

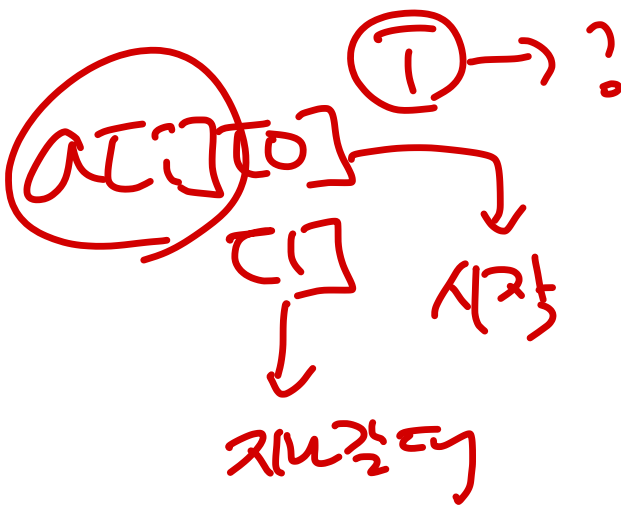
CS

최백준 choi@startlink.io



Java

```
1 import java.util.*;
2
3 public class Main {
4     static final int n = 33;
5     static int[] score = {
6         0,2,4,6,8,
7         10,13,16,19,25,
8         12,14,16,18,20,
9         22,24,22,24,26,
10        28,26,27,28,30,
11        32,34,36,38,30,
12        35,40,0
13    };
14    static final int t = 10;
15    static int[] dice = new int[t];
16    static int[][] a = new int[n][];
17    static int get_next(int start, int k) {
18        int now = start;
19        for (int i=0; i<k; i++) {
20            if (i == 0) {
21                now = a[now][0];
22            } else {
23                now = a[now][1];
24            }
25        }
26        return now;
27    }
28    static int go(int index, int[] horse, int sum) {
29        if (index == t) return sum;
30        int ans = 0;
31        for (int i=0; i<4; i++) {
32            int nxt = get_next(horse[i], dice[index]);
33            boolean ok = true;
34            if (nxt != n-1) {
35                for (int j=0; j<4; j++) {
36                    if (i == j) continue;
37                    if (nxt == horse[j]) ok = false;
38                }
39            }
40            if (ok) {
41                int[] nhorse = horse.clone();
42                nhorse[i] = nxt;
43                int temp = go(index+1, nhorse, sum+score[nxt]);
44                if (ans < temp) ans = temp;
45            }
46        }
47        return ans;
48    }
49    public static void main(String[] args) {
50        Scanner sc = new Scanner(System.in);
51        a[0] = new int[]{1,1};
52        a[1] = new int[]{2,2};
53        a[2] = new int[]{3,3};
54        a[3] = new int[]{4,4};
55        a[4] = new int[]{5,5};
56        a[5] = new int[]{6,10};
57        a[6] = new int[]{7,7};
58        a[7] = new int[]{8,8};
59        a[8] = new int[]{9,9};
60        a[9] = new int[]{29,29};
61        a[10] = new int[]{11,11};
62        a[11] = new int[]{12,12};
63        a[12] = new int[]{13,13};
64        a[13] = new int[]{14,14};
65        a[14] = new int[]{15,17};
66        a[15] = new int[]{16,16};
67        a[16] = new int[]{9,9};
68        a[17] = new int[]{18,18};
69        a[18] = new int[]{19,19};
70        a[19] = new int[]{20,20};
71        a[20] = new int[]{24,24};
72        a[21] = new int[]{9,9};
73        a[22] = new int[]{21,21};
74        a[23] = new int[]{22,22};
75        a[24] = new int[]{23,25};
76        a[25] = new int[]{26,26};
77        a[26] = new int[]{27,27};
78        a[27] = new int[]{28,28};
79        a[28] = new int[]{31,31};
80        a[29] = new int[]{30,30};
81        a[30] = new int[]{31,31};
82        a[31] = new int[]{32,32};
83        a[32] = new int[]{32,32};
84        for (int i=0; i<t; i++) {
85            dice[i] = sc.nextInt();
86        }
87        System.out.println(go(0, new int[]{0, 0, 0, 0}, 0));
88    }
89 }
90
```



시작칸

> 함수

> 재귀

반말이

반말이
의미?

now → nxt

0(4¹⁰)

