2k+1) = 4k^2 + 6k + 2$

$= 2(2k^2 + 3k + 1)$ 짝수.

9번. 자연수 n, n^2+5 가 홀수 → n 짝수

n이 홀수 → $n^2 + 5$ 짝수

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

$= 2(2k^2 + 2k + 3)$ 짝수

10번. n^2이 짝수 → n은 짝수

n이 홀수 → n^2 홀수

$n = 2k+1$ $\quad (2k+1)^2 = 2(2k^2 + 2k) + 1$ 홀수

11번. 자연수 n, n^2+5n+3은 홀수

i) $n=2k$

$4k^2+10k+3 = 2(2k^2+5k+1)+1$ 홀수

ii) $n=2k+1$

$(4k^2+4k+1)+(10k+5)+3 = 4k^2+14k+9$

$= 2(2k^2+7k+4)+1$ 홀수

12번. n^2이 3의 배수, $\longrightarrow$ n은 3의 배수

n이 3의 배수 X $\longrightarrow$ n^2이 3의 배수 X

i) $n=3k+1$

$n^2=9k^2+6k+1 = 3(3k^2+2k)+1$ 3의 배수 X

ii) $n=3k+2$

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

[2. 수와 표현]

2번. 스무고개, 맞출수 있는 답의 종류

정답 : 2^{20}가지

3-2번. n이 큰값.

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

$2^{\frac{n}{2}} \quad 3^{\frac{n}{2}}$

지수가 $\frac{n}{2}$로 동일, 밑이 더 큰 $3^{\frac{n}{2}}$ 크다.

3-4번. $\log 2^{2n} < n\sqrt{n}$

$2n \cancel{\log 2} \quad n^{\frac{3}{2}}$

$2\cancel{n} \quad \cancel{n}\sqrt{n}$

4번. $x = \log_a yz$,

$$\frac{1}{\log_2 a}\left(\log_2 y + \log_2 z\right)$$

$$\frac{\log yz}{\log a}$$

5-2번. 역함수.

$f(x) = 3\log(x+3) + 1$

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

$2^{\left(\frac{f(x)-1}{3}\right)} = x+3$

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

[3. 집합과 조합]

3. n개의 원소를 가진 집합의 가능한 부분집합 종류 2^n개 증명

집합 $A = \{a_1, a_2, a_3, \cdots, a_n\}$

부분집합 $B_1 = \{0, 0, 0, \cdots, 0\}$ 원소가 있거나, 없거나 2 가지
(1) (0)

$B_2 = \{1, 0, 0, \cdots, 0\}$

$\vdots$

$B_{2^n} = \{1, 1, 1, \cdots, 1\}$

10번 0~9 숫자. 4개 이상 6개 이하 숫자. 비밀번호 가지수

$${}_{10}P_4 + {}_{10}P_5 + {}_{10}P_6$$

$$10\cdot9\cdot8\cdot7 + 10\cdot9\cdot8\cdot7\cdot6 + 10\cdot9\cdot8\cdot7\cdot6\cdot5$$

$$= 10\cdot9\cdot8\cdot7(1+6+30)$$

$$= 10\cdot9\cdot8\cdot7\cdot37$$

$$= 186{,}480$$

정답 : 186, 480 가지

13번. 52개의 카드, 5개 카드 조합 중 같은 무늬 카드 3개인 경우

$$\frac{13\ 12\ 11}{3\ 2\ 1} \times 4 \times \frac{39\cdot38}{2\cdot1}$$

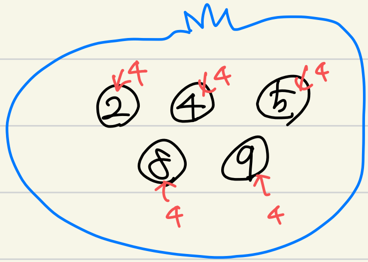
$${}_{13}C_3 \times 4 \times {}_{39}C_2$$

정답. 847, 704 가지

16번. 52개 카드. 5개 카드 조합. 숫자 같은 카드 X

$52 \div 4 = 13$

${}_{13}C_5 \cdot 4^5$



1,317,888 가지

[4. 기초수식]

다음 재귀식들을 O() notation 수준으로 풀 것

#2. $T(n) = T(\cancel{n-1}) + n$ ($T(0)=1$ 로 설정)

$$
\begin{aligned}
T(\cancel{n-1}) &= T(\cancel{n-2}) + (n-1) \\
T(\cancel{n-2}) &= T(\cancel{n-3}) + (n-2) \\
&\vdots \\
T(\cancel{n-k}) &= T(\cancel{n-k-1}) + (n-k) \\
&\vdots \\
+ \quad T(\cancel{1}) &= T(0) + 1
\end{aligned}
$$

$$
\begin{aligned}
T(n) &= n + (n-1) + (n-2) + \cdots 1 + T(0) \\
&= 1 + (1 + \cdots + (n-1) + n) \\
&= 1 + \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n + \frac{3}{2}
\end{aligned}
$$

