

# 다이나믹 프로그래밍 2 (연습)

최백준 [choi@startlink.io](mailto:choi@startlink.io)

# 동전 1

<https://www.acmicpc.net/problem/2293>

- $n$ 가지 종류의 동전이 있다
- 각각의 동전이 나타내는 가치는 다르다
- 이 동전들을 적당히 사용해서, 그 가치의 합이  $k$ 원이 되도록 하고 싶다
- 그 경우의 수를 구하시오.
- 각각의 동전은 몇 개라도 사용할 수 있다.

# 동전 1

<https://www.acmicpc.net/problem/2293>

- 1, 2, 3 더하기 4와 같은 문제이다

# 동전 1

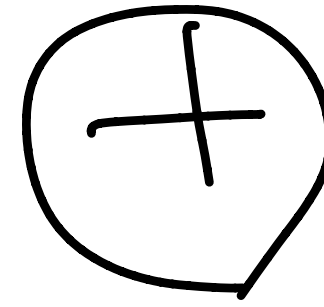
<https://www.acmicpc.net/problem/2293>

- 소스: <http://codeplus.codes/1926f9298a7447f1af724b0a81232a18>

# 동전 2

<https://www.acmicpc.net/problem/2294>

- n가지 종류의 동전이 있다
- 각각의 동전이 나타내는 가치는 다르다
- 이 동전들을 적당히 사용해서, 그 가치의 합이 k원이 되도록 하고 싶다
- 그러면서 동전의 개수가 최소가 되도록 하려고 한다
- 각각의 동전은 몇개라도 사용할 수 있다



min

# 동전 2

<https://www.acmicpc.net/problem/2294>

- 동전 1과 비슷한 방법으로 풀 수 있다
- $D[i]$  = i원을 만드는데 필요한 동전의 최소 개수

# 동전 2

<https://www.acmicpc.net/problem/2294>

- 소스: <http://codeplus.codes/e1183e8f1adf4312b079550212930667>

# 크리보드

<https://www.acmicpc.net/problem/11058>

$D[N]$  = 버튼을  $N$ 번  
화면의  $A$  최대 개수

8

1. 화면에  $A$ 를 출력한다.  $A$
2. Ctrl-A: 화면을 전체 선택한다
3. Ctrl-C: 전체 선택한 내용을 버퍼에 복사한다
4. Ctrl-V: 버퍼가 비어있지 않은 경우에는 화면에 출력된 문자열의 바로 뒤에 버퍼의 내용을 붙여넣는다.

$$D[N-1] + 1$$

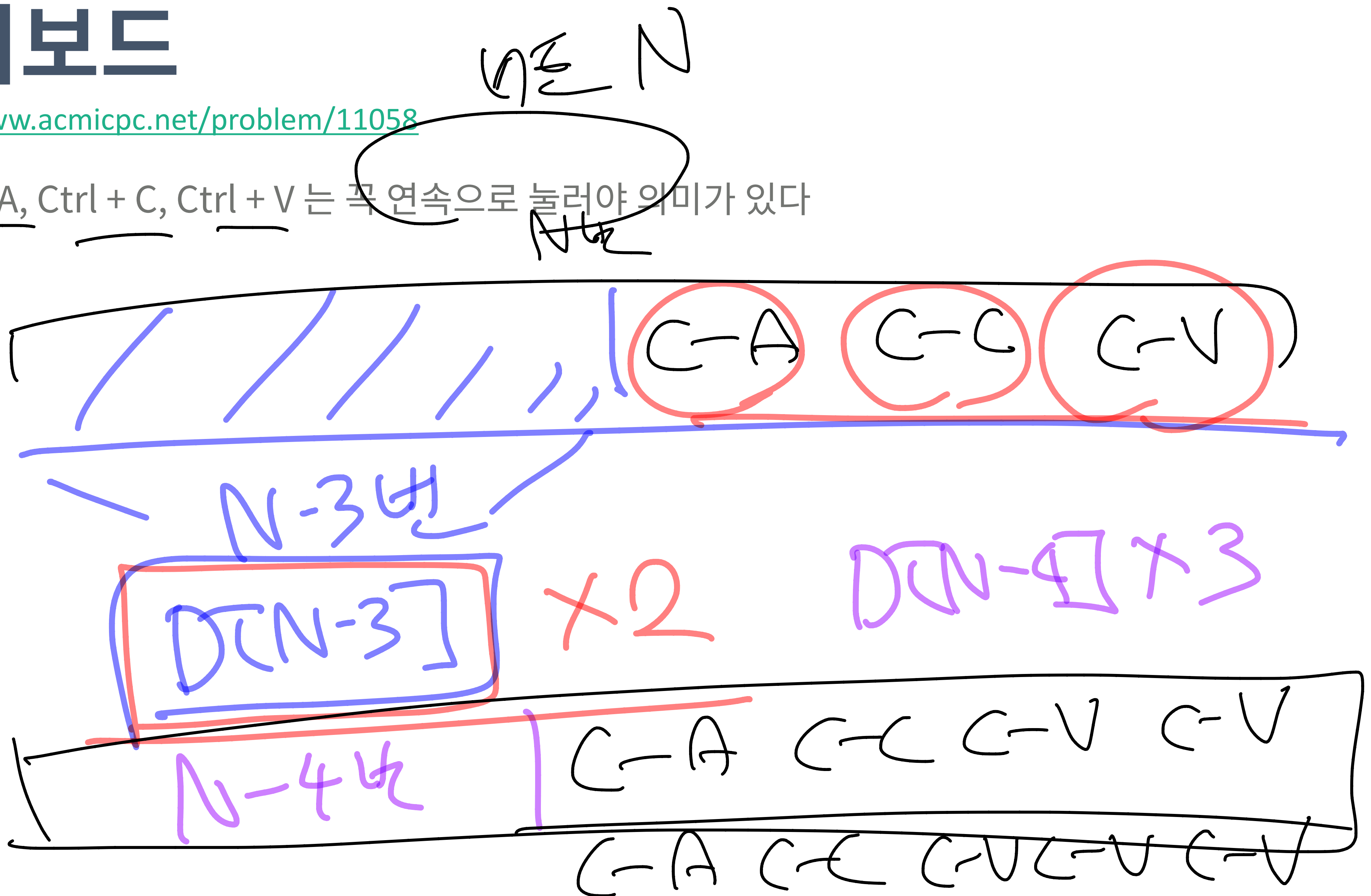
- 크리보드의 버튼을 총  $N$ 번 눌러서 화면에 출력된  $A$  개수를 최대화하는 프로그램을 작성하시오.



# 크리보드

<https://www.acmicpc.net/problem/11058>

- Ctrl + A, Ctrl + C, Ctrl + V 는 꼭 연속으로 눌러야 의미가 있다



# 크리보드

<https://www.acmicpc.net/problem/11058>

- $D[i]$  = 크리보드의 버튼을 총  $i$ 번 눌러서 화면에 출력된  $A$ 개수의 최대값
- 화면에  $A$ 를 출력하는 버튼을 누른 경우:
- 마지막에  $\text{Ctrl} + A$ ,  $\text{Ctrl} + C$ ,  $\text{Ctrl} + V$ 를 누른 경우:

# 크리보드

<https://www.acmicpc.net/problem/11058>

- $D[i]$  = 크리보드의 버튼을 총  $i$ 번 눌러서 화면에 출력된  $A$ 개수의 최대값
- 화면에  $A$ 를 출력하는 버튼을 누른 경우:  $D[i-1] + 1$
- 마지막에  $\text{Ctrl} + A$ ,  $\text{Ctrl} + C$ ,  $\text{Ctrl} + V$ 를 누른 경우:  $D[i-3] \times 2$

# 크리보드

<https://www.acmicpc.net/problem/11058>

- $D[i]$  = 크리보드의 버튼을 총  $i$ 번 눌러서 화면에 출력된  $A$ 개수의 최대값
- 화면에  $A$ 를 출력하는 버튼을 누른 경우:  $D[i-1] + 1$
- 마지막에 Ctrl + A, Ctrl + C, Ctrl + V를 누른 경우:  $D[i-3] \times 2$
- 마지막에 Ctrl + A, Ctrl + C, Ctrl + V, Ctrl + V를 누른 경우:  $D[i-4] \times 3$

# 크리보드

<https://www.acmicpc.net/problem/11058>

- $D[i]$  = 크리보드의 버튼을 총  $i$ 번 눌러서 화면에 출력된  $A$ 개수의 최대값
- 화면에  $A$ 를 출력하는 버튼을 누른 경우:  $D[i-1] + 1$
- 마지막에 Ctrl + A, Ctrl + C, Ctrl + V를 누른 경우:  $D[i-3] \times 2$
- 마지막에 Ctrl + A, Ctrl + C, Ctrl + V, Ctrl + V를 누른 경우:  $D[i-4] \times 3$
- 마지막에 Ctrl + A, Ctrl + C, Ctrl + V, Ctrl + V, Ctrl + V를 누른 경우:  $D[i-5] \times 4$

# 크리보드

<https://www.acmicpc.net/problem/11058>

14

•  $D[i]$  = 크리보드의 버튼을 총  $i$ 번 눌러서 화면에 출력된 A개수의 최대값

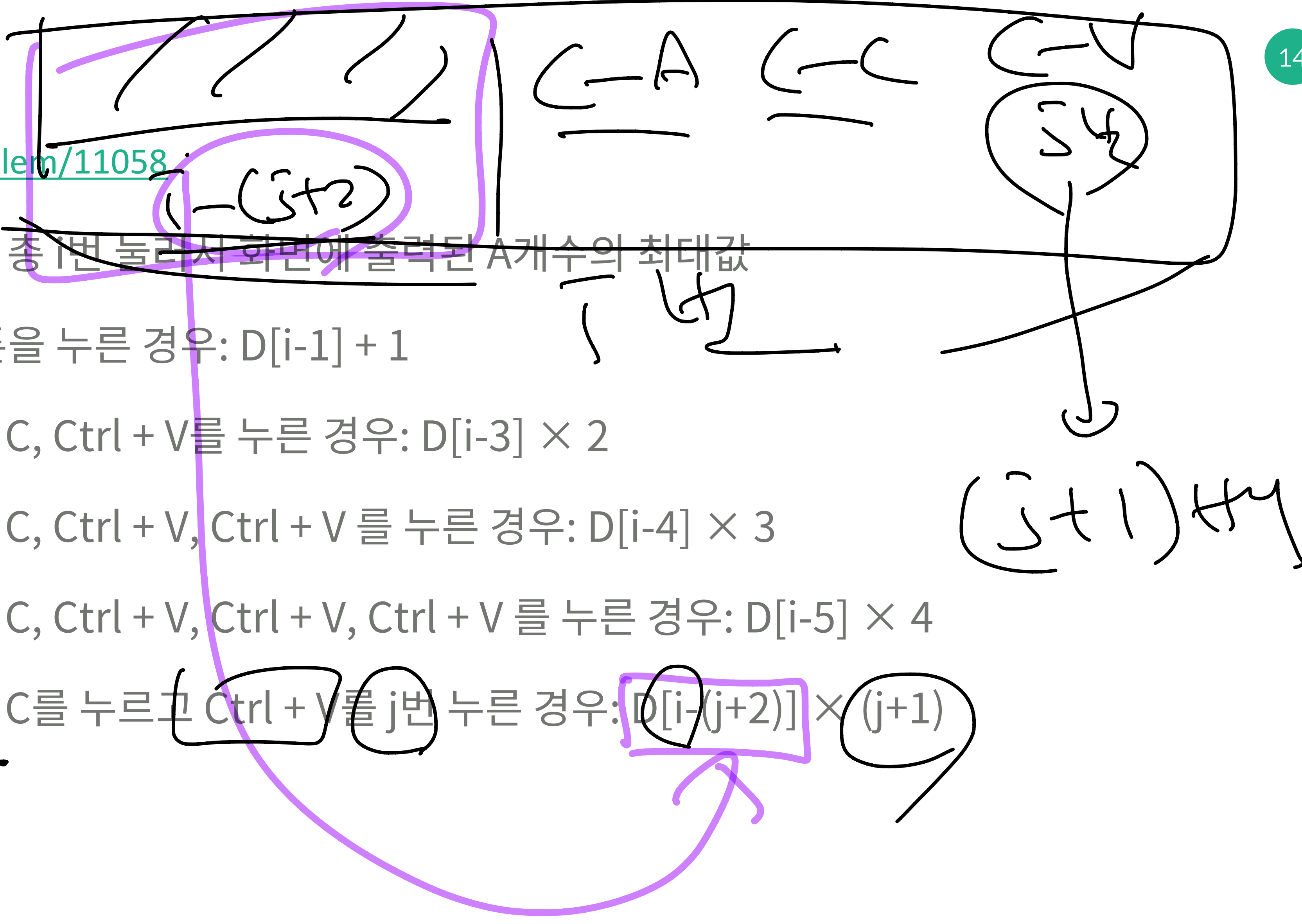
• 화면에 A를 출력하는 버튼을 누른 경우:  $D[i-1] + 1$

• 마지막에 Ctrl + A, Ctrl + C, Ctrl + V를 누른 경우:  $D[i-3] \times 2$

• 마지막에 Ctrl + A, Ctrl + C, Ctrl + V, Ctrl + V를 누른 경우:  $D[i-4] \times 3$

• 마지막에 Ctrl + A, Ctrl + C, Ctrl + V, Ctrl + V, Ctrl + V를 누른 경우:  $D[i-5] \times 4$

• 마지막에 Ctrl + A, Ctrl + C를 누르고 Ctrl + V를  $j$ 번 누른 경우:  $D[i-(j+2)] \times (j+1)$