BFS

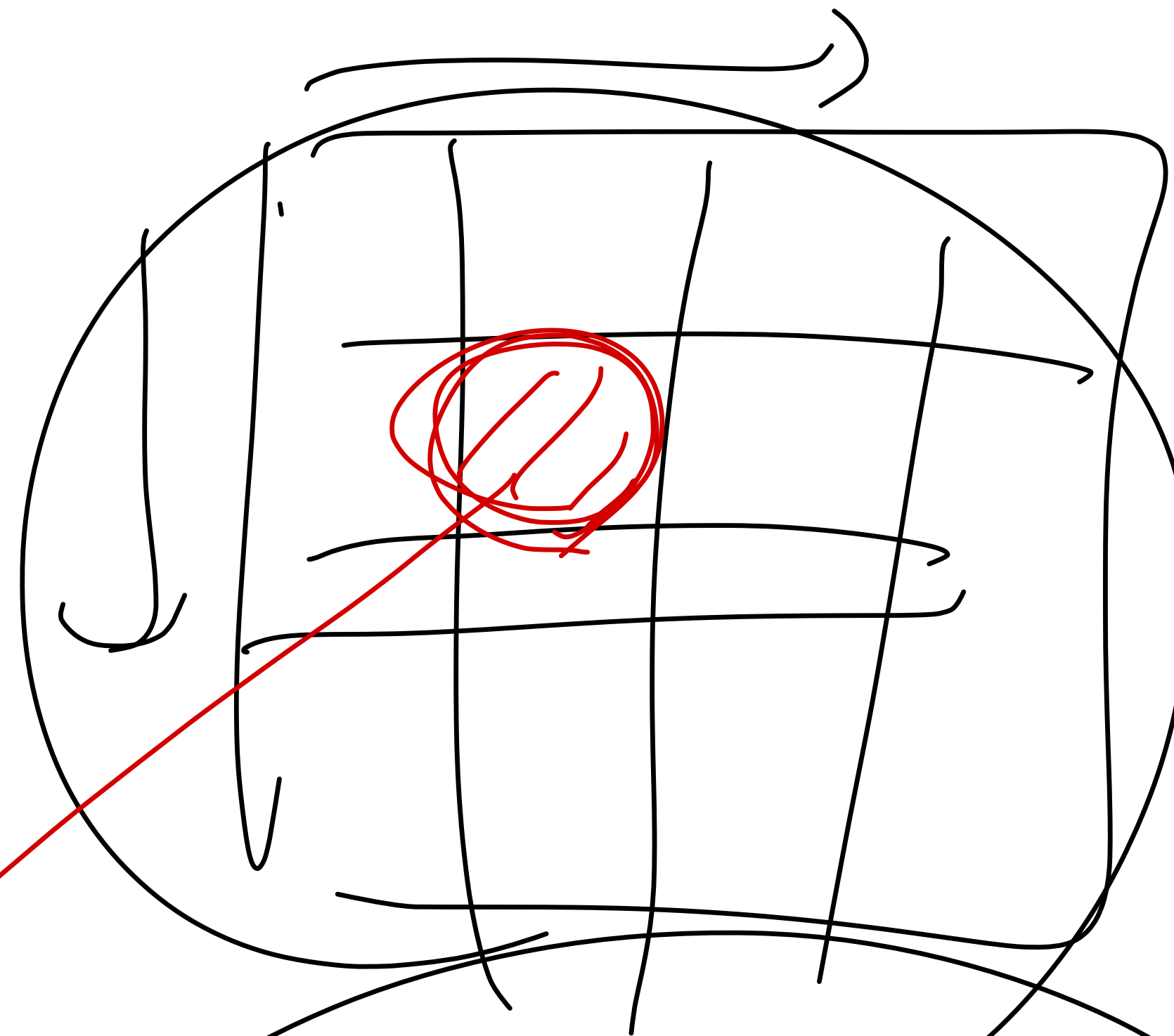
2822

row
col



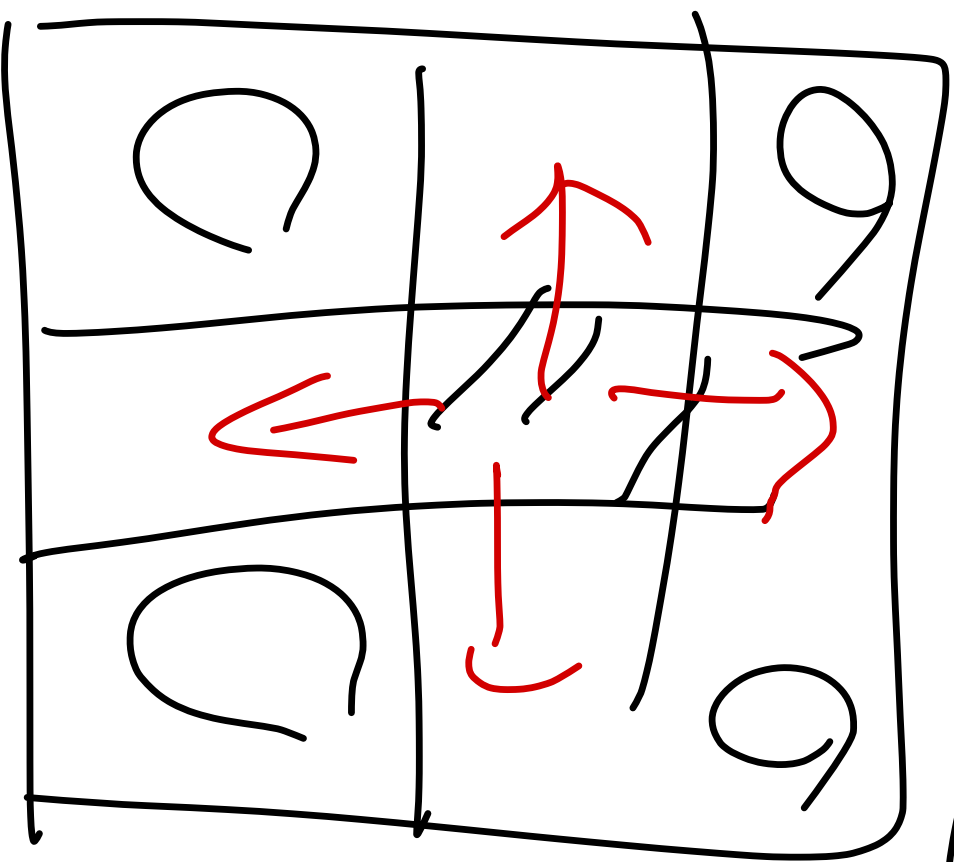
(3, 2)

2822
row col



unvisited, unvisited

$\frac{dx}{dy}, \frac{dy}{dx}$



$$(x, y) \rightarrow (nx, ny)$$

$$\frac{dx}{dy} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$(x-1, y)$$

$$(x, y+1)$$

$$(x, y-1)$$

$$(x, y)$$

$$(x+1, y)$$

$$nx = x + dx \cdot k$$

$$ny = y + dy \cdot k$$

$M_{3/2}^2$

$$d_X = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}$$

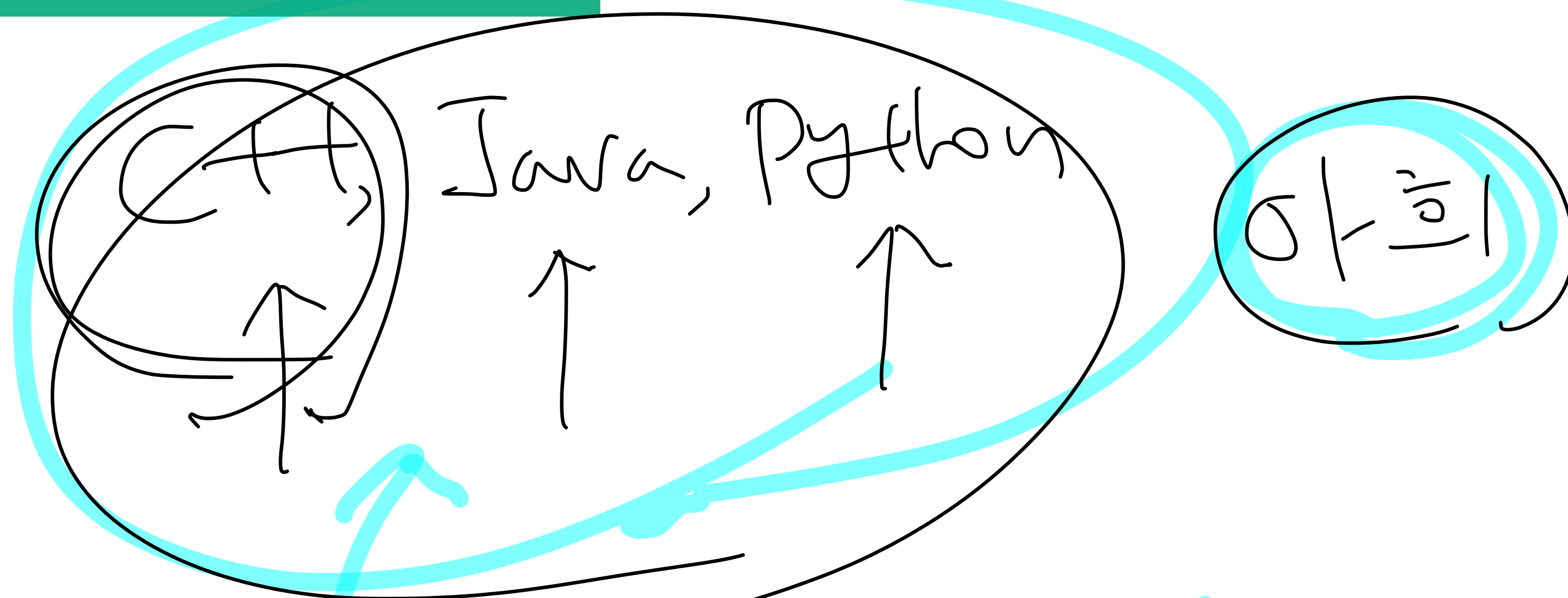
$$d_g = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

정렬된 수열 u, v

	8		1	
7				2
		⊕		
6				3
	5		4	

$$dx = [-2, -1, 1, 2, 2, 1, -1, -2]$$

$$dy = [1, 2, 2, 1, -1, -2, -2, -1]$$



교집합

차집

C++

~
o

Cin / cout

Java ;

Scanner

System.out

Python

2/3/2

If x In @

O(N)

Python

$a = [1, 2, 3]$

$a[1:]$

$[2, 3]$

BFS

$a[0:1]$

Queue

$q = []$

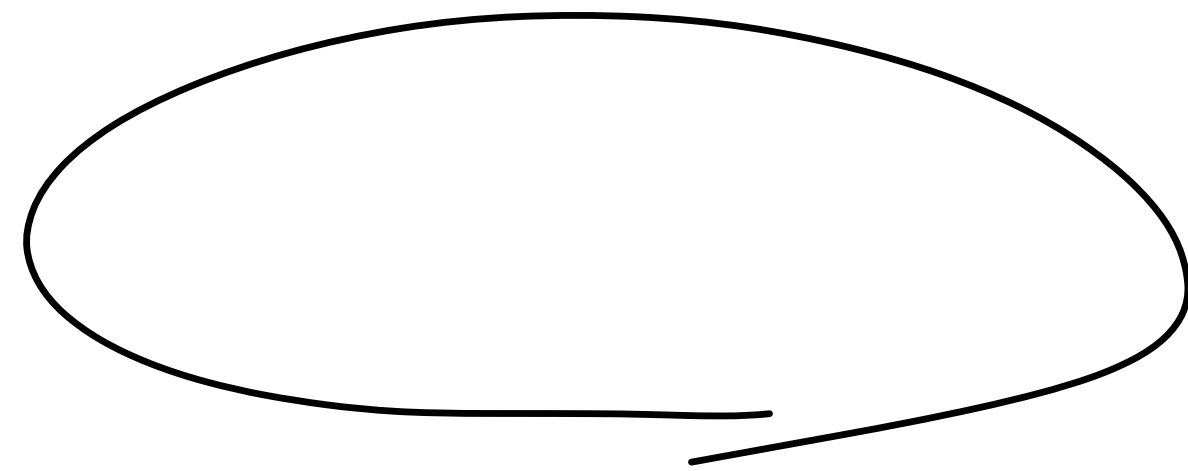
$q.append(start)$

while q :