$O(n^2)$

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

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

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

$\times 2$

$\times 2$

$$\vdots$$

$$T(2) = T(1) + 1$$

$2 \times 2^k = n$

$$T(n) = 1 + \cdots + 1 + T(1)$$

$$= \log_2 n + 1$$

$O(\log n)$

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

$$2T\left(\frac{n}{2}\right) = 4T\left(\frac{n}{4}\right) + \frac{n}{\cancel{2}} \cdot \cancel{2}$$

$$4T\left(\frac{n}{4}\right) = 8T\left(\frac{n}{8}\right) + n$$

$$\vdots$$

$$= \overset{n}{\cancel{2}^{\log n}}\, T(1) + n$$

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

$O(n\log n)$

☆ #8. $T(n) = T(\cancel{n-1}) + \frac{1}{n}$

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

$\vdots$

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

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

$= \ln n - 1 + T(1)$

$$\left(\because \sum_{k=1}^{n} \frac{1}{k} \approx \int_1^n \frac{1}{x}\,dx = \ln x \Big]_1^n = \ln n \right)$$

$O(\log n)$

[5. 재귀]

#2. Merge Sort Time Complexity Proof

MergeSort, 크기 n 배열, 배열 절반으로 나눔, 재귀 정렬, 결과 Merge.

MergeSort 수도 코드 작성, 시간 복잡도 증명

문제 이해) n개의 원소

$$[a_1, a_2, \cdots, a_n]$$

나눈 횟수 : 1

$$[a_1, a_2, \cdots, a_{n/2}], [a_{n/2+1}, \cdots, a_n]$$

⋮

〃 : log n

$$[a_1][a_2] \cdots [a_n]$$

정렬된 리스트 = [n개의 원소 순서대로 정렬됨]

수도코드)

정렬 안된 리스트

```
mergesort (arr) {
    list_my = arr
        n = len(list_my) //2
    left = list_my [ : n]
    right = list_my [n : ]
l = merge sort ( left)
r = mergesort (right)
    return merge(l , r)
}
```

```
if len (arr) = 1
    return arr
```

```
merge( l, r) {
 list_ans = []
  idx1 = 0
  idx2 = 0
while  idx1 < len(l) or  idx2 < len(r) :
      if  l[idx1] > r[idx2] :
           list_ans.append(r[idx2])
           idx2 += 1
      else :
           list_ans.append(l[idx1])
           idx1 += 1

while  idx1 < len(l) :
      list_ans.append(l[idx1])
       idx1 += 1
while  idx2 < len(r) :
      list_ans.append(r[idx2])
      idx2 += 1

return  list_ans }
```

✓ ✓ ✓ ~~1~~ ~~2~~ ~~3~~ 4 4개 → 3번

n개

시간복잡도) n개의 원소

$$[a_1, a_2, \cdots, a_n]$$

나눈 횟수 : 1

$$[a_1, a_2, \cdots, a_{n/2}], [a_{n/2+1}, \cdots, a_n]$$

⋮

〃 : log n

$$[a_1][a_2] \cdots [a_n]$$

정렬된 리스트 = [n개의 원소 순서대로 정렬됨]

전체 소요 : $\underbrace{(n-1)\log n}_{\text{합쳐지는 과정 (정렬)}} + \underbrace{\log n}_{\text{나누는 과정}}$

think while 문

시간 복잡도 : $O(n\log n)$

#4. 위의 소팅 알고리즘에서 Swap 횟수

$\underbrace{(n-1)}_{\text{l vs r 비교}} \underbrace{\log n}_{\text{나눠질 때.}}$

#6. 입력 : 트리 → 출력 알고리즘 작성

```
[030] --+-- [054] ------- [001]
루트    +--[002]
        L-- [045] ------ [123]
```

ex) 입력

30 54 30 2 30 45 54 1 54 3 45 123 1 101 1 102 3 103

1. Input

tree.get(edges[i], [])

없으면 → 빈 리스트 생성해준다.

2. 재귀로 들어가기

set 생성, 루트값 부터 대입

정리)

1. node in tree

leaf 인지 아닌지 확인

2. leaf인 경우 출력 접근. (rtn)

3. printed 이용

이미 출력한 조상 노드인지 확인.

있으면 | 3 부리를 내린다 (세로)

4. 출력 X 조상노드이면, 옆으로 확장 (가로)

- sibling_ranking 과 sibling_count 이용

ranking 이 first, count 와 같을 때, 등으로 분기

[6. 동적프로그래밍]

2. Dynamic Programming 피보나치 수열

$$F(n) = F(n-1) + F(n-2)$$

```
Fibonacci(n)
{   F[0] ← 0
    F[1] ← 1
    for i ← 2 , i ≤ n,  i ← i+1
        F[i] = F[i-1] + F[i-2]
    return F[n]
}
```

1. 리스트 생성 (F)
2. 0, 1 값이 주어짐
3. for 문을 돌면서 결과값이 나온다.