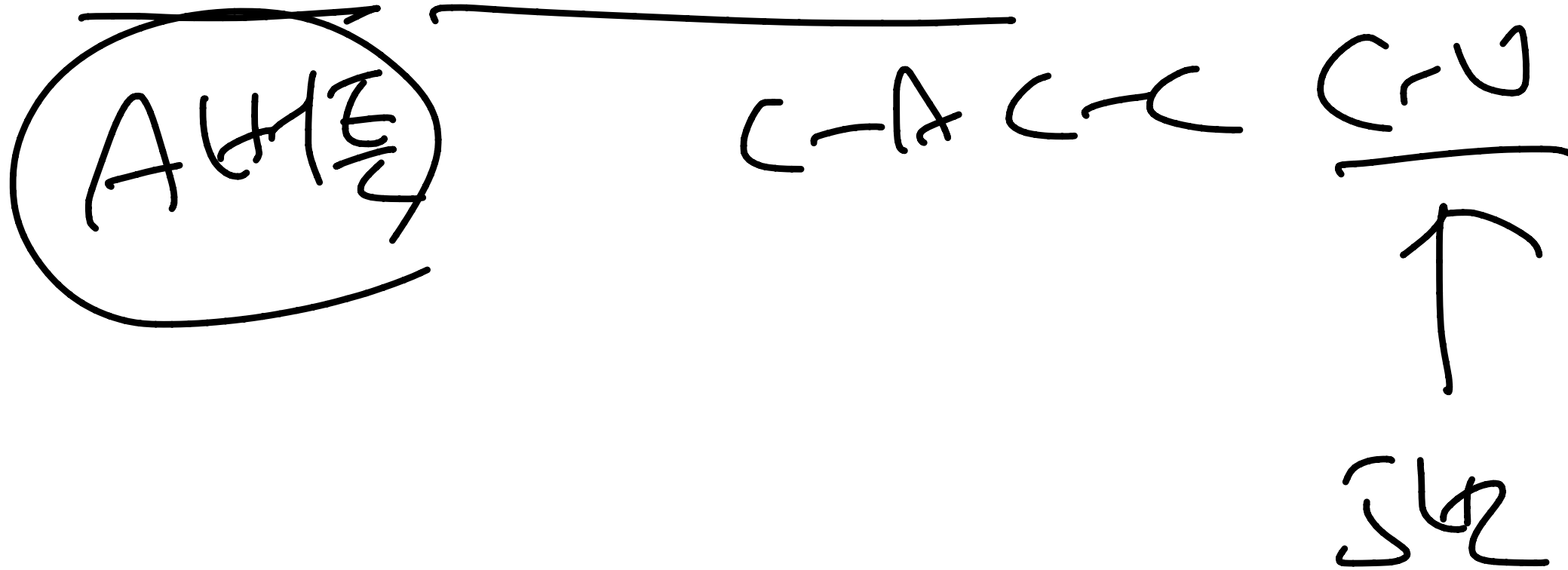


# 크리보드

<https://www.acmicpc.net/problem/11058>

- $D[i]$  = 크리보드의 버튼을 총  $i$ 번 눌러서 화면에 출력된  $A$ 개수의 최대값

- $D[i] = \max(D[i-1]+1, D[i-(j+2)] \times (j+1)) \quad (1 \leq j \leq i-3)$



# 크리보드

<https://www.acmicpc.net/problem/11058>

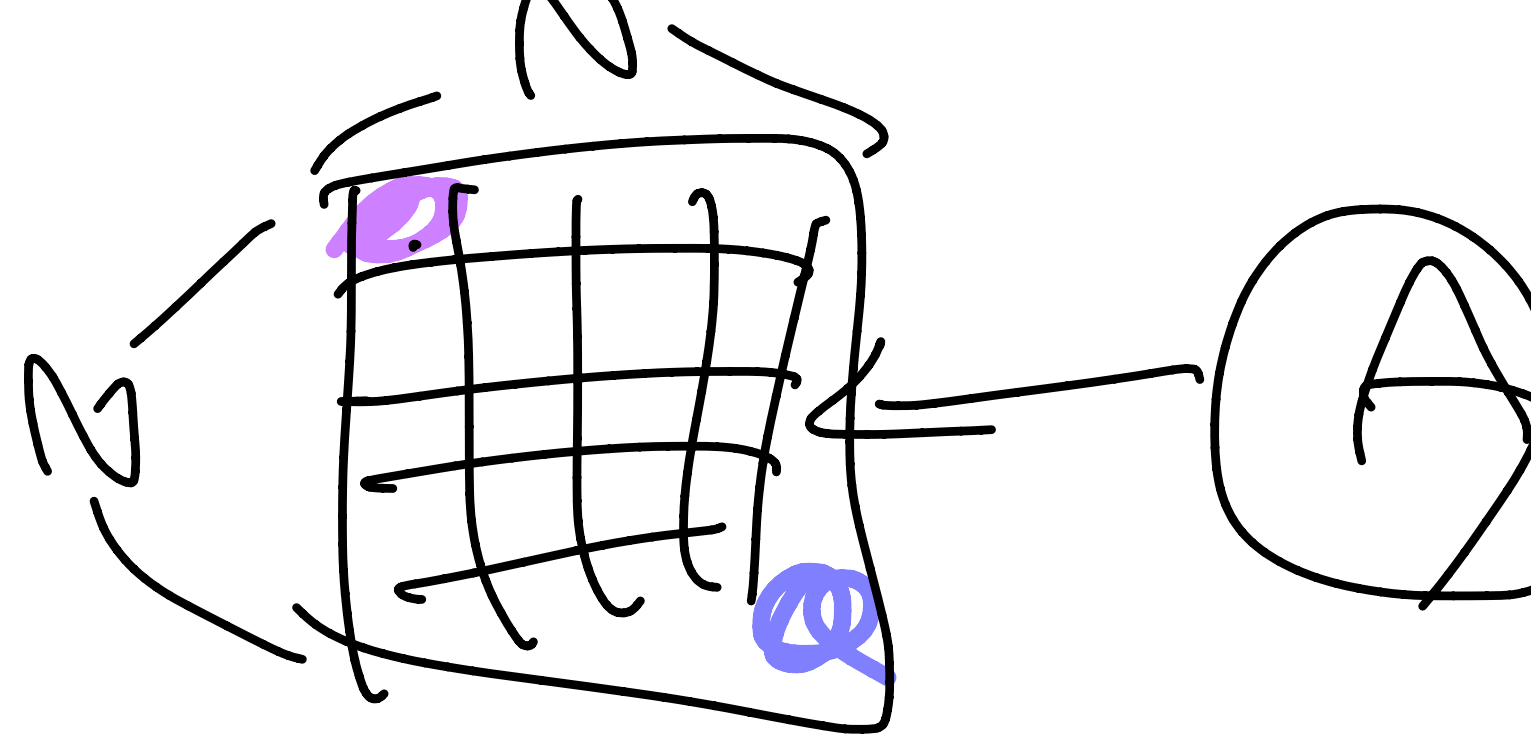
- 소스: <http://codeplus.codes/f73554f7f909421ea8e06daab610d290>



# 점프

<https://www.acmicpc.net/problem/1890>

17



- $N \times N$  게임판에 수가 적혀져 있음

• 게임의 목표는 가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 규칙에 맞게 점프를 해서 가는 것

• 각 칸에 적혀있는 수는 현재 칸에서 갈 수 있는 거리를 의미

• 반드시 오른쪽이나 아래쪽으로만 이동해야 함

• 0은 더 이상 진행을 막는 종착점이며 항상 현재 칸에 적혀있는 수만큼 오른쪽이나 아래로 가야 함

• 가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 규칙에 맞게 이동할 수 있는 경로의 개수를 구하는 문제

# 점프

$N^2 \times N$

$(1, 1)$

<https://www.acmicpc.net/problem/1890>

18

$D[i][j]$   $(i, j)$ 칸에 갈 수 있는 경로의 개수

$(i, j)$ 칸에 올 수 있는 칸을 찾아야 한다.

① + ②

$(1, 1)$

$\sim$

$\sim$

$\sim$

?

$\sim$

$(1, 5)$

①

$\rightarrow$

$D[i][k]$

$(k < j, j - k == A[i][k])$

②

$\downarrow$

$D[k][1]$

$(k < 1, 1 - k == A[k][1])$

$A[k][1]$

k의 수  $N^2$

# 점프

<https://www.acmicpc.net/problem/1890>

- $D[i][j]$  = (i, j)칸에 갈 수 있는 경로의 개수
- (i, j)칸에 올 수 있는 칸을 찾아야 한다.
- $D[i][j] += D[i][k] \text{ (} k+A[i][k] == j, 0 \leq k < j \text{)}$
- $D[i][j] += D[k][j] \text{ (} k+A[k][j] == i, 0 \leq k < i \text{)}$

# 점프

<https://www.acmicpc.net/problem/1890>

- $D[i][j]$  = (i, j)칸에 갈 수 있는 경로의 개수
- (i, j)칸에 올 수 있는 칸을 찾아야 한다.
- $D[i][j] += D[i][k] \text{ (} k+A[i][k] == j, 0 \leq k < j \text{)}$
- $D[i][j] += D[k][j] \text{ (} k+A[k][j] == i, 0 \leq k < i \text{)}$
- 한 칸을 채우는데 필요한 복잡도:  $O(N)$
- 총 시간 복잡도 :  $O(N^3)$

# 점프

<https://www.acmicpc.net/problem/1890>

- 소스: <http://codeplus.codes/d949f4fa18e74b8a881c2971fedde0f9>

# 점프

<https://www.acmicpc.net/problem/1890>

22

- $D[i][j]$  = (i, j)칸에 갈 수 있는 경로의 개수
- (i, j)에서 갈 수 있는 칸을 찾아야 한다.
- $D[i][j+A[i][j]] += D[i][j];$
- $D[i+A[i][j]][j] += D[i][j];$

$271.2$

$O(N^2)$



# 점프

<https://www.acmicpc.net/problem/1890>

- $D[i][j] = (i, j)$ 칸에 갈 수 있는 경로의 개수
- $(i, j)$ 에서 갈 수 있는 칸을 찾아야 한다.
- $D[i][j+A[i][j]] += D[i][j];$
- $D[i+A[i][j]][j] += D[i][j];$
- 한 칸을 채우는데 필요한 복잡도:  $O(1)$
- 총 시간 복잡도 :  $O(N^2)$

# 점프

<https://www.acmicpc.net/problem/1890>

- 소스: <http://codeplus.codes/5fec32d053d34663bb402554170c330c>



# 행렬 곱셈 순서

파일 형식)

25

<https://www.acmicpc.net/problem/11049>

- 크기가  $N \times M$ 인 행렬 A와  $M \times K$ 인 B를 곱할 때 필요한 곱셈 연산의 수는 총  $N \times M \times K$ 번
- 행렬 N개를 곱하는데 필요한 곱셈 연산의 수는 행렬을 곱하는 순서에 따라 다르다
- A의 크기가  $5 \times 3$ 이고, B의 크기가  $3 \times 2$ , C의 크기가  $2 \times 6$ 인 경우
- $(AB)C = 5 \times 3 \times 2 + 5 \times 2 \times 6 = 30 + 60 = 90$
- $A(BC) = 3 \times 2 \times 6 + 5 \times 3 \times 6 = 36 + 90 = 126$

$O(N^3)$

# 행렬 곱셈 순서

<https://www.acmicpc.net/problem/11049>

- $D[i][j]$  =  $i$ 번째 행렬부터  $j$ 번째 행렬까지 곱했을 때, 곱셈 연산의 최소값
- 행렬의 순서를 바꿀 수 없다

i		k		j
---	--	---	--	---

- $i$ 와  $j$  사이의 어딘가( $k$ )에서 행렬을 나눠서 곱셈을 해야 한다
- $(i \sim k \text{까지 곱한 행렬}) \times (k+1 \sim j \text{까지 곱한 행렬})$
- $D[i][k] + D[k+1][j] + \text{행렬 곱셈에서 필요한 연산 횟수}$

# 행렬 곱셈 순서

<https://www.acmicpc.net/problem/11049>

- $D[i][j]$  = i번째 행렬부터 j번째 행렬까지 곱했을 때, 곱셈 연산의 최소값
- $A[i]$  = i번째 행렬의 크기 ( $A[i][0] \times A[i][1]$ )
- $D[i][j] = \text{Min}(D[i][k] + D[k+1][j] + A[i][0] * A[k][1] * A[j][1])$

# 행렬 곱셈 순서

<https://www.acmicpc.net/problem/11049>

28

```
int go(int x, int y) {  
    if (d[x][y]) return d[x][y];  
    if (x == y) return 0;  
    if (x+1 == y) return a[x][0]*a[x][1]*a[y][1];  
    int &ans = d[x][y];  
    ans = -1;  
    for (int k=x; k<=y-1; k++) {  
        int t1 = go(x,k); int t2 = go(k+1,y);  
        if (ans == -1 || ans > t1+t2+a[x][0]*a[k][1]*a[y][1]) {  
            ans = t1+t2+a[x][0]*a[k][1]*a[y][1];  
        }  
    }  
    return ans;  
}
```

The diagram illustrates the recursive call for matrix multiplication order. A red box labeled 'x' and a blue box labeled 'y' are shown. A red arrow points from 'x' to the first 'a[x][0]' in the code. A blue arrow points from 'y' to the last 'a[y][1]' in the code. A black arrow points from the 'return 0' line to the 'if (x == y)' condition. A black arrow points from the 'return a[x][0]\*a[x][1]\*a[y][1]' line to the 'if (x+1 == y)' condition. A black arrow points from the 'return ans;' line to the 'return' statement.