$now = q[0]$

$q = q[1:]$

for y in $q[now]$:



$O(N)$
 \nearrow
 $q.pop()$

from collections import deque

Java

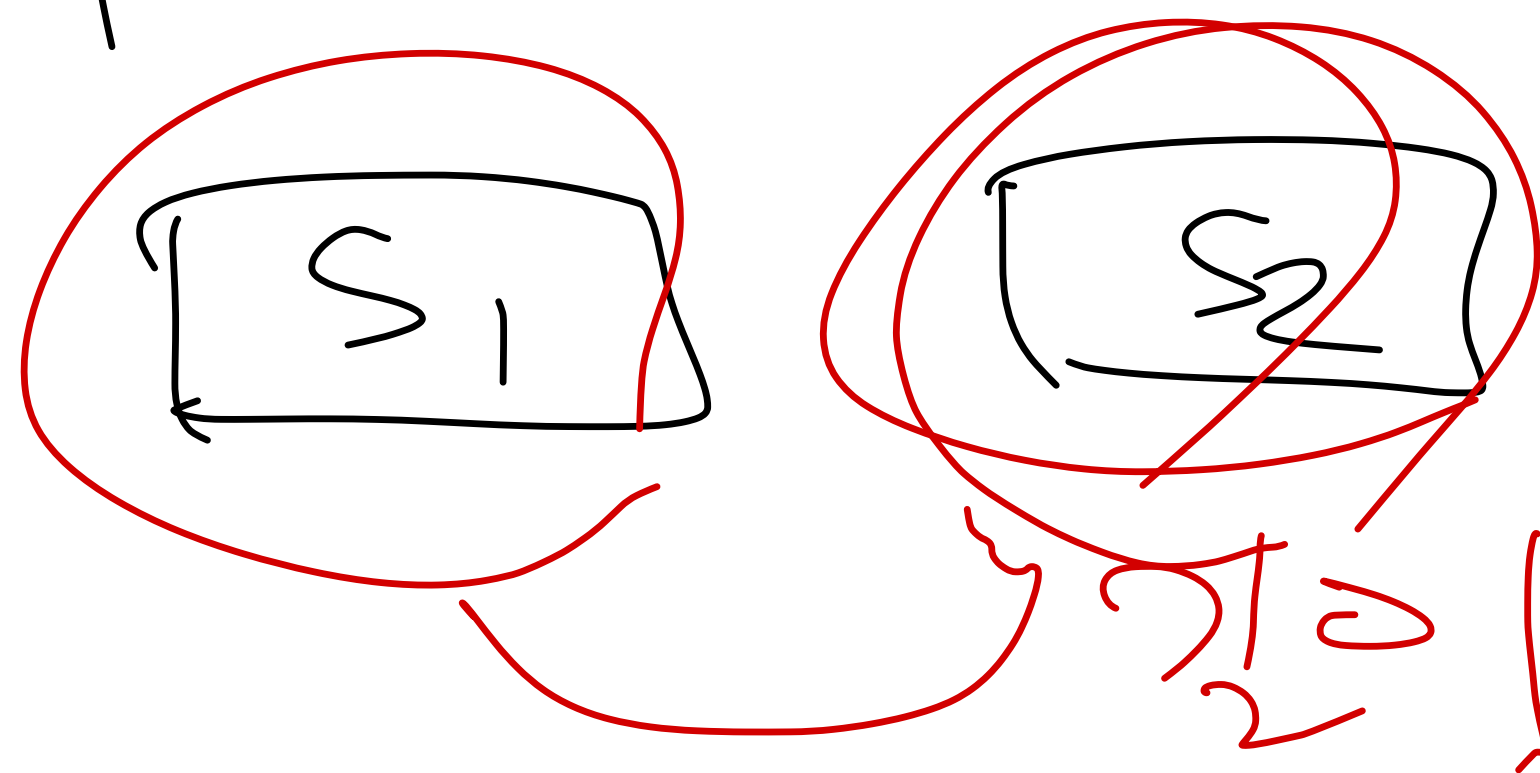
String S = "";

for (i=0; i<100000; i++) {

S += "abc";

}

C++



Jan

Java

```
1 import java.util.*;
2
3 class Pair {
4     int first, second;
5     Pair(int first, int second) {
6         this.first = first;
7         this.second = second;
8     }
9 }
10 public class Main {
11     static final int[] dx = {0,0,1,-1};
12     static final int[] dy = {1,-1,0,0};
13     static void go(Integer[][] a, int x, int d, int k) {
14         if (d == 0) {
15             Collections.rotate(Arrays.asList(a[x]), k);
16         } else {
17             Collections.rotate(Arrays.asList(a[x]), (a[x].length-k));
18         }
19     }
20     static boolean bfs(Integer[][] a) {
21         int n = a.length-1;
22         int m = a[1].length;
23         boolean[][] d = new boolean[n+1][m];
24         boolean ok = false;
25         for (int i=1; i<=n; i++) {
26             for (int j=0; j<m; j++) {
27                 if (d[i][j] == true) continue;
28                 if (a[i][j].intValue() == 0) continue;
29                 ArrayList<Pair> group = new ArrayList<>();
30                 int cnt = 0;
31                 Queue<Pair> q = new LinkedList<>();
32                 q.add(new Pair(i,j));
33                 group.add(new Pair(i,j));
34                 d[i][j] = true;
35                 while (!q.isEmpty()) {
36                     Pair p = q.remove();
37                     int x = p.first;
38                     int y = p.second;
39                     cnt += 1;
40                     for (int k=0; k<4; k++) {
41                         int nx = x+dx[k];
42                         int ny = y+dy[k];
43                         ny = (ny + m) % m;
44                         if (1 <= nx && nx <= n && 0 <= ny && ny < m) {
45                             if (d[nx][ny] == false && a[x][y].intValue() == a[nx][ny].intValue()) {
46                                 q.add(new Pair(nx,ny));
47                                 group.add(new Pair(nx,ny));
48                                 d[nx][ny] = true;
49                             }
50                         }
51                     }
52                     if (group.size() == 1) continue;
53                     ok = true;
54                     for (Pair p : group) {
55                         int x = p.first;
56                         int y = p.second;
57                         a[x][y] = 0;
58                     }
59                 }
60             }
61         }
62         return ok;
63     }
64     static void adjust(Integer[][] a) {
65         int n = a.length-1;
66         int m = a[1].length;
67         long sum = 0;
68         long cnt = 0;
69         for (int i=1; i<=n; i++) {
70             for (int j=0; j<m; j++) {
71                 if (a[i][j].intValue() == 0) continue;
72                 sum += a[i][j];
73                 cnt += 1;
74             }
75         }
76         if (cnt == 0) return;
77         for (int i=1; i<=n; i++) {
78             for (int j=0; j<m; j++) {
79                 if (a[i][j].intValue() == 0) continue;
80                 if (sum < (long)a[i][j].intValue()*cnt) {
81                     // sum/cnt < a[i][j] (-1)
82                     a[i][j] -= 1;
83                 } else if (sum > (long)a[i][j].intValue()*cnt) {
84                     // sum/cnt > a[i][j] (+1)
85                     a[i][j] += 1;
86                 }
87             }
88         }
89     }
90     public static void main(String[] args) {
91         Scanner sc = new Scanner(System.in);
92         int n = sc.nextInt();
93         int m = sc.nextInt();
94         int t = sc.nextInt();
95         Integer[][] a = new Integer[n+1][m];
96         for (int i=1; i<=n; i++) {
97             for (int j=0; j<m; j++) {
98                 a[i][j] = sc.nextInt();
99             }
100         }
101         while (t-- > 0) {
102             int x = sc.nextInt();
103             int d = sc.nextInt();
104             int k = sc.nextInt();
105             for (int y=x; y<=n; y+=x) {
106                 go(a, y, d, k);
107             }
108             boolean ok = bfs(a);
109             if (ok == false) {
110                 adjust(a);
111             }
112         }
113         int ans = 0;
114         for (int i=1; i<=n; i++) {
115             for (int j=0; j<m; j++) {
116                 ans += a[i][j].intValue();
117             }
118         }
119         System.out.println(ans);
120     }
121 }
122 }
```

결과	메모리	시간	코드 길이
맞았습니다!	46196 KB	404 ms	3907 B

C++14

```
1 #include <iostream>
2 #include <map>
3 #include <queue>
4 #include <vector>
5 #include <algorithm>
6 using namespace std;
7 int dx[] = {0,0,1,-1};
8 int dy[] = {1,-1,0,0};
9 void go(vector<vector<int>> &a, int x, int d, int k) {
10     if (d == 0) {
11         rotate(a[x].rbegin(), a[x].rbegin()+k, a[x].rend());
12     } else { // d == 1
13         rotate(a[x].begin(), a[x].begin()+k, a[x].end());
14     }
15 }
16 bool bfs(vector<vector<int>> &a) {
17     int n = (int)a.size() - 1;
18     int m = a[1].size();
19     vector<vector<bool>> d(n+1, vector<bool>(m, 0));
20     bool ok = false;
21     for (int i=1; i<=n; i++) {
22         for (int j=0; j<m; j++) {
23             if (d[i][j] == true) continue;
24             if (a[i][j] == 0) continue;
25             vector<pair<int,int>> group;
26             int cnt = 0;
27             queue<pair<int,int>> q;
28             q.push(make_pair(i,j));
29             group.push_back(make_pair(i,j));
30             d[i][j] = true;
31             while (!q.empty()) {
32                 int x, y;
33                 tie(x, y) = q.front(); q.pop();
34                 cnt += 1;
35                 for (int k=0; k<4; k++) {
36                     int nx = x+dx[k];
37                     int ny = y+dy[k];
38                     ny = (ny + m) % m;
39                     if (1 <= nx && nx <= n && 0 <= ny && ny < m) {
40                         if (d[nx][ny] == false && a[x][y] == a[nx][ny]) {
41                             q.push(make_pair(nx,ny));
42                             group.push_back(make_pair(nx,ny));
43                             d[nx][ny] = true;
44                         }
45                     }
46                 }
47             }
48             if (group.size() == 1) continue;
49             ok = true;
50             for (auto &p : group) {
51                 int x, y;
52                 tie(x,y) = p;
53                 a[x][y] = 0;
54             }
55         }
56     }
57     return ok;
58 }
59 void adjust(vector<vector<int>> &a) {
60     int n = (int)a.size() - 1;
61     int m = a[1].size();
62     long long sum = 0;
63     long long cnt = 0;
64     for (int i=1; i<=n; i++) {
65         for (int j=0; j<m; j++) {
66             if (a[i][j] == 0) continue;
67             sum += a[i][j];
68             cnt += 1;
69         }
70     }
71     if (cnt == 0) return;
72     for (int i=1; i<=n; i++) {
73         for (int j=0; j<m; j++) {
74             if (a[i][j] == 0) continue;
75             if (sum < (long long)a[i][j]*cnt) {
76                 // sum/cnt < a[i][j] (-1)
77                 a[i][j] -= 1;
78             } else if (sum > (long long)a[i][j]*cnt) {
79                 // sum/cnt > a[i][j] (+1)
80                 a[i][j] += 1;
81             }
82         }
83     }
84 }
85 int main() {
86     int n, m, t;
87     cin >> n >> m >> t;
88     vector<vector<int>> a(n+1, vector<int>(m));
89     for (int i=1; i<=n; i++) {
90         for (int j=0; j<m; j++) {
91             cin >> a[i][j];
92         }
93     }
94     while (t--) {
95         int x, d, k;
96         cin >> x >> d >> k;
97         for (int y=x; y<=n; y+=x) {
98             go(a, y, d, k);
99         }
100         bool ok = bfs(a);
101         if (ok == false) {
102             adjust(a);
103         }
104     }
105     int ans = 0;
106     for (int i=1; i<=n; i++) {
107         for (int j=0; j<m; j++) {
108             ans += a[i][j];
109         }
110     }
111     cout << ans << '\n';
112     return 0;
113 }
```

결과	메모리	시간	코드 길이
맞았습니다!	1988 KB	28 ms	3149 B

17425 $O(\frac{N}{2})$ δ_2 $N \leq 1012$

$f(A) = A$ $B \subseteq$ $O(\frac{N}{2})$ δ_6

$\underbrace{f(24)}_{O(N)} = 1+2+3+4+6+8+12+24$

$$g(N) = f(1) + \dots + f(N)$$

\textcircled{N} $\textcircled{g(N)} \rightarrow O(N^2)$

$$f(1) + f(2) + \dots + f(24)$$

int

f(int N) $O(N)$

int ans = 0;

for(i = 1; i <= N; i++)

if ($N \% i == 0$)

ans += i;

return ans;

}

$$\textcircled{11} = 1, 7, 11, 17$$

$\textcircled{2}$

2 4 6 8 ...

$$f[2]_t = 2$$

$$f(4)_t = ?$$

$\textcircled{3}$

$$f(8)_t = 2$$

3 6 9 12 ...

C++14

```
1 #include <iostream>
2 #include <tuple>
3 #include <algorithm>
4 #include <vector>
5 using namespace std;
6 int dx[] = {0,0,-1,1};
7 int dy[] = {1,-1,0,0};
8 int opposite(int dir) {
9     if (dir == 0) return 1;
10    if (dir == 1) return 0;
11    if (dir == 2) return 3;
12    return 2;
13 }
14 void go(vector<vector<vector<pair<int,int>>>> &a, vector<pair<int,int>> &where, int x, int y, int nx, int ny) {
15     for (auto &p : a[x][y]) {
16         a[nx][ny].push_back(p);
17         where[p.first] = make_pair(nx, ny);
18     }
19     a[x][y].clear();
20 }
21 void print(vector<vector<vector<pair<int,int>>>> &a) {
22     int n = a.size();
23     for (int i=0; i<n; i++) {
24         for (int j=0; j<n; j++) {
25             if (a[i][j].size() == 0) {
26                 cout << "- ";
27             } else {
28                 for (auto &p : a[i][j]) {
29                     cout << p.first << ',' << p.second << ' ';
30                 }
31             }
32             cout << '\n';
33         }
34     }
35 }
36 int main() {
37     int n, m;
38     cin >> n >> m;
39     vector<vector<int>> board(n, vector<int>(n));
40     for (int i=0; i<n; i++) {
41         for (int j=0; j<n; j++) {
42             cin >> board[i][j];
43         }
44     }
45     vector<vector<vector<pair<int,int>>>> a(n, vector<vector<pair<int,int>>>(n));
46     vector<pair<int,int>> where(m);
47     for (int i=0; i<m; i++) {
48         int x, y, dir;
49         cin >> x >> y >> dir;
50         a[x-1][y-1].push_back(make_pair(i, dir-1));
51         where[i] = make_pair(x-1, y-1);
52     }
53     for (int turn=1; turn<=1000; turn++) {
54         for (int k=0; k<m; k++) {
55             int x, y;
56             tie(x,y) = where[k];
57             if (a[x][y][0].first == k) { // bottom
58                 int dir = a[x][y][0].second;
59                 int nx = x+dx[dir];
60                 int ny = y+dy[dir];
61                 if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
62                     if (board[nx][ny] == 2) {
63                         a[x][y][0].second = opposite(dir);
64                     }
65                 } else { // out
66                     a[x][y][0].second = opposite(dir);
67                 }
68                 dir = a[x][y][0].second;
69                 nx = x+dx[dir];
70                 ny = y+dy[dir];
71                 if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
72                     if (board[nx][ny] == 0) {
73                         go(a, where, x, y, nx, ny);
74                     } else if (board[nx][ny] == 1) {
75                         reverse(a[x][y].begin(), a[x][y].end());
76                         go(a, where, x, y, nx, ny);
77                     }
78                     if (a[nx][ny].size() >= 4) {
79                         cout << turn << '\n';
80                         return 0;
81                     }
82                 } else { // out
83                     }
84             }
85             //cout << "#" << turn << ' ' << k << '\n';
86             //print(a);
87         }
88     }
89     cout << -1 << '\n';
90     return 0;
91 }
92 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

1988 KB

0 ms

2835 B

C++14