# 행렬 곱셈 순서

<https://www.acmicpc.net/problem/11049>

- 소스: <http://codeplus.codes/eddcfd73120f45dc849d24dccc21302>

# 1학년

수 N개

기타리즈

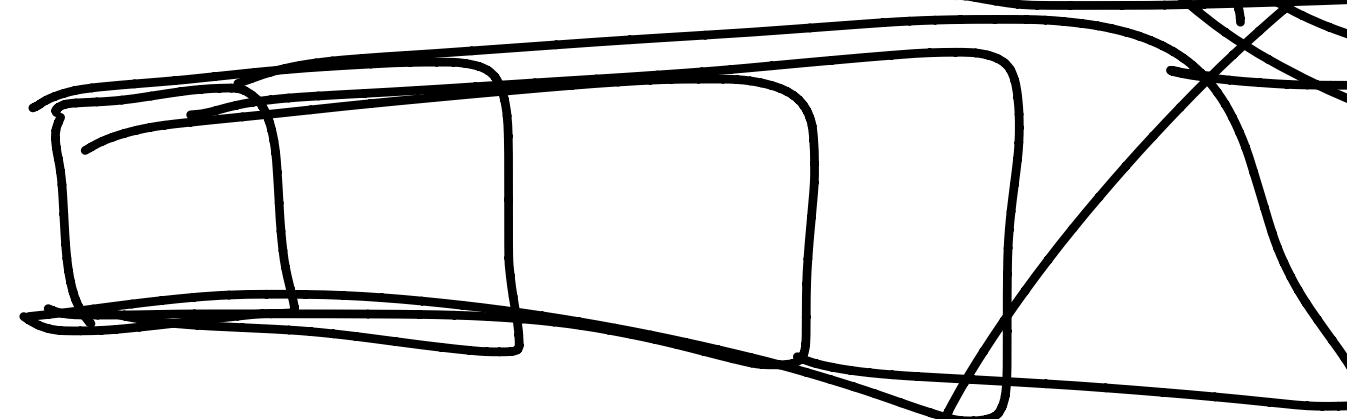
30

<https://www.acmicpc.net/problem/5557>

- 마지막 두 숫자 사이에 '='을 넣고, 나머지 숫자 사이에는 '+' 또는 '-'를 넣어 등식을 만든다
- 예를 들어, "8 3 2 4 8 7 2 4 0 8 8"에서 등식 "8+3-2-4+8-7-2-4-0+8=8"을 만들 수 있다



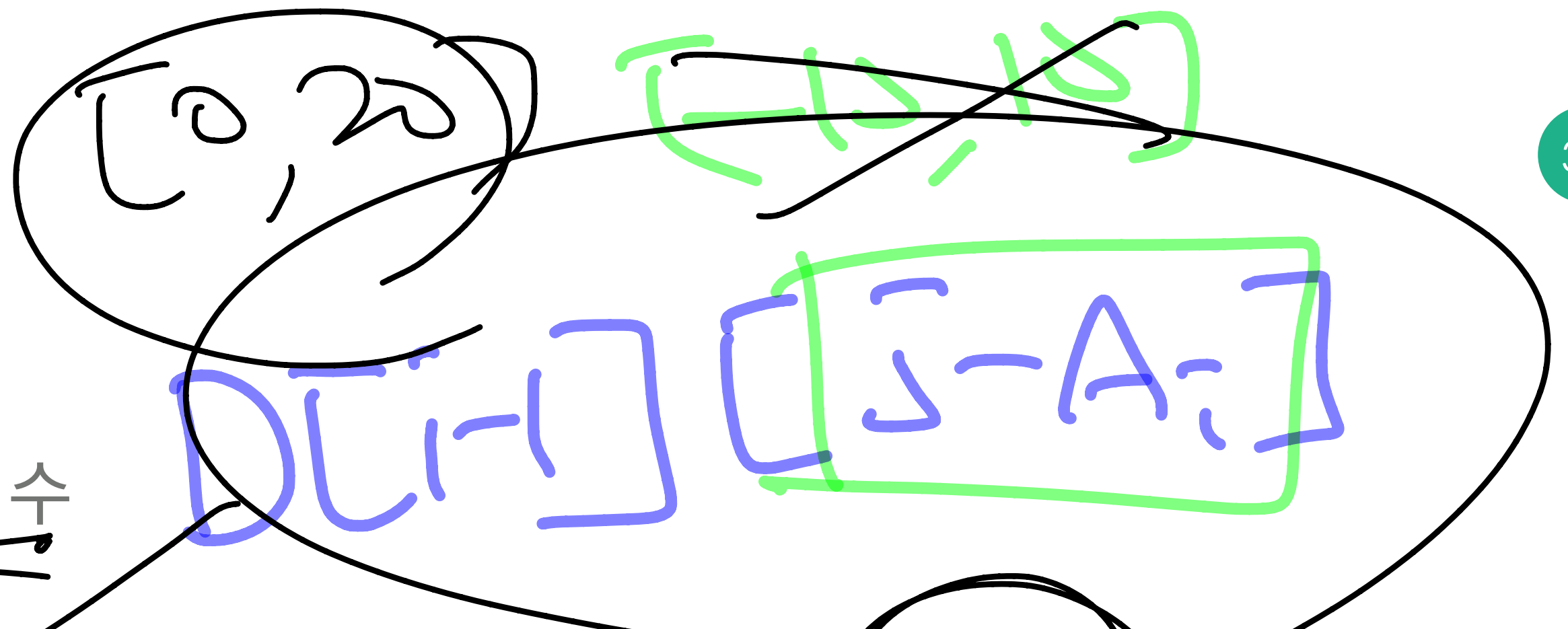
수의 개수



# 1학년

<https://www.acmicpc.net/problem/5557>

•  $D[i][j]$  =  $i$ 까지 수를 사용해서  $j$ 를 만드는 방법의 수



$A_1 \pm A_2 \dots \pm A_{i-1} \pm A_i = j$

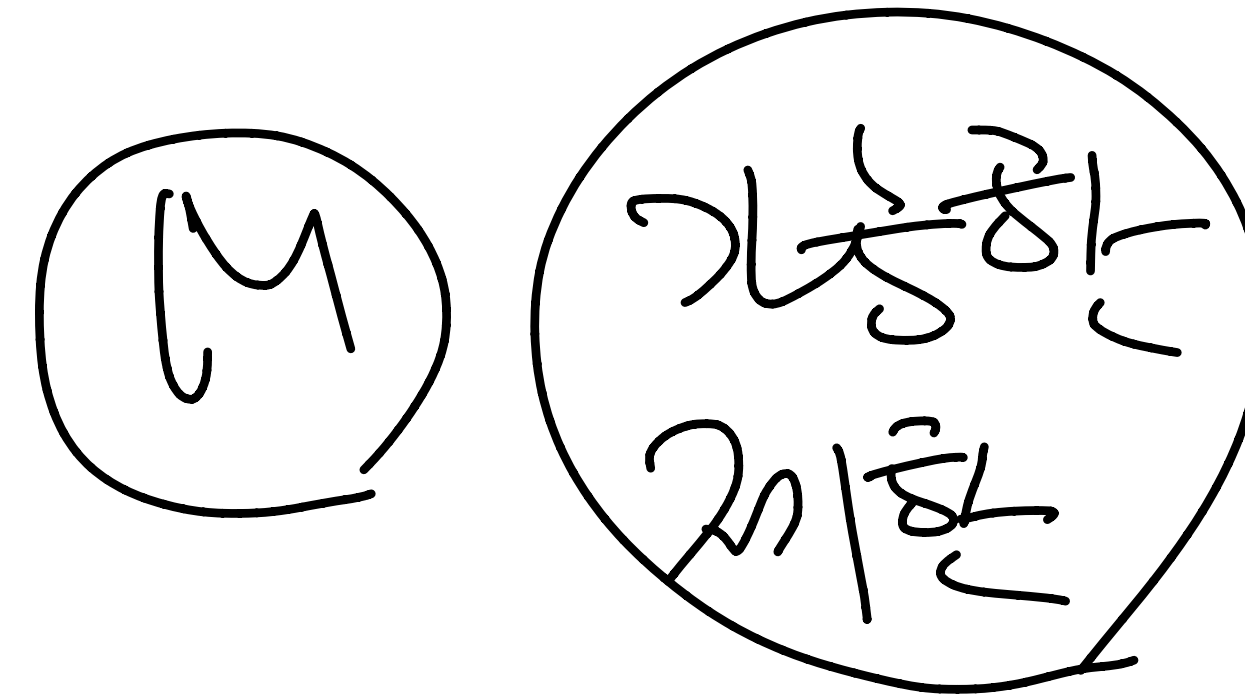
$A_1 \pm A_2 \dots \pm A_{i-1} = A_i = j$



# 1학년

<https://www.acmicpc.net/problem/5557>

- $D[i][j] = D[i-1][j-A[i]] + D[i-1][j+A[i]]$



$O(2N)$

$O(2^N)$  또는  $(N^2)$



# 1학년

<https://www.acmicpc.net/problem/5557>

- 소스: <http://codeplus.codes/1b015a9b941742e282a5361039bd11e3>

# ABC

<https://www.acmicpc.net/problem/12969>

- N과 K가 주어졌을 때, 아래 두 조건을 만족하는 문자열 S를 아무거나 찾는 문제
- $3 \leq N \leq 30, 0 \leq K \leq N(N-1)/2$
- 문자열 S의 길이는 N이고, A, B, C로만 이루어져 있다
- $0 \leq i < j < N$ 이고,  $S[i] < S[j]$ 를 만족하는  $(i, j)$ 쌍이 K개 있다

A < B

A < C

B < C

# ABC

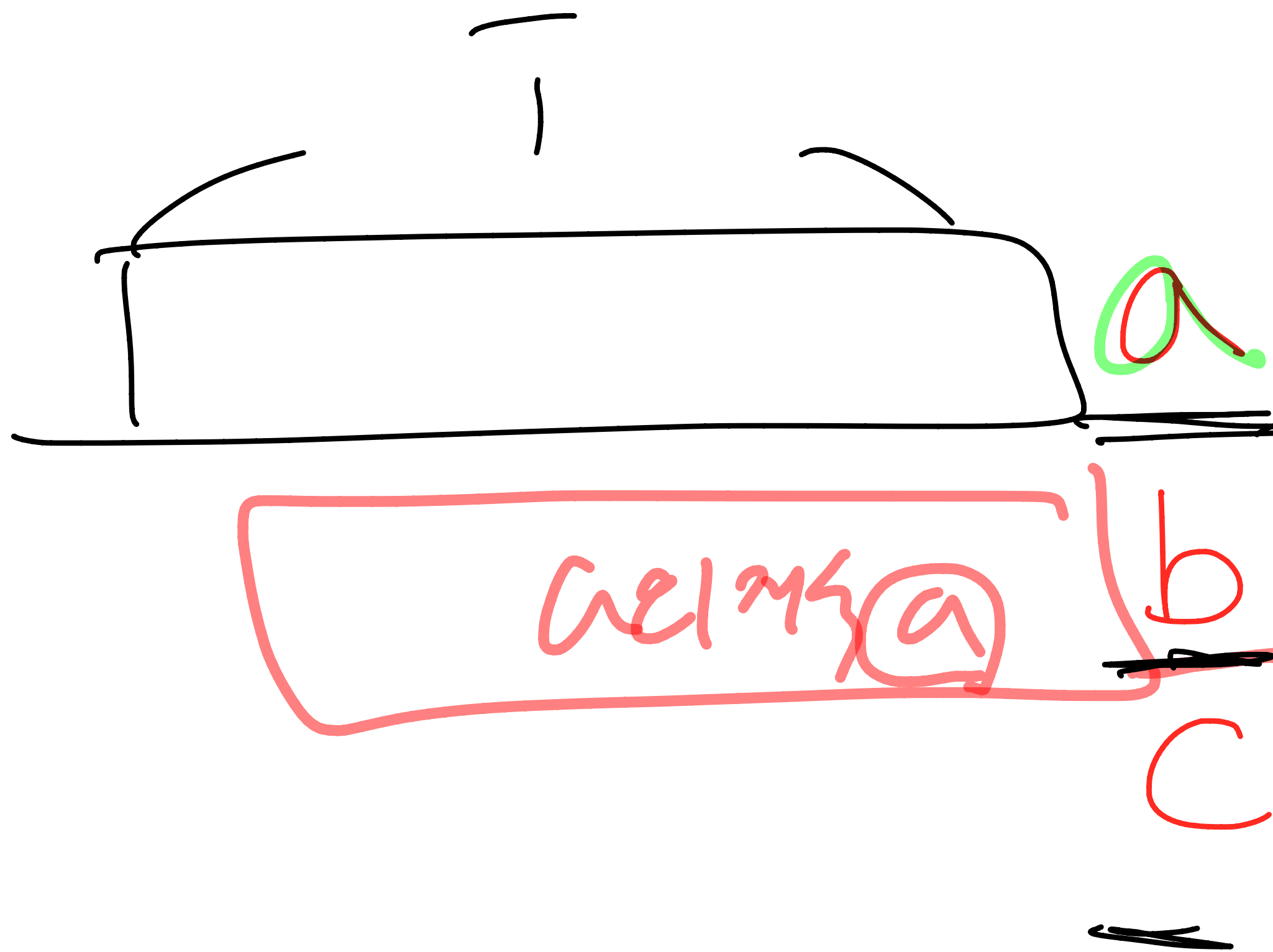
<https://www.acmicpc.net/problem/12969>

35



$$C = 1 - a - b$$

- $D[i][a][b][p]$  = 길이가  $i$ 이고, A의 개수가  $a$ 개, B의 개수가  $b$ 개,  $S[i] < S[j]$  쌍이  $p$ 개 있는 문자열이 가능한가?



$$D[i+1][a+1][b][p]$$

$$D[i+1][a][b+1][p+a]$$

$$D[i+1][a][b][p+a+b]$$

# ABC

<https://www.acmicpc.net/problem/12969>

- $D[i][a][b][p]$  = 길이가  $i$ 이고, A의 개수가  $a$ 개, B의 개수가  $b$ 개,  $S[i] < S[j]$  쌍이  $p$ 개 있는 문자열이 가능한가?
- $i$ 번째 글자가 A인 경우
- $i$ 번째 글자가 B인 경우
- $i$ 번째 글자가 C인 경우

# ABC

<https://www.acmicpc.net/problem/12969>

- $D[i][a][b][p]$  = 길이가  $i$ 이고, A의 개수가  $a$ 개, B의 개수가  $b$ 개,  $S[i] < S[j]$  쌍이  $p$ 개 있는 문자열이 가능한가?
- $i$ 번째 글자가 A인 경우:  $D[i+1][a+1][b][p]$
- $i$ 번째 글자가 B인 경우:  $D[i+1][a][b+1][p+a]$
- $i$ 번째 글자가 C인 경우:  $D[i+1][a][b][p+a+b]$

# ABC

<https://www.acmicpc.net/problem/12969>

- 소스: <http://codeplus.codes/481ee4b01ea14b6592bb94119a40534c>

# 출근 기록

<https://www.acmicpc.net/problem/14238>

39

- A: 아무 때나 가능

- B: 출근을 하면 다음날 쉬어야 함

- C: 출근한 다음날과 다다음날 쉬어야 함

- 출근 기록 S의 모든 순열 중에서 올바른 출근 기록인 것 아무거나 찾는 문제

- S의 길이  $\leq 50$

A B C

A A A A A B B A A

B X

C X X X

X X C

X B X B

# 출근 기록

<https://www.acmicpc.net/problem/14238>

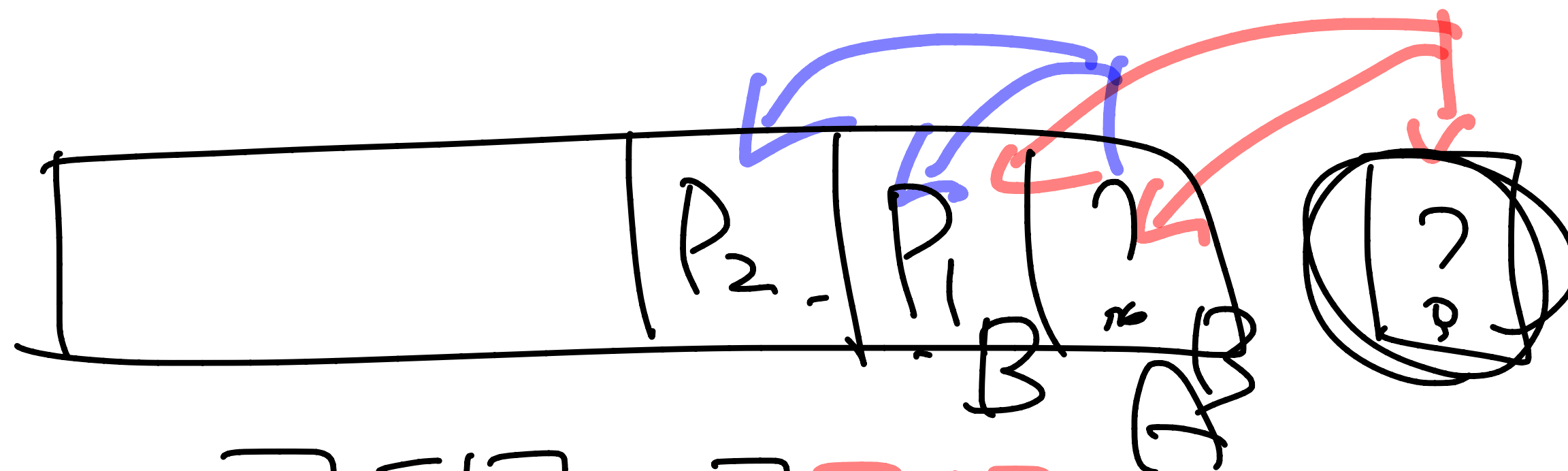
- 출근 기록에서 중요한 정보는 A의 개수(a), B의 개수(b), C의 개수 (c)
- B와 C때문에, 전날과 전전날 누가 일했는지가 필요하다.



# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$  = A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이  $p1$ , 그 전날 일한 사람이  $p2$ 인 것이 가능한가?



A가 출근:  $D[a+1][b][c][A][P1]$

B가 출근:  $D[a][b+1][c][B][P1]$

3번:  $P1 \neq B$

C가 출근:  $D[a][b][c+1][C][P1]$

$P1 \neq C$   
 $P2 \neq C$

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가  $(a, b, c)$  이고, 전날 일한 사람이  $p1$ , 그 전날 일한 사람이  $p2$ 인 것이 가능한가?
- 오늘 일한 사람이 A인 경우
- 오늘 일한 사람이 B인 경우
- 오늘 일한 사람이 C인 경우

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가  $(a, b, c)$  이고, 전날 일한 사람이  $p1$ , 그 전날 일한 사람이  $p2$ 인 것이 가능한가?
- 오늘 일한 사람이 A인 경우:  $D[a+1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우:  $D[a][b+1][c][1][p1]$
- 오늘 일한 사람이 C인 경우:  $D[a][b][c+1][2][p1]$

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2] = A, B, C$ 의 개수가  $(a, b, c)$  이고, 전날 일한 사람이  $p1$ , 그 전날 일한 사람이  $p2$ 인 것이 가능한가?
- 오늘 일한 사람이 A인 경우:  $D[a+1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우:  $D[a][b+1][c][1][p1]$  ( $p1 \neq 1$ )
- 오늘 일한 사람이 C인 경우:  $D[a][b][c+1][2][p1]$  ( $p1 \neq 2 \ \&\& \ p2 \neq 2$ )

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- 소스: <http://codeplus.codes/819c2b4d41ec49c1823ca9e09c3f15a2>

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$  = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$  = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?
- 오늘 일한 사람이 A인 경우
- 오늘 일한 사람이 B인 경우
- 오늘 일한 사람이 C인 경우

# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$  = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?
- 오늘 일한 사람이 A인 경우:  $D[a-1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우:  $D[a][b-1][c][1][p1]$
- 오늘 일한 사람이 C인 경우:  $D[a][b][c-1][2][p1]$



# 출근 기록

<https://www.acmicpc.net/problem/14238>

- $D[a][b][c][p1][p2]$  = 앞으로 추가해야 하는 A, B, C의 개수가 (a, b, c) 이고, 전날 일한 사람이 p1, 그 전날 일한 사람이 p2인 것이 가능한가?
- 오늘 일한 사람이 A인 경우:  $D[a-1][b][c][0][p1]$
- 오늘 일한 사람이 B인 경우:  $D[a][b-1][c][1][p1]$  ( $p1 \neq 1$ )
- 오늘 일한 사람이 C인 경우:  $D[a][b][c-1][2][p1]$  ( $p1 \neq 2 \ \&\& \ p2 \neq 2$ )

# 출근 기록

50

<https://www.acmicpc.net/problem/14238>

- 소스: <http://codeplus.codes/4795f662eaa94c7b95e12e8aa499ce9e>

# BOJ 거리

<https://www.acmicpc.net/problem/12026>

- 스타트는 BOJ를 외치면서 링크를 만나러 가려고 한다
- 따라서, 스타트는 B, O, J, B, O, J, B, O, J, ... 순서로 보도블럭을 밟으면서 점프를 할 것이다
- 스타트가 링크를 만나는데 필요한 에너지 양의 최소값을 구하는 프로그램을 작성하시오.

BOJ

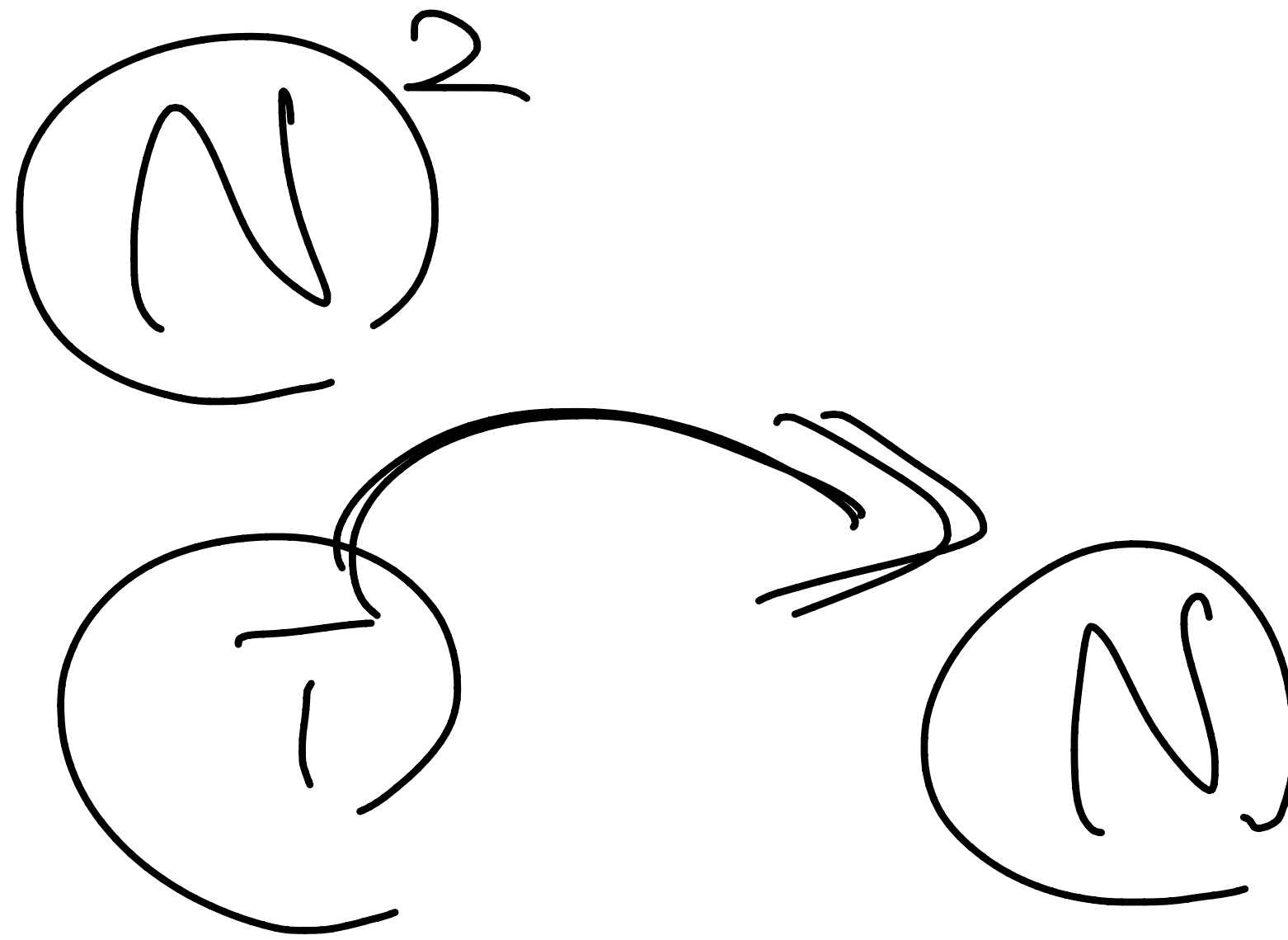


$$k^2$$

# BOJ 거리

<https://www.acmicpc.net/problem/12026>

- $D[N] = N$ 에 오는 최소 에너지



$$= \min(D[1] + (N - 1)^2)$$

$$\begin{array}{rcl} A[1] = 1 & A[N] = B & \\ \hline & 0 & \end{array}$$

$$\begin{array}{rcl} B & & \\ \hline 0 & & 11 \\ \hline \end{array}$$

# BOJ 거리

<https://www.acmicpc.net/problem/12026>

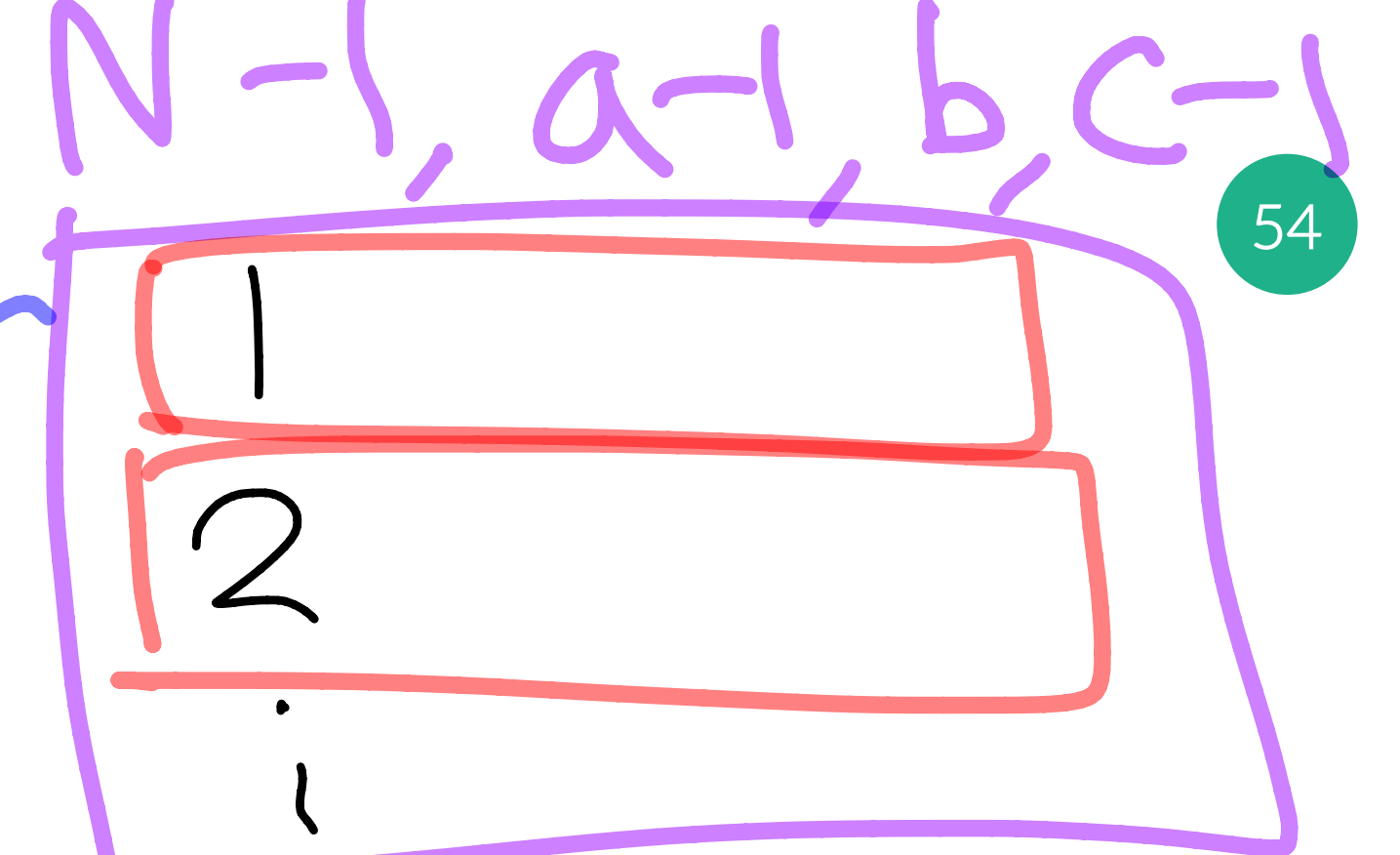
- 소스: <http://codeplus.codes/dbe2712ee56c4d2381d3ad479421f1d4>