```
1 #include <iostream>
2 #include <cassert>
3 #include <tuple>
4 #include <algorithm>
5 #include <vector>
6 using namespace std;
7 int dx[] = {0,0,-1,1};
8 int dy[] = {1,-1,0,0};
9 int opposite(int dir) {
10     if (dir == 0) return 1;
11     if (dir == 1) return 0;
12     if (dir == 2) return 3;
13     return 2;
14 }
15 void go(vector<vector<vector<pair<int,int>>>> &a, vector<tuple<int,int,int>> &where, int x, int y, int nx, int
ny, int start) {
16     for (int i=start; i<a[x][y].size(); i++) {
17         auto &p = a[x][y][i];
18         a[nx][ny].push_back(p);
19         where[p.first] = make_tuple(nx, ny, (int)a[nx][ny].size()-1);
20     }
21     a[x][y].resize(start);
22 }
23 void print(vector<vector<vector<pair<int,int>>>> &a) {
24     int n = a.size();
25     for (int i=0; i<n; i++) {
26         for (int j=0; j<n; j++) {
27             if (a[i][j].size() == 0) {
28                 cout << "- ";
29             } else {
30                 for (auto &p : a[i][j]) {
31                     cout << p.first << ',' << p.second << ' ';
32                 }
33             }
34         }
35         cout << '\n';
36     }
37 }
38 int main() {
39     int n, m;
40     cin >> n >> m;
41     vector<vector<int>> board(n, vector<int>(n));
42     for (int i=0; i<n; i++) {
43         for (int j=0; j<n; j++) {
44             cin >> board[i][j];
45         }
46     }
47     vector<vector<vector<pair<int,int>>>> a(n, vector<vector<pair<int,int>>(n));
48     vector<tuple<int,int,int>> where(m);
49     for (int i=0; i<m; i++) {
50         int x, y, dir;
51         cin >> x >> y >> dir;
52         a[x-1][y-1].push_back(make_pair(i, dir-1));
53         where[i] = make_tuple(x-1, y-1, (int)a[x-1][y-1].size()-1);
54     }
55     for (int turn=1; turn<=1000; turn++) {
56         for (int k=0; k<m; k++) {
57             int x, y, index;
58             tie(x,y,index) = where[k];
59             assert(a[x][y][index].first == k);
60             int dir = a[x][y][index].second;
61             int nx = x+dx[dir];
62             int ny = y+dy[dir];
63             if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
64                 if (board[nx][ny] == 2) {
65                     a[x][y][index].second = opposite(dir);
66                 }
67             } else {
68                 a[x][y][index].second = opposite(dir);
69             }
70             dir = a[x][y][index].second;
71             nx = x+dx[dir];
72             ny = y+dy[dir];
73             if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
74                 if (board[nx][ny] == 0) {
75                     go(a, where, x, y, nx, ny, index);
76                 } else if (board[nx][ny] == 1) {
77                     reverse(a[x][y].begin()+index, a[x][y].end());
78                     go(a, where, x, y, nx, ny, index);
79                 }
80                 if (a[nx][ny].size() >= 4) {
81                     cout << turn << '\n';
82                     return 0;
83                 }
84             } else { // out
85             }
86             //cout << "#" << turn << ' ' << k << '\n';
87             //print(a);
88         }
89     }
90     cout << -1 << '\n';
91     return 0;
92 }
```

where[i] = tie(x,y,index)의 의미
(x,y,index)의 세 값을 묶어서 tuple로 만든다
이렇게 만든 tuple을 where라는 벡터에 저장한다

Java

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.util.ArrayList;
5 import java.util.StringTokenizer;
6
7 public class Main {
8     static int N, K;
9     static final int WHITE = 0;
10    static final int RED = 1;
11    static final int BLUE = 2;
12    static int[][] map;
13    static ArrayList<Integer>[][] arr;
14    static ArrayList<Horse> horse_list;
15    static int[] di = { 0, 0, 0, -1, 1 };
16    static int[] dj = { 0, 1, -1, 0, 0 };
17
18    public static void main(String[] args) throws IOException {
19        init();
20        solution();
21    }
22
23    private static void init() throws IOException {
24        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
25        StringTokenizer st = new StringTokenizer(br.readLine());
26        N = Integer.parseInt(st.nextToken());
27        K = Integer.parseInt(st.nextToken());
28        map = new int[N + 2][N + 2];
29        arr = new ArrayList[N + 2][N + 2];
30        horse_list = new ArrayList<>();
31        for (int j = 0; j < N + 2; j++) {
32            map[0][j] = BLUE;
33            map[j][0] = BLUE;
34            map[N + 1][j] = BLUE;
35            map[j][N + 1] = BLUE;
36        }
37        // 맵에 대한 정보와 2차원 어레이리스트 배열 초기화
38        for (int i = 1; i <= N; i++) {
39            st = new StringTokenizer(br.readLine());
40            for (int j = 1; j <= N; j++) {
41                arr[i][j] = new ArrayList<>();
42                map[i][j] = Integer.parseInt(st.nextToken());
43            }
44        }
45        // 말 리스트 초기화
46        horse_list.add(new Horse(0, -1, -1, -1)); // 의미없는 말 추가 (인덱스맞추기)
47        for (int i = 1; i <= K; i++) {
48            st = new StringTokenizer(br.readLine());
49            int ni = Integer.parseInt(st.nextToken());
50            int nj = Integer.parseInt(st.nextToken());
51            int dir = Integer.parseInt(st.nextToken());
52            Horse h = new Horse(i, ni, nj, dir);
53            horse_list.add(h);
54            arr[ni][nj].add(i); // [ni][nj] 번째는 i번째 말이 들어간다.
55        }
56    }
57
58    private static void solution() {
59        int turn = 0;
60        while (true) {
61            turn++;
62            boolean isend = false;
63            for (int h = 1; h <= K; h++) {
64                Horse now = horse_list.get(h); // 현재 말을 받아 움직일려는 쪽이여딘지박아야함.
65                int nexti = now.i + di[now.dir];
66                int nextj = now.j + dj[now.dir];
67                int color = map[nexti][nextj]; // 움직일려는 칸의 색깔.
68                switch (color) {
69                    case WHITE: // 하얀색이면,
70                        white(now);
71                        break;
72                    case RED:
73                        red(now);
74                        break;
75                    case BLUE:
76                        nexti = now.i - di[now.dir];
77                        nextj = now.j - dj[now.dir];
78                        // 반대방향으로 한칸뒤로 움직이게끔 바꿔줌.
79                        if (map[nexti][nextj] == WHITE) {
80                            now.dir = dirReverse(now.dir);
81                            white(now);
82                        } else if (map[nexti][nextj] == RED) {
83                            now.dir = dirReverse(now.dir);
84                            red(now);
85                        } else { // 또 블루면 이동 안하고 방향만 바꿔주기.
86                            now.dir = dirReverse(now.dir);
87                        }
88                        break;
89                }
90                if (size() >= 4) {
91                    System.out.println(turn);
92                    isend = true;
93                    break;
94                }
95            }
96            if (isend) {
97                break;
98            }
99        }
100        if (turn > 1000) {
101            System.out.println(-1);
102            break;
103        }
104    }
105 }
106
107 private static int size() {
108     int cnt = 0;
109     for (int i = 1; i < horse_list.size(); i++) {
110         cnt = Math.max(arr[horse_list.get(i).i][horse_list.get(i).j].size(), cnt);
111     }
112     return cnt;
113 }
114
115 private static int dirReverse(int dir) {
116     if (dir == 1) {
117         return 2;
118     } else if (dir == 2) {
119         return 1;
120     } else if (dir == 3) {
121         return 4;
122     } else {
123         return 3;
124     }
125 }
126
127 private static void red(Horse horse) {
128     ArrayList<Integer> temp = new ArrayList<>();
129     // 현재 위치의 배열에서 horse 인덱스번째를 찾는다.
130     int size = arr[horse.i][horse.j].size();
131     for (int i = size - 1; i >= 0; i--) {
132         int removeIDX = arr[horse.i][horse.j].remove(i);
133         if (removeIDX == horse.idx) {
134             temp.add(removeIDX);
135             break;
136         } else {
137             temp.add(removeIDX);
138         }
139     }
140
141     int nexti = horse.i + di[horse.dir];
142     int nextj = horse.j + dj[horse.dir];
143     for (int i = 0; i < temp.size(); i++) {
144         int idx = temp.get(i);
145         arr[nexti][nextj].add(idx);
146         horse_list.get(idx).i = nexti;
147         horse_list.get(idx).j = nextj;
148     }
149 }
150
151 private static void white(Horse horse) {
152     // 내가 움직일려는 nexti, nextj
153     int nexti = horse.i + di[horse.dir];
154     int nextj = horse.j + dj[horse.dir];
155     ArrayList<Integer> temp = new ArrayList<>();
156     int size = arr[horse.i][horse.j].size();
157     for (int i = size - 1; i >= 0; i--) {
158         int removeIDX = arr[horse.i][horse.j].remove(i);
159         if (removeIDX == horse.idx) {
160             temp.add(removeIDX);
161             break;
162         } else {
163             temp.add(removeIDX);
164         }
165     }
166
167     for (int i = 0; i < temp.size(); i++) {
168         int idx = temp.get(temp.size() - i - 1);
169         arr[nexti][nextj].add(idx);
170         horse_list.get(idx).i = nexti;
171         horse_list.get(idx).j = nextj;
172     }
173 }
174
175 static class Horse {
176     int idx;
177     int i;
178     int j;
179     int dir;
180
181     Horse(int idx, int i, int j, int dir) {
182         this.idx = idx;
183         this.i = i;
184         this.j = j;
185         this.dir = dir;
186     }
187 }
188 }
```

Dynamic programming

1450142. 회사

$1일 \sim N일$

$1일$

$\frac{1[1]일}{p[1]일}$ 기각

$1일$

$(N+1)일$ 회사

$1 \leq N \leq 15$

$3[1]일$

백준이는 비서에게 최대한 많은 상담을 잡으라고 부탁을 했고, 비서는 하루에 하나씩 서로 다른 사람의 상담을 잡아놓았다.

$$2^{16} = 65536$$

N일

각각의 상담은 상담을 완료하는데 걸리는 시간 T_i 와 상담을 했을 때 받을 수 있는 금액 P_i 로 이루어져 있다.

N = 7인 경우에 다음과 같은 상담 일정표를 보자.

	1일	2일	3일
T_i	3	5	1
P_i	10	20	10

4일	5일	6일	7일
1	2	4	2
20	15	40	200

1일에 잡혀있는 상담은 총 3일이 걸리며, 상담했을 때 받을 수 있는 금액은 10이다. 5일에 잡혀있는 상담은 총 2일이 걸리며, 받을 수 있는 금액은 15이다.

상담을 하는데 필요한 기간은 1일보다 클 수 있기 때문에, 모든 상담을 할 수는 없다. 예를 들어서 1일(상담을 하게 되면, 2일, 3일에 있는 상담은 할 수 없게 된다. 2일에 있는 상담을 하게 되면, 3, 4, 5, 6일

방향을 수: (2^N)

8일
5/1/1

1042

$$2^{20} = 1048576$$

$$2^{15} = 32768$$

$$2^{31} - 1 = 2147483647$$

$$2^{31}$$

C++

long long

Java

long

$$\log_2 + \log_2$$

Gyal Notes 5

$$2^{63} - 1 = ?$$

4

Java

```
1 import java.util.*;
2 public class Main {
3     static int n;
4     static int[] t;
5     static int[] p;
6     static int ans=0;
7     static void go(int index, int sum) {
8         if (index == n) {
9             if (ans < sum) {
10                 ans = sum;
11             }
12             return;
13         }
14         if (index > n) {
15             return;
16         }
17         go(index+t[index], sum + p[index]);
18         go(index+1, sum);
19     }
20     public static void main(String[] args) {
21         Scanner sc = new Scanner(System.in);
22         n = sc.nextInt();
23         t = new int[n];
24         p = new int[n];
25         for (int i=0; i<n; i++) {
26             t[i] = sc.nextInt();
27             p[i] = sc.nextInt();
28         }
29         go(0,0);
30         System.out.println(ans);
31     }
32 }
```

$(0 \sim n-1)$ ↑
퇴사
 $O(2^N)$

5개 할
15개 할

Java

```

1 import java.util.*;
2 public class Main {
3     static int n;
4     static int[] t;
5     static int[] p;
6     static int go(int index, int sum) {
7         if (index == n) {
8             return sum;
9         }
10        if (index > n) {
11            return 0;
12        }
13        int t1 = go(index+t[index], sum + p[index]);
14        int t2 = go(index+1, sum);
15        return Math.max(t1, t2);
16    }
17    public static void main(String[] args) {
18        Scanner sc = new Scanner(System.in);
19        n = sc.nextInt();
20        t = new int[n];
21        p = new int[n];
22        for (int i=0; i<n; i++) {
23            t[i] = sc.nextInt();
24            p[i] = sc.nextInt();
25        }
26        System.out.println(go(0,0));
27    }
28 }

```

$go(index)$
 $= index \times \text{배리}$
 상한 값을 더
 최대

$sum + p[index]$
 Sum-

결과

메모리

시간

코드 길이

채점 중

700 B

Java

```

1 import java.util.*;
2 public class Main {
3     static int n;
4     static int[] t;
5     static int[] p;
6     // 0~(n-1), 퇴사: n일
7     // go(index) = index일부터 얻을수 있는 최대 수익
8     static int go(int index) {
9         if (index == n) { // 퇴사하는 날
10             return 0;
11         }
12         if (index > n) { // 조건 위배 (퇴사날짜 넘김)
13             return -1000000000;
14         }
15         // index index+t[index] ... n
16         // -----
17         // p[index] go(index+t[index])
18         int t1 = p[index] + go(index+t[index]);
19         int t2 = go(index+1);
20         return Math.max(t1, t2);
21     }
22     public static void main(String[] args) {
23         Scanner sc = new Scanner(System.in);
24         n = sc.nextInt();
25         t = new int[n];
26         p = new int[n];
27         for (int i=0; i<n; i++) {
28             t[i] = sc.nextInt();
29             p[i] = sc.nextInt();
30         }
31         System.out.println(go(0));
32     }
33 }