# Acka A, B, C

<https://www.acmicpc.net/problem/12996>

$N$

$a, b, c$   
3개

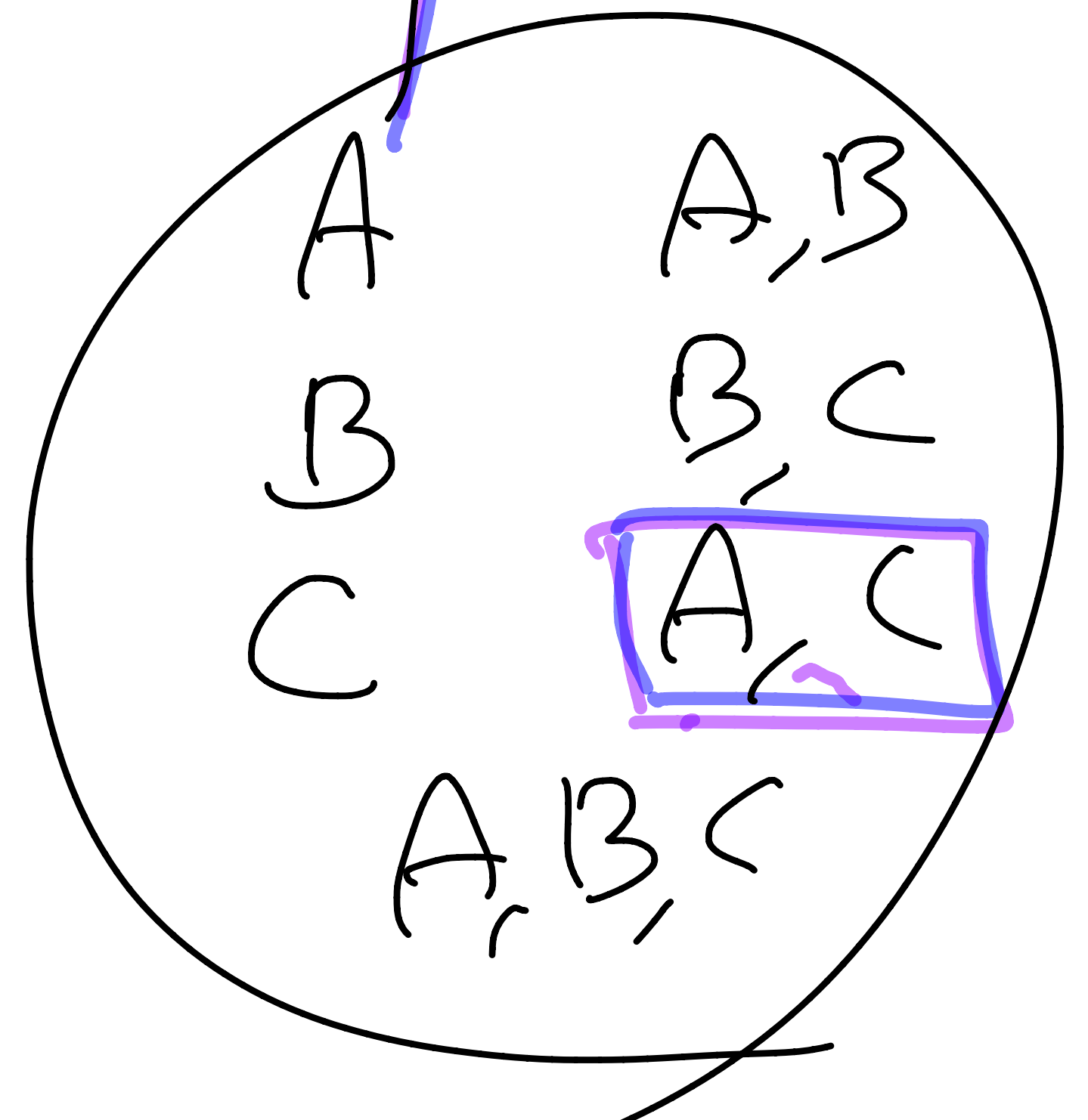


54

- 총  $N$ 개의 곡이 포함되어 있는 음반을 만들어야 한다.
- 음반에 포함되어 있는 모든 곡은 세 사람 중 적어도 한 명이 부른 곡이다.
- 두 음반 A와 B가 있을 때, 참여한 사람이 다른 곡이 존재한다면, 두 음반은 다른 음반이다.
- 음반을 만들 수 있는 방법의 수를 구하는 문제

$N$ 가지  
 $a, b, c$   
 $0, 0, 0$   
 ~~$1, 1, 1$~~   
 $2 \times 2 \times 3$

$8$ 가지  
 $-1$   
 $7$ 가지



# Acka

<https://www.acmicpc.net/problem/12996>

- $D[N][a][b][c]$  = 곡의 수  $N$ 개, 세 사람이 부른 곡의 수가  $a, b, c$ 개 일 때, 경우의 수

# Acka

<https://www.acmicpc.net/problem/12996>

- $D[N][a][b][c] = \sum D[N-1][a-i][b-j][c-k]$  ( $i, j, k$ 는 세 사람의 참가 여부)

1 2 3  
0 0 0  
/ / /



# Acka

<https://www.acmicpc.net/problem/12996>

- $D[N][a][b][c] = \sum D[N-1][a-i][b-j][c-k]$  (i, j, k는 세 사람의 참가 여부)

# Acka

<https://www.acmicpc.net/problem/12996>

- 소스: <http://codeplus.codes/d179509569354833b1d5e60742c5a6c5>

# 데스노트

<https://www.acmicpc.net/problem/2281>

59

$N$ 개의  $\frac{N^2}{2}$

높이  $M$ 칸

넓이  $M$ 칸

등차

$\frac{7 \times 10}{2}$

7 4 2 3

2 5 1 12

7 5 6

*	*	*	*	*	*	*		*	*	*	*		*	*		*	*	*
*	*		*	*	*	*	*		*									
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*		*	*	*	*	*	*							

*	*	*	*	*	*	*		*	*	*	*		*	*				
*	*	*		*	*		*	*	*	*	*		*					
*	*	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*
*	*	*	*	*		*	*	*	*	*	*							

높이  $M$ 칸

가운데  $M$ 칸

가운데  $M$ 칸

$\frac{2 \times 1}{2}$

$$10^2 + 1^2 = 101$$

$$5^2 + 6^2 = 25 + 36 = 61$$

# 데스노트

<https://www.acmicpc.net/problem/2281>

- D[index][count] = index를 썼고, 그 줄에 총 count개의 글자를 썼을 때



# 데스노트

61

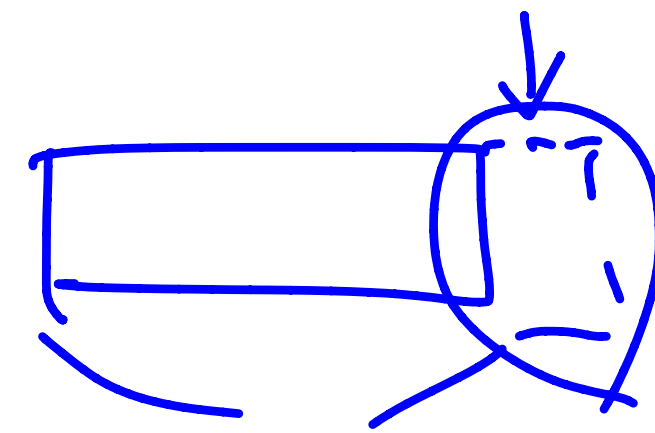
<https://www.acmicpc.net/problem/2281>

•  $D[index][count] = index$ 를 ~~보고~~, 그 줄에 총  $count$ 개의 글자를 썼을 때

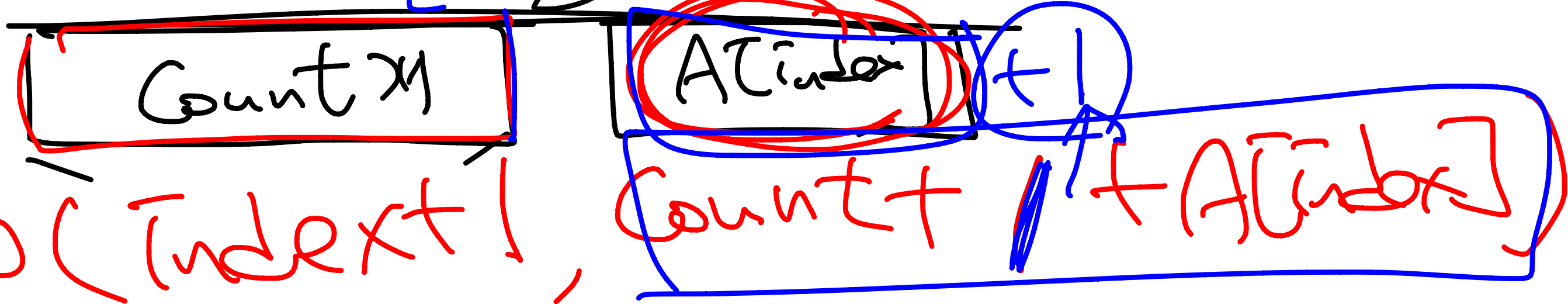
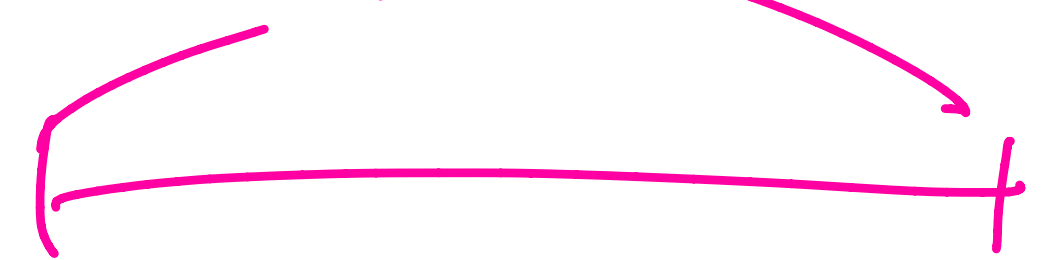
• 다음 줄로 넘어가는 경우

• 현재 줄에 이어서 쓰는 경우

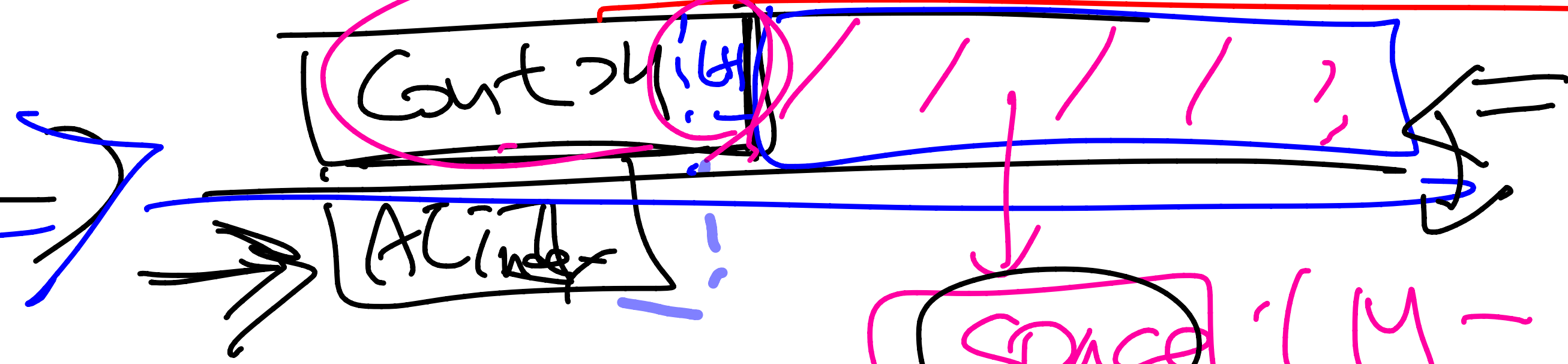
남는 것 > 0



M



$go(Index + 1, Count + 1 + A[Index])$



$Space * (M - Count + 1)$

$go(Index + 1, A[Index] + 1) + Space * Space$

# 데스노트

<https://www.acmicpc.net/problem/2281>

- $D[index][count]$  = index를 썼고, 그 줄에 총 count개의 글자를 썼을 때
- 다음 줄로 넘어가는 경우

```
int space = (m - (cnt - 1));  
int cost = space * space;  
ans = go(index + 1, a[index] + 1) + cost;
```

- 현재 줄에 이어서 쓰는 경우

```
if (cnt + a[index] <= m) {  
    int cur = go(index + 1, cnt + a[index] + 1);  
    if (ans > cur) ans = cur;  
}
```

# 데스노트

<https://www.acmicpc.net/problem/2281>

- 소스: <http://codeplus.codes/166910de252643538dfb6932eb4d3bff>

# 올바른 괄호 문자열

②

<https://www.acmicpc.net/problem/3012>

- 만들 수 있는 괄호 문자열의 개수를 구하는 문제

- $(?([?])?}? \rightarrow 3$ 개

( [ {  
) ] }

?

$D[i][j] : 1 \sim j$  문자열

괄호의 개수

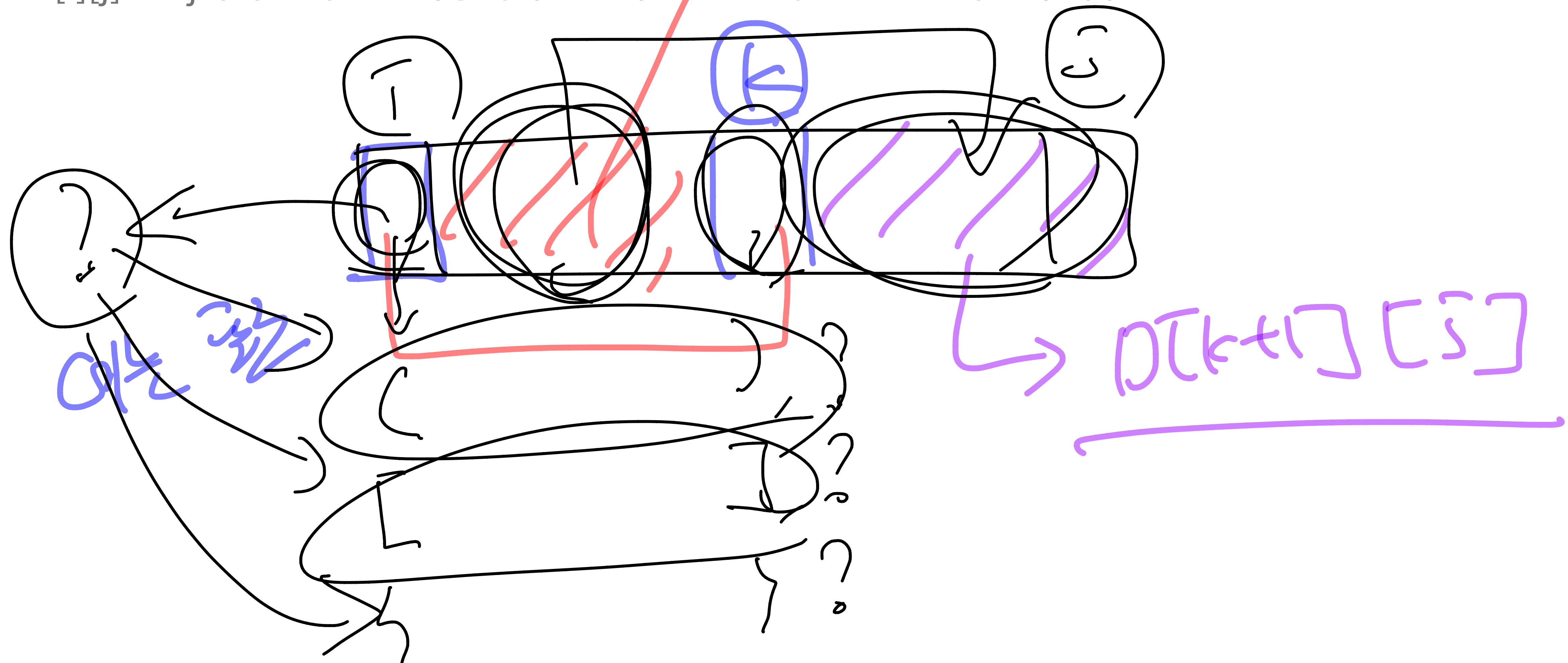


# 올바른 괄호 문자열

65

<https://www.acmicpc.net/problem/3012>

- $D[i][j]$  =  $i \sim j$ 까지 문자열을 이용해서 만들 수 있는 올바른 괄호 문자열의 개수



# 올바른 괄호 문자열

<https://www.acmicpc.net/problem/3012>

- $D[i][j]$  =  $i \sim j$ 까지 문자열을 이용해서 만들 수 있는 올바른 괄호 문자열의 개수
- $i$ 번째에 있는 왼쪽 괄호와 짝이 맞는 오른쪽 괄호의 위치를  $k$  라고 했을 때,
- $(i+1, k-1)$ 와  $(k+1, j)$ 로 나눌 수 있다.