```

결과

메모리

시간

코드 길이

채점 중

961 B



50170

201711

55172

22

50170

22

21

22

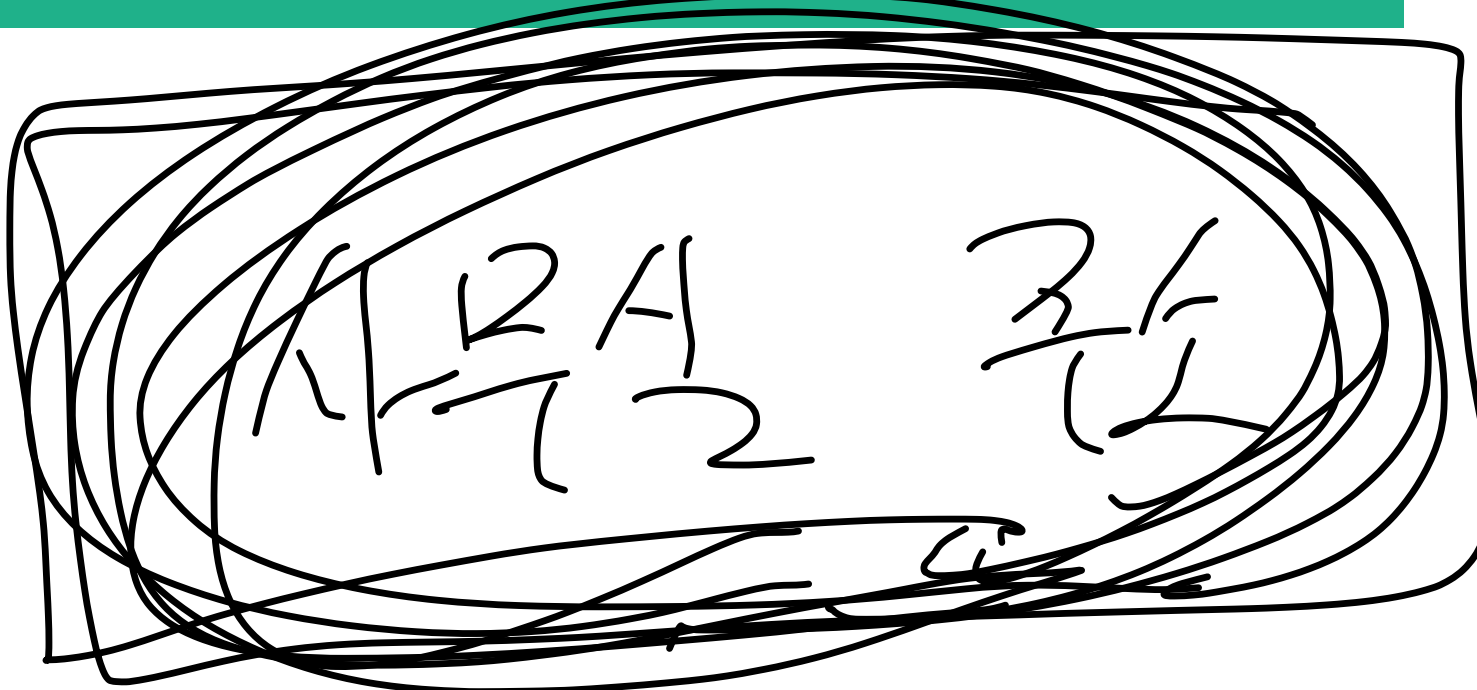
22

Memorization

24 / 20

2m에 2840

2m에 2840 2840 2840

A large, hand-drawn rectangular box containing a dense, overlapping scribble of black lines. Within the scribble, the text 'A12A 3/1' is visible, written in a simple, handwritten style. The 'A' is followed by a vertical line, then '12', then another vertical line, then 'A'. To the right of this sequence is '3/1' with a vertical line underneath the '1'. The entire content is heavily obscured by the scribbles.

A handwritten note consisting of the date '4/25' circled in black ink.

— 2 2 2 2

~~326 424~~

32: 1명이 지분을 22% 22% 22% 22% 22%

7232 73

6

5

②



1

1

2/17/20

450 AP2

~~1132~~

1721

МЗГ МЗГБНЧЗ 2

12/22/2024

~~2/4~~

\exists $\{x_1, \dots, x_n\}$ \exists

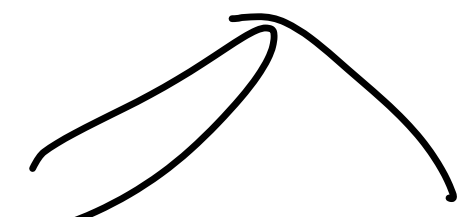
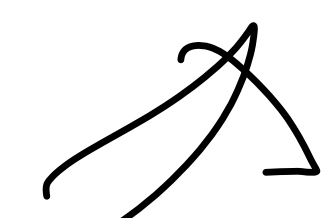
$\leq N$

N $\{x_1, \dots, x_n\}$

\exists $\{x_1, \dots, x_n\}$ \exists

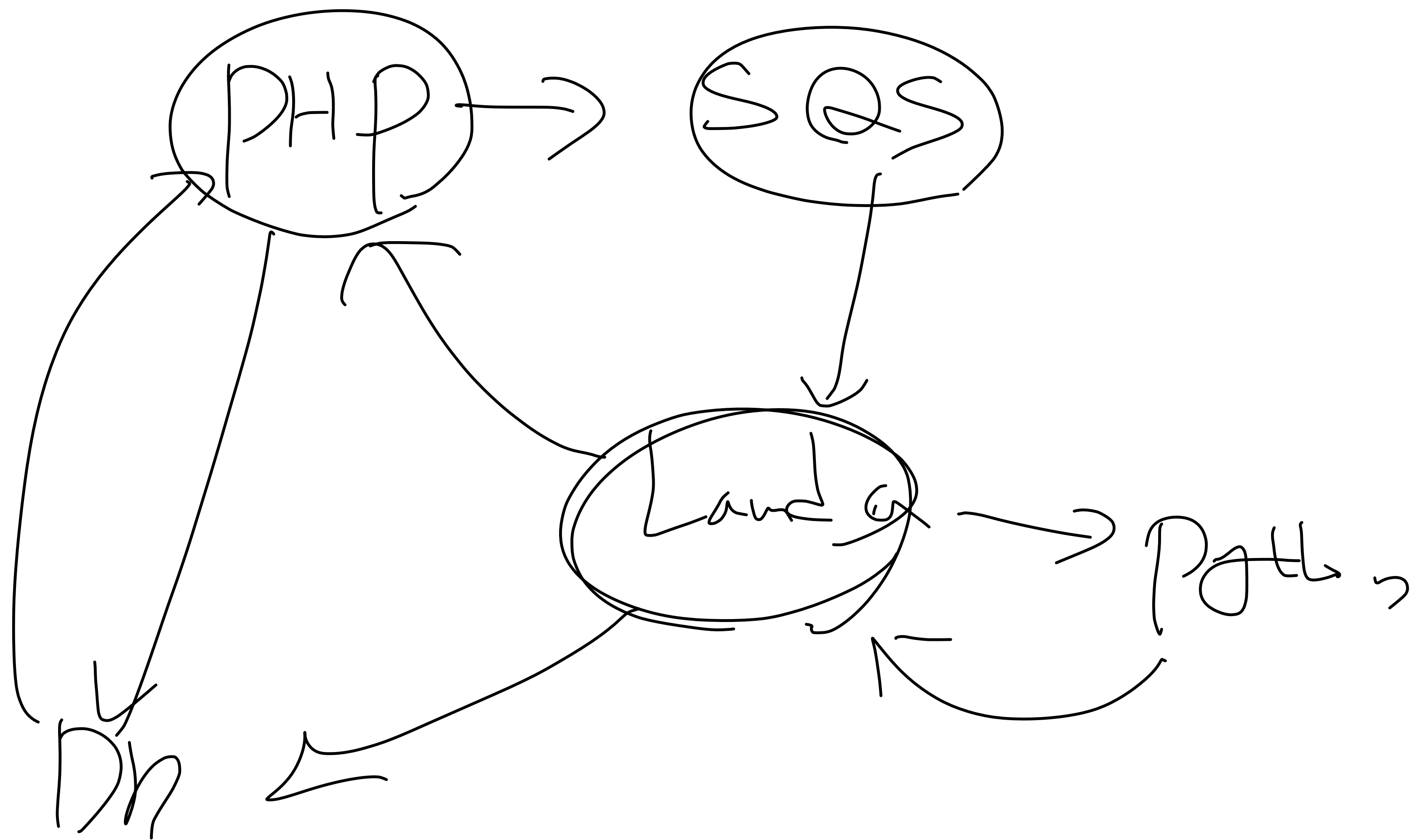
\exists $\{x_1, \dots, x_n\}$

\exists $\{x_1, \dots, x_n\}$



Cache

Pedis
Mem



1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

HHH 1 : 50%

HHH 2 : 75%

HHH 3 : 30%

(50%)


~~100%~~

Java

```

1 import java.util.*;
2 import java.io.*;
3 public class Main {
4     static int n;
5     static int[] t;
6     static int[] p;
7     static int[] d;
8     // 0~(n-1), 퇴사: n일
9     // go(index) = index일부터 얻을수 있는 최대 수익
10    static int go(int index) {
11        if (index == n) { // 퇴사하는 날
12            return 0;
13        }
14        if (index > n) { // 조건 위배 (퇴사날짜 넘김)
15            return -1000000000;
16        }
17        if (d[index] >= 0) return d[index];
18        // index index+t[index] ... n
19        // -----
20        //.    p[index]    go(index+t[index])
21        int t1 = p[index] + go(index+t[index]);
22        int t2 = go(index+1);
23        d[index] = Math.max(t1, t2);
24        return d[index];
25    }
26    public static void main(String[] args) throws IOException {
27        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
28        n = Integer.parseInt(bf.readLine());
29        t = new int[n];
30        p = new int[n];
31        d = new int[n];
32        for (int i=0; i<n; i++) {
33            String[] temp = bf.readLine().split(" ");
34            t[i] = Integer.parseInt(temp[0]);
35            p[i] = Integer.parseInt(temp[1]);
36            d[i] = -1;
37        }
38        System.out.println(go(0));
39    }
40 }