# 올바른 괄호 문자열

67

<https://www.acmicpc.net/problem/3012>

- $D[i][j]$  =  $i \sim j$ 까지 문자열을 이용해서 만들 수 있는 올바른 괄호 문자열의 개수
- $D[i][j] += D[i+1][k-1] * D[k+1][j]$

# 올바른 괄호 문자열

68

<https://www.acmicpc.net/problem/3012>

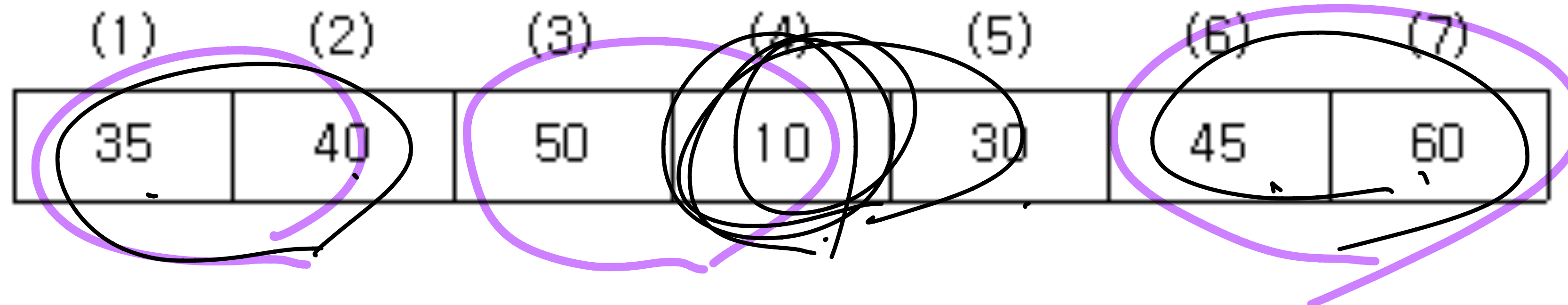
- 소스: <http://codeplus.codes/4dc43ee410194620a483d6ca2f5829e5>

# 소형기관차

<https://www.acmicpc.net/problem/2616>

69

- $N (\leq 50,000)$ 개의 객차가 있고, 각 객차에 타고 있는 사람의 수  $A[i]$  명
- 3개의 소형 기관차가 최대로 끌 수 있는 객차의 수는 같다
- 소형 기관차 3대를 이용하여 최대한 많은 손님을 목적지까지 운송하도록 한다
- 각 소형 기관차는 번호가 연속적으로 이어진 객차를 끌게 한다.



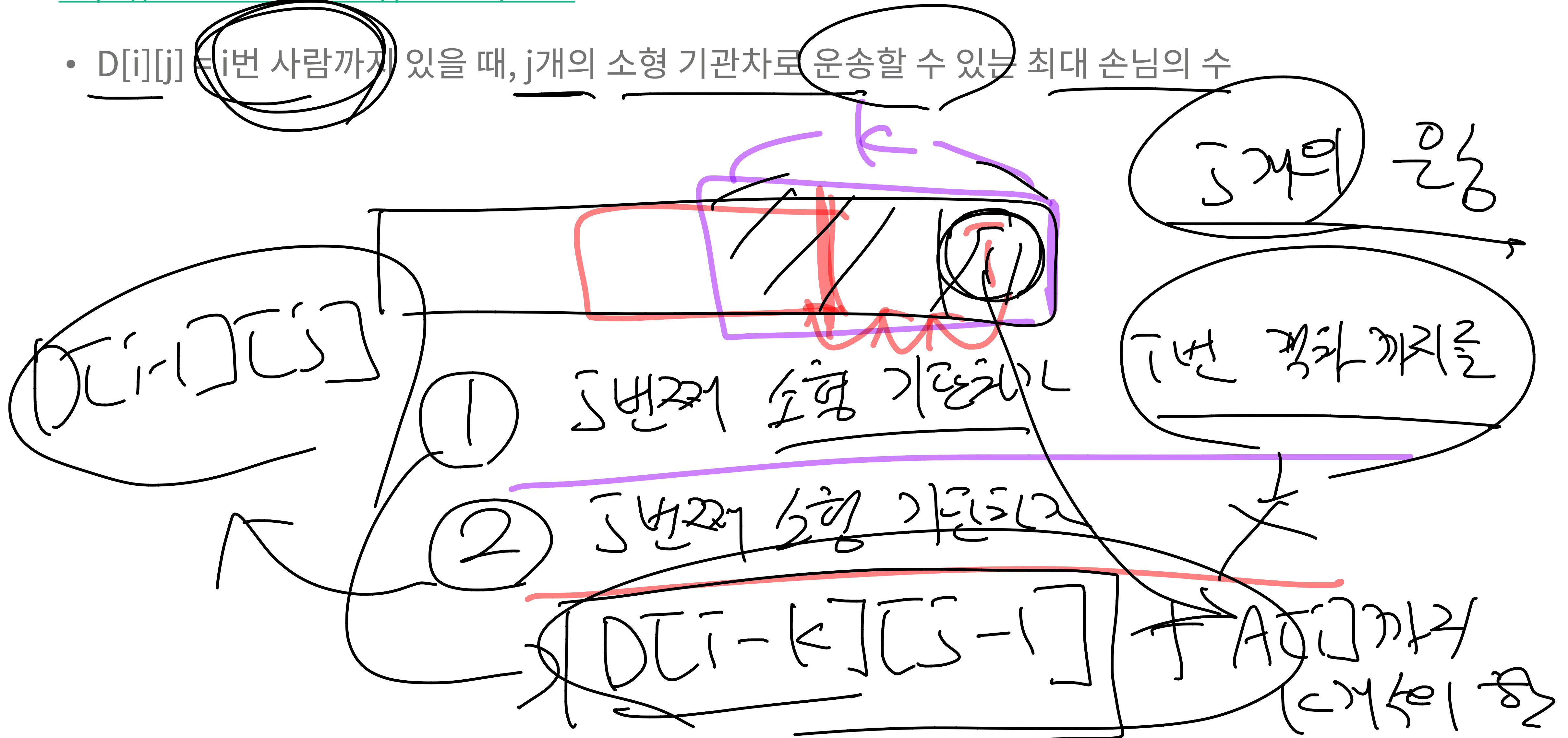
# 소형기관차

100%

70

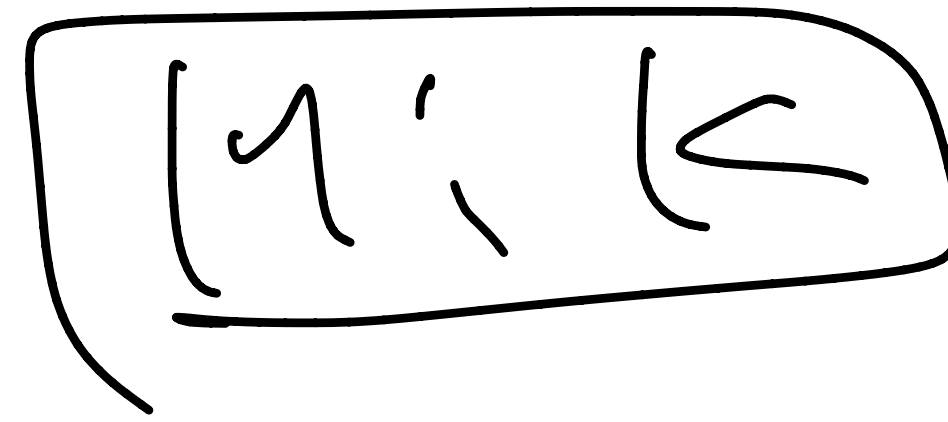
<https://www.acmicpc.net/problem/2616>

- $D[i][j]$  =  $i$ 번 사람까지 있을 때,  $j$ 개의 소형 기관차로 운송할 수 있는 최대 손님 수



# 소형기관차

<https://www.acmicpc.net/problem/2616>



- $D[i][j] = \max(D[i-1][j], D[i-m][j-1] + (A[i-m+1] \sim A[i] \text{ 까지 합}))$

# 소형기관차

<https://www.acmicpc.net/problem/2616>

- 소스: <http://codeplus.codes/564960f6993249959b5ee6b1c39cef06>



# 박스 안의 열쇠

73

<https://www.acmicpc.net/problem/1413>

- 사이클의 개수가 M개를 넘지 않는 길이가 N인 순열의 개수를 구하는 문제

# 박스 안의 열쇠

<https://www.acmicpc.net/problem/1413>

- 사이클의 개수가 M개를 넘지 않는 길이가 N인 순열의 개수를 구하는 문제
- 이 수는 제 1종 스털링 수 (Stirling numbers of the first kind) 와 같다

# 박스 안의 열쇠

<https://www.acmicpc.net/problem/1413>

- 사이클의 개수가 M개를 넘지 않는 길이가 N인 순열의 개수를 구하는 문제
- 이 수는 제 1종 스털링 수 (Stirling numbers of the first kind) 와 같다

$$D[i][j] = D[i-1][j-1] + (i-1) * D[i-1][j]$$

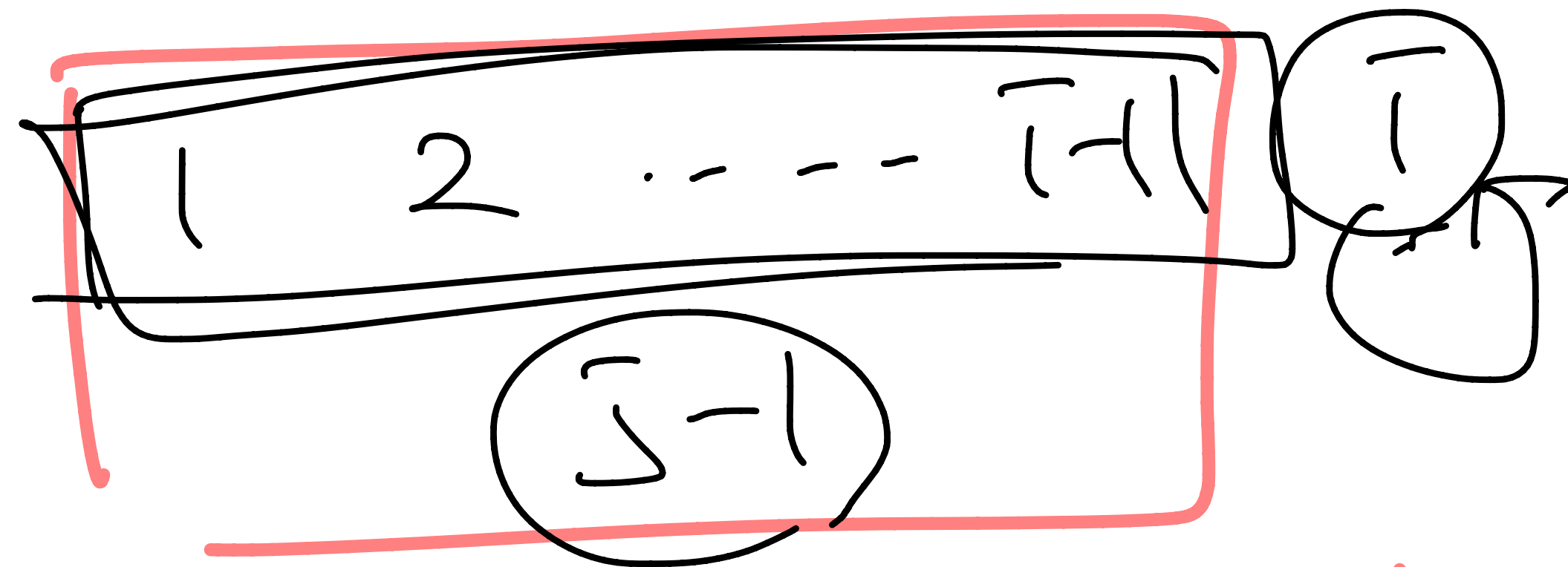
# 박스 안의 열쇠

<https://www.acmicpc.net/problem/1413>

- 사이클의 개수가 M개를 넘지 않는 길이가 N인 순열의 개수를 구하는 문제
- 이 수는 제 1종 스털링 수 (Stirling numbers of the first kind) 와 같다
- i번째 수가 혼자 사이클을 이루는 경우
- $D[i-1][j-1]$

$$D[i][j] = \text{길이가 } i \text{인 순열에서 } j \text{개의 사이클이 있는 경우}$$

사이클의 수  $j$

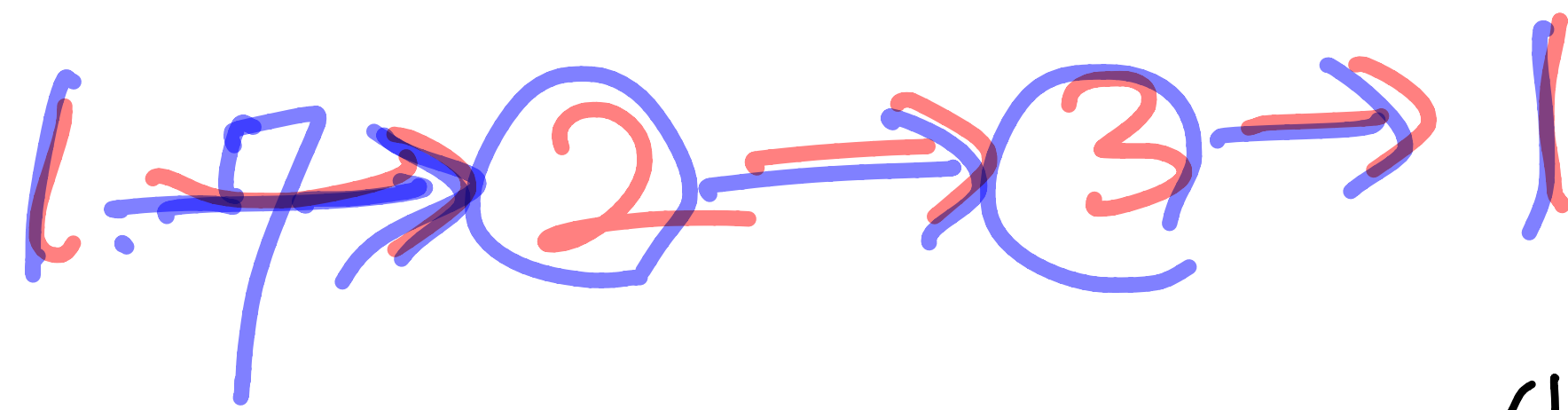


$$D[i][j] = D[i-1][j-1] + (i-1) \cdot D[i-1][j]$$

# 박스 안의 열쇠

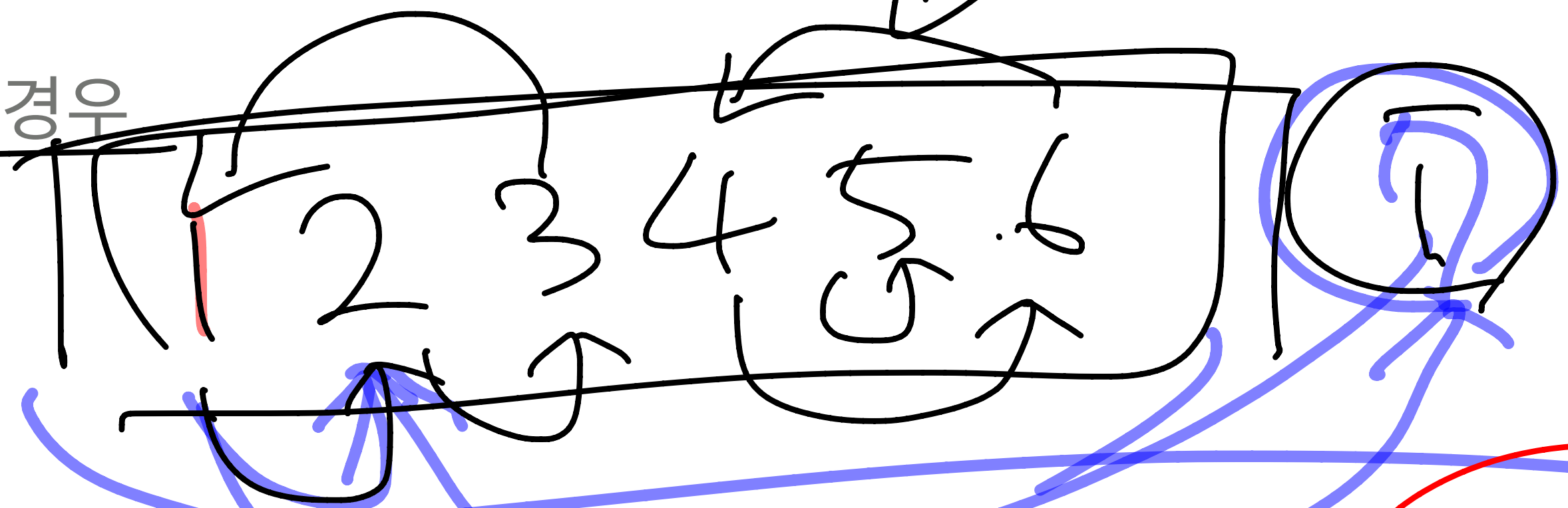
<https://www.acmicpc.net/problem/1413>

77

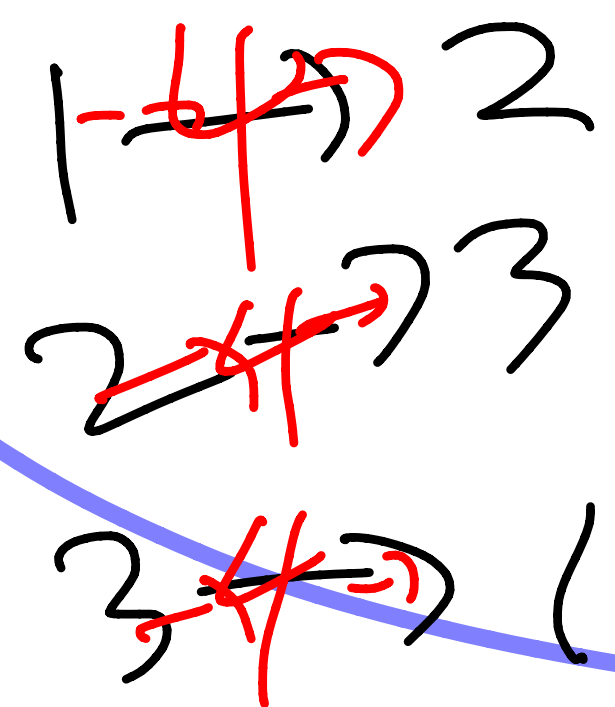


1 2 3 5

- 사이클의 개수가 M개를 넘지 않는 길이가 N인 순열의 개수를 구하는 문제
- 이 수는 제 1종 스털링 수 (Stirling numbers of the first kind) 와 같다
- i번째 수가 기존에 있던 사이클에 들어가는 경우
- $D[i-1][j] * (i-1)$



$D[i-1][j] * (i-1)$



4

# 박스 안의 열쇠

78

<https://www.acmicpc.net/problem/1413>

- 소스: <http://codeplus.codes/5af1df198ae648528edbe79dd9ec7272>

# 팔굽혀펴기

<https://www.acmicpc.net/problem/10564>

- 어떤 팀이 득점에 성공할 때마다, 그 팀이 점수만큼 팔굽혀펴기를 해야한다.
- 예를 들어, 어떤 팀이 7, 3, 2점의 순서로 득점을 했으면, 팔굽혀펴기를 한 횟수는 7, 7+3, 7+3+2번으로 총 29번이다.
- 팔굽혀펴기를 한 횟수와 어떤 경기에서 가능한 득점의 종류가 주어졌을 때, 팀의 점수의 최댓값을 구하는 문제
- 팔굽혀펴기를 한 횟수  $\leq 5000$

$$7 + 10 + 12 = 29$$

2, 3

1 - 4

# 팔굽혀펴기

<https://www.acmicpc.net/problem/10564>

80

•  $D[i][j]$  = 총 팔굽혀펴기를  $i$  번 했고, 이 때, 점수  $j$  가 가능한가?

• 새로운 점수를  $k$  라고 했을 때

•  $D[i][j] \rightarrow$

$[1 + j + k] [j + k]$

이런 경계이식  
가능성은 없는 것 같음



# 팔굽혀펴기

<https://www.acmicpc.net/problem/10564>

- $D[i][j]$  = 총 팔굽혀펴기를  $i$ 번했고, 이 때, 점수  $j$ 가 가능한가?
- 새로운 점수를  $k$ 라고 했을 때
- $D[i][j] \rightarrow D[i+j+k][j+k]$

Handwritten diagram illustrating the recurrence relation  $D[i][j] = 1$ . The expression is enclosed in an oval. Arrows point from the indices  $i$  and  $j$  to the corresponding positions in the array notation, and an arrow points from the result  $1$  to the right side of the equation.

# 팔굽혀펴기

82

<https://www.acmicpc.net/problem/10564>

- 소스: <http://codeplus.codes/f25d445f10e64083abf7bce002ebd250>

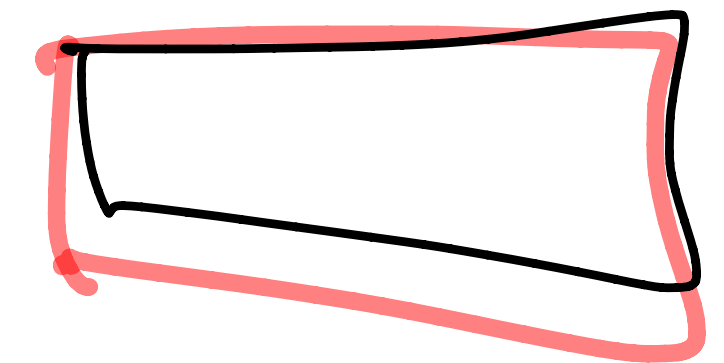
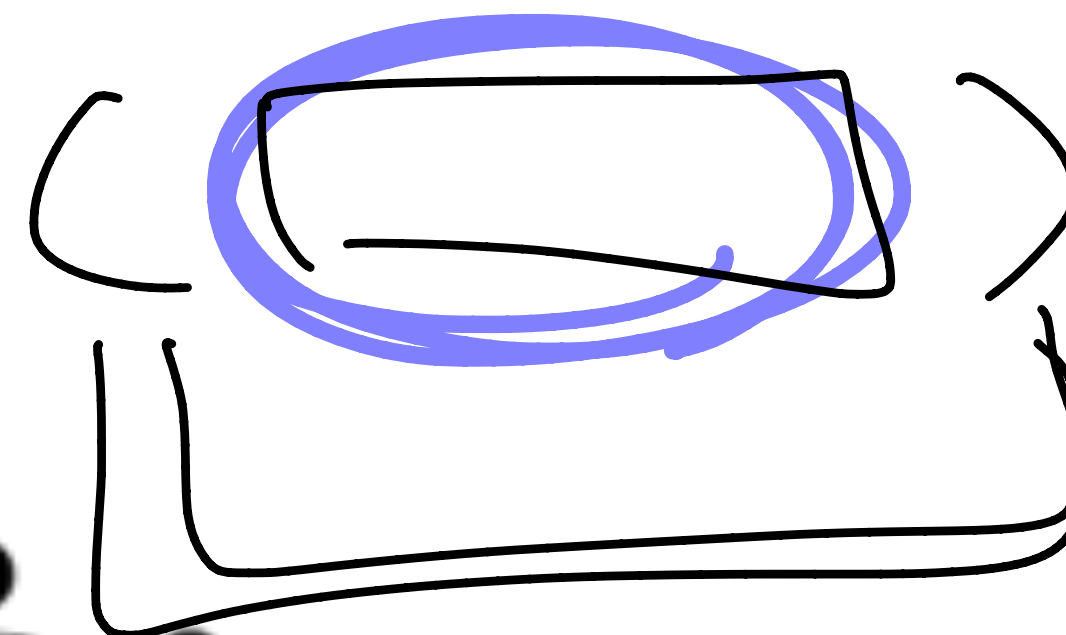
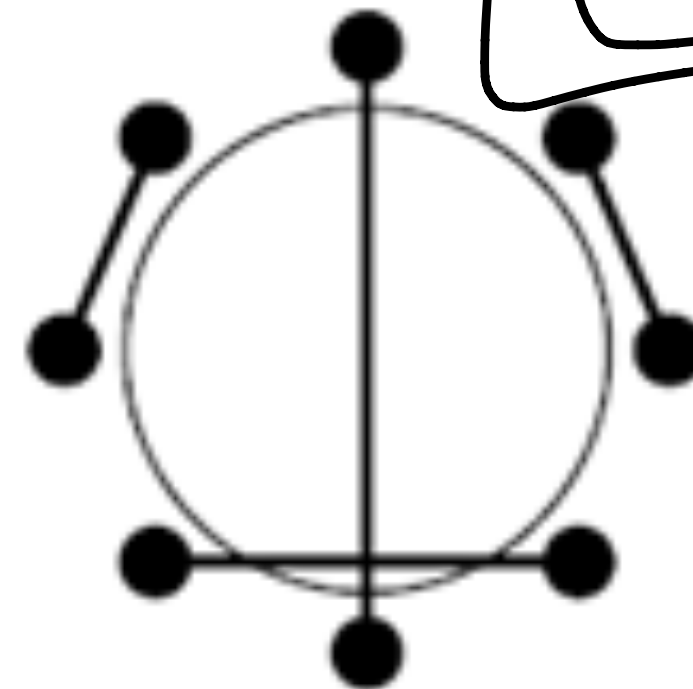
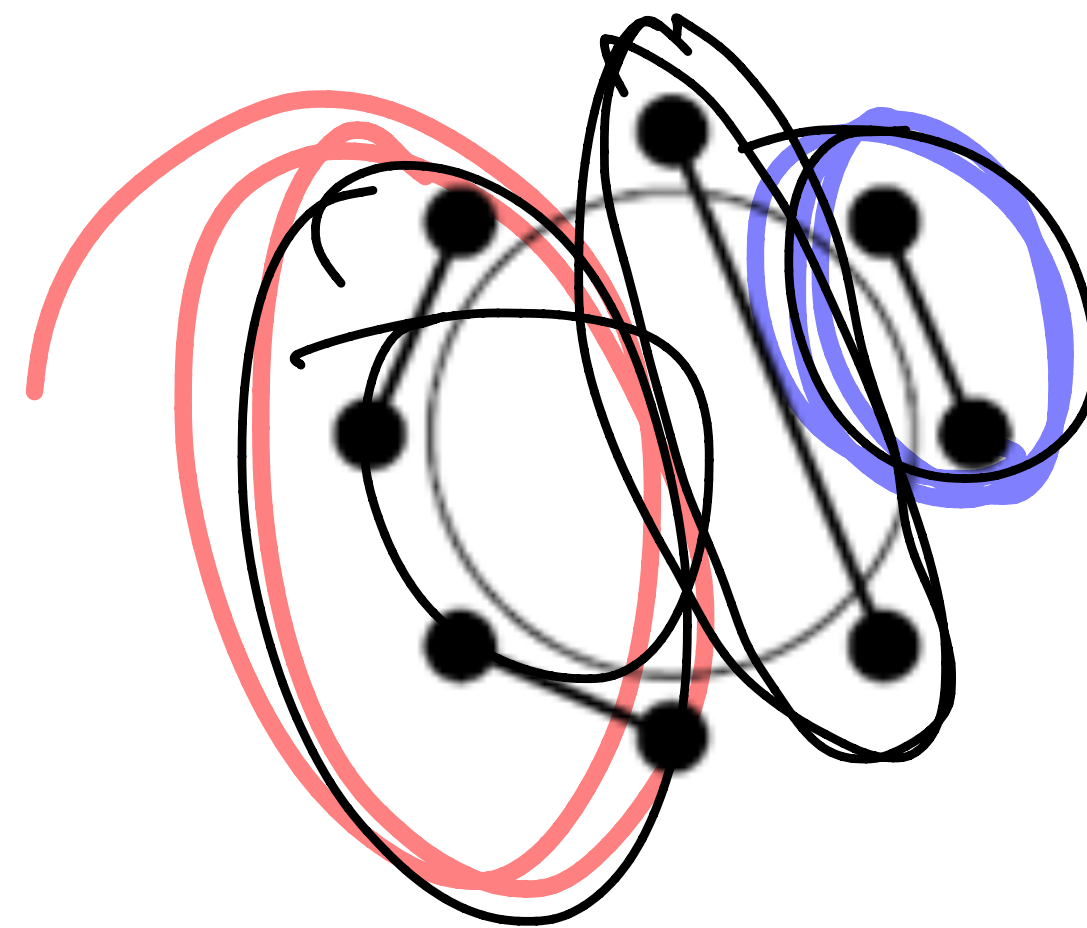
# 건배

N

<https://www.acmicpc.net/problem/1970>

- 겹치지않게 최대 몇 쌍이 건배를 할 수 있는가?
- 콜라 브랜드가 같아야 건배를 할 수 있다

같은 콜라병이 개수  
올바른 건배 개수 ?

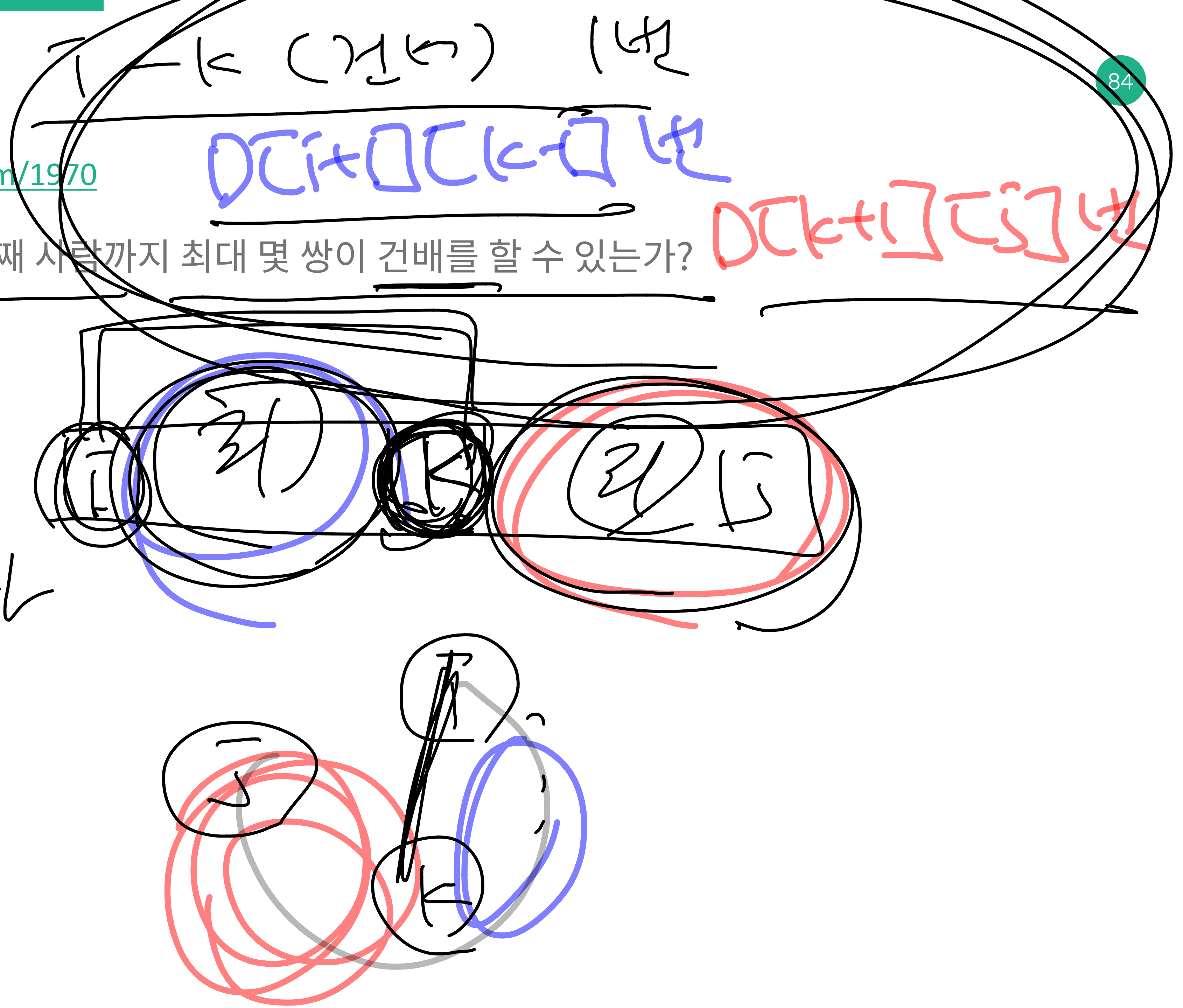


# 건배

<https://www.acmicpc.net/problem/1970>

- $D[i][j]$  =  $i$ 번째 사람부터  $j$ 번째 사람까지 최대 몇 쌍이 건배를 할 수 있는가?

$T < K \leq J$   
이 조건만  
= 이 조건만  
만족



# 건배

<https://www.acmicpc.net/problem/1970>

- $i$ 번째 사람과 건배를 할 사람  $k$ 번째를 찾아야 한다.

# 건배

<https://www.acmicpc.net/problem/1970>

- $A[i] == A[k]$ 인 경우 (건배를 할 수 있는 경우)
- $A[i] != A[k]$ 인 경우 (건배를 할 수 없는 경우)

# 건배

<https://www.acmicpc.net/problem/1970>

- $A[i] == A[k]$ 인 경우 (건배를 할 수 있는 경우)
- $D[i+1][k-1] + D[k+1][j] + 1$
- $A[i] != A[k]$ 인 경우 (건배를 할 수 없는 경우)
- 없음

# 건배

<https://www.acmicpc.net/problem/1970>

- i번째 사람과 건배를 할 사람 k번째가 없는 경우
- $D[i+1][j]$



# 건배

<https://www.acmicpc.net/problem/1970>

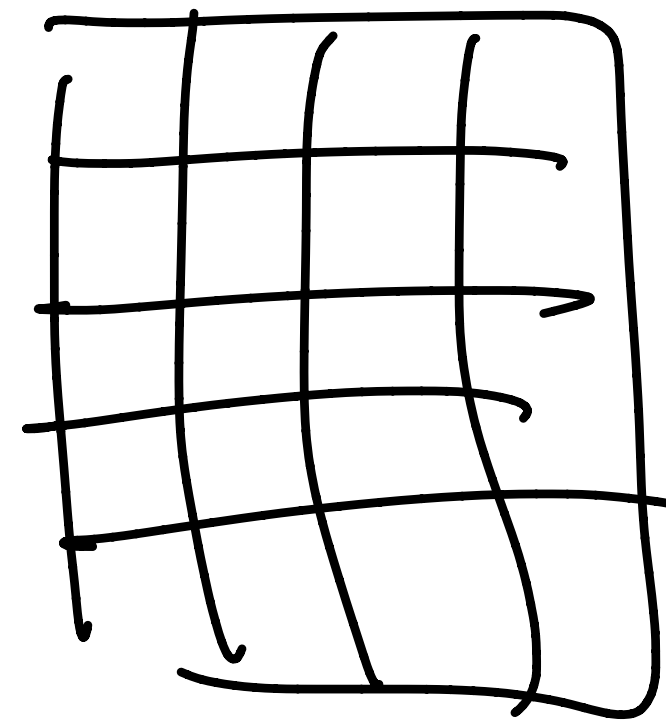
- 소스: <http://codeplus.codes/78dddba7dafb4bcaaa3b93e6ed5463c4>

# 초콜릿 자르기

90

<https://www.acmicpc.net/problem/2163>

- 정화는  $N \times M$  크기의 초콜릿을 하나 가지고 있다
- 초콜릿은 금이 가 있는 모양을 하고 있으며, 그 금에 의해  $N \times M$ 개의 조각으로 나뉘질 수 있다
- 초콜릿의 크기가 주어졌을 때, 이를  $1 \times 1$  크기의 초콜릿으로 쪼개기 위한 최소 쪼개기 회수를 구하는 문제



# 초콜릿 자르기

<https://www.acmicpc.net/problem/2163>

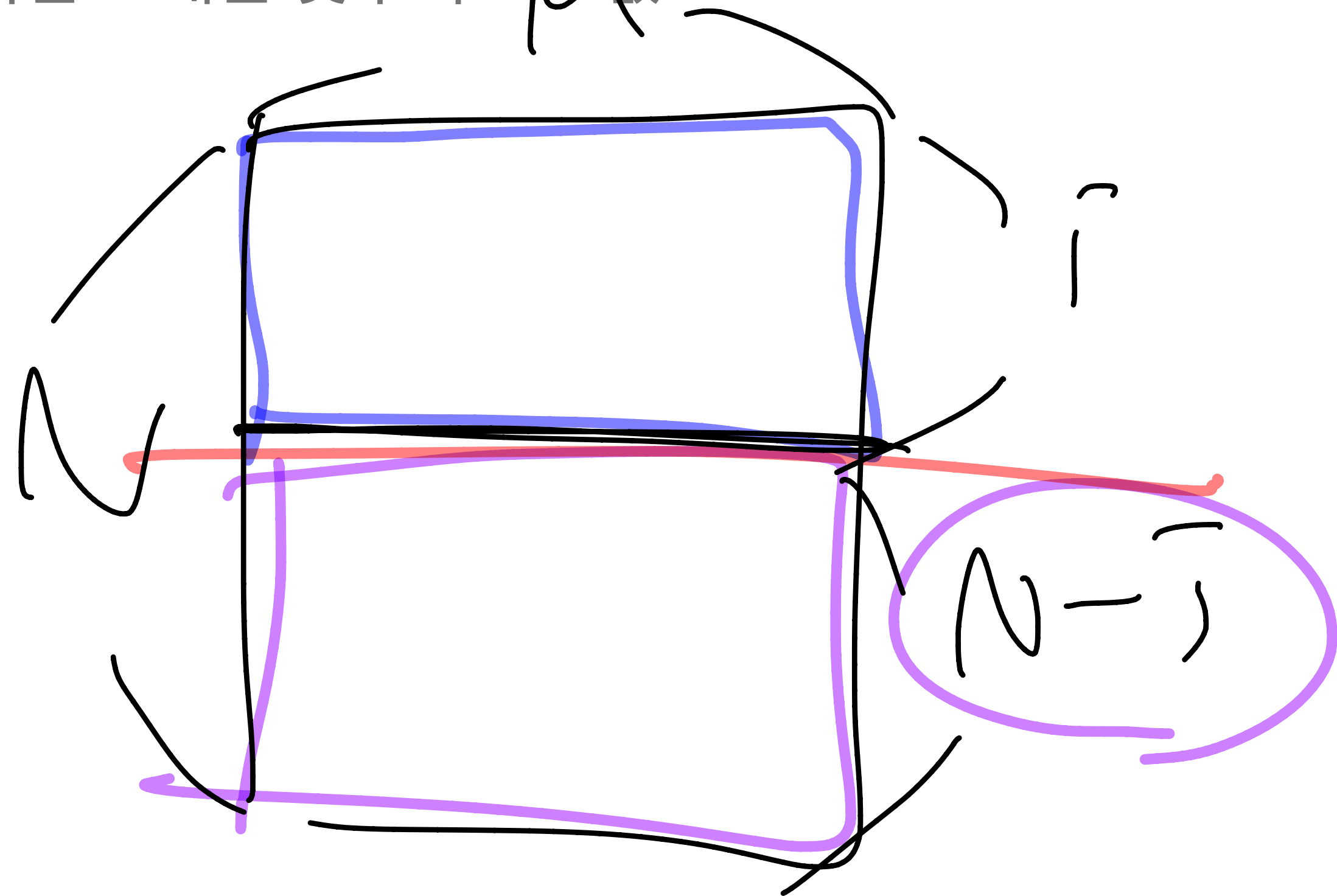
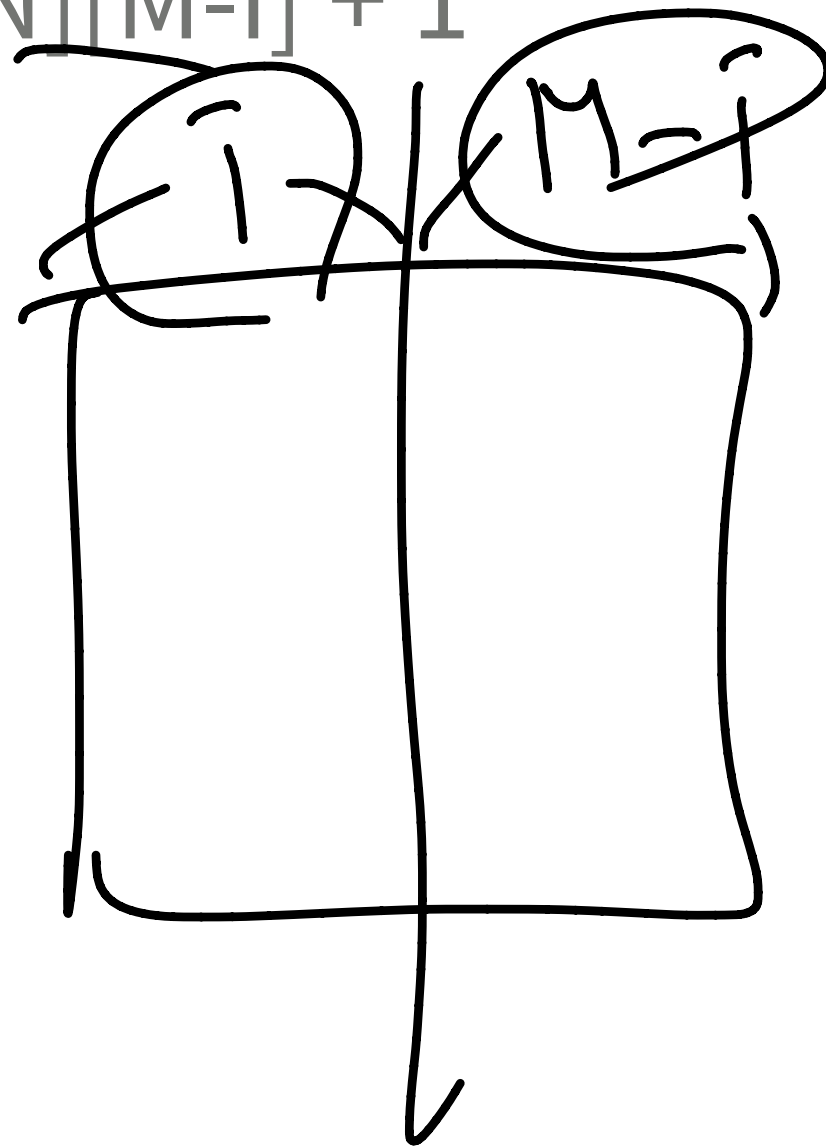
- $D[N][M] = N \times M$  초콜릿을  $1 \times 1$ 로 만들기 위한 쪼개는 횟수의 최소값

- 가로로 자르는 경우

- $D[i][M] + D[N-i][M] + 1$

- 세로로 자르는 경우

- $D[N][i] + D[N][M-i] + 1$



# 초콜릿 자르기

<https://www.acmicpc.net/problem/2163>

- 소스: <http://codeplus.codes/c03316520f1e4ebfbf0b27eec6d46ce0>