```



결과

메모리

시간

코드 길이

채점 중

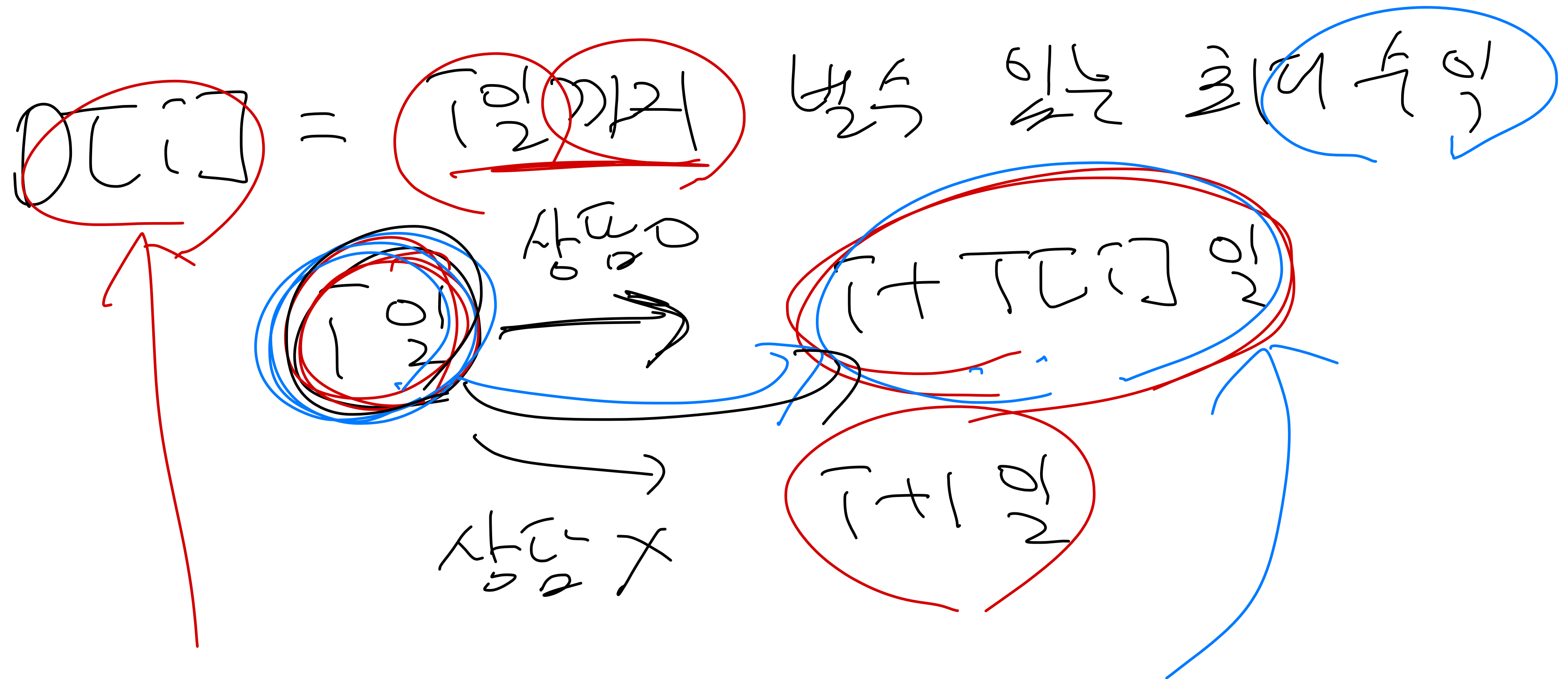
1274 B

Dynamic Programming

Top Down



Top Down
Bottom Up



$$D[\bar{i} + D[\bar{i}]] = D[\bar{i}] + P[\bar{i}]$$

$$D[\bar{i} + 1] = D[\bar{i}]$$

Java

```
1 import java.util.*;
2 import java.io.*;
3 public class Main {
4     static int n;
5     static int[] t;
6     static int[] p;
7     static int[] d;
8     public static void main(String[] args) throws IOException {
9         BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
10        n = Integer.parseInt(bf.readLine());
11        t = new int[n];
12        p = new int[n];
13        d = new int[n+1000];
14        for (int i=0; i<n; i++) {
15            String[] temp = bf.readLine().split(" ");
16            t[i] = Integer.parseInt(temp[0]);
17            p[i] = Integer.parseInt(temp[1]);
18        }
19        for(int i=0; i<n; i++) {
20            d[i+t[i]] = Math.max(d[i+t[i]], d[i]+p[i]);
21            d[i+1] = Math.max(d[i+1], d[i]);
22        }
23        System.out.println(d[n]);
24    }
25 }
```

결과

메모리

시간

코드 길이

채점 중

779 B

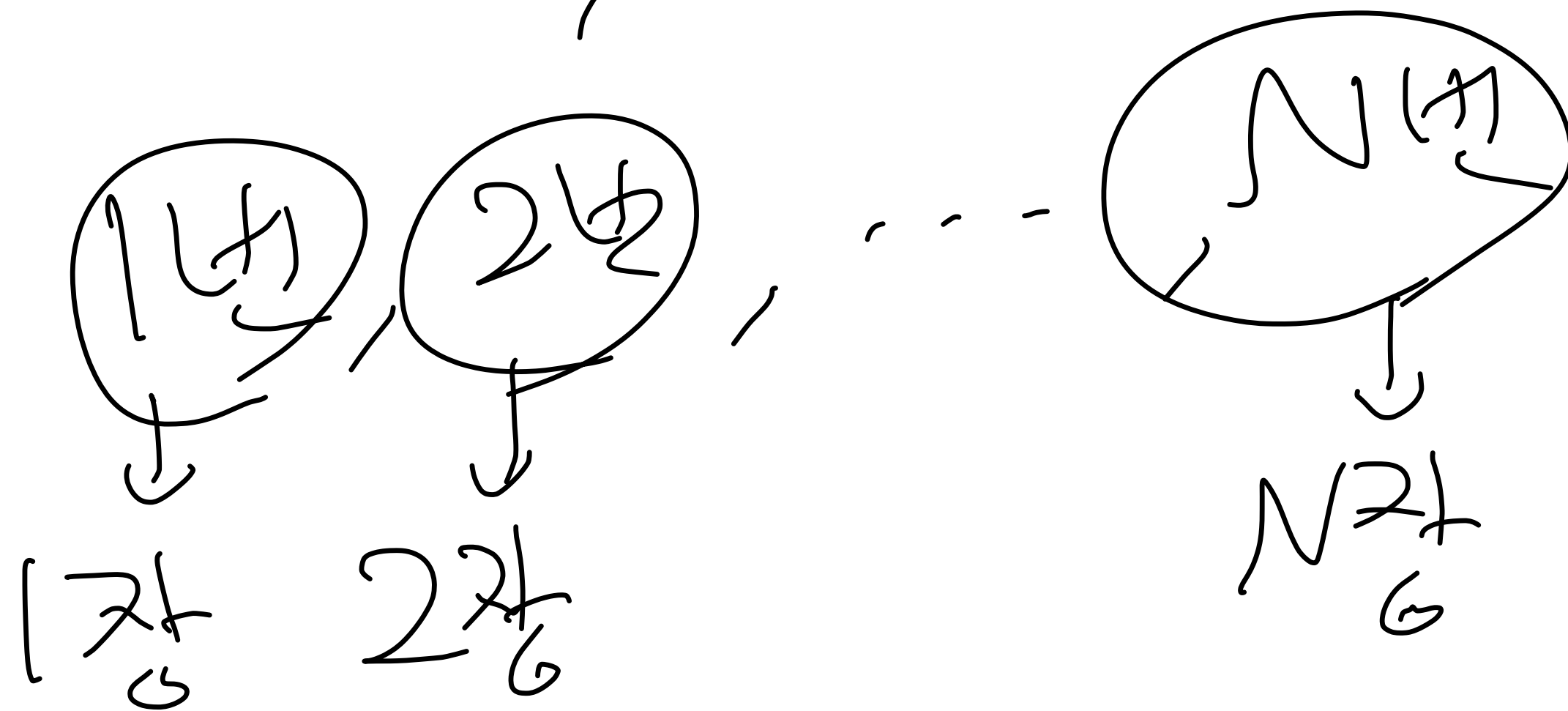
110524 : 카드 구매금)

카드금 N개 구매 회비비중
N개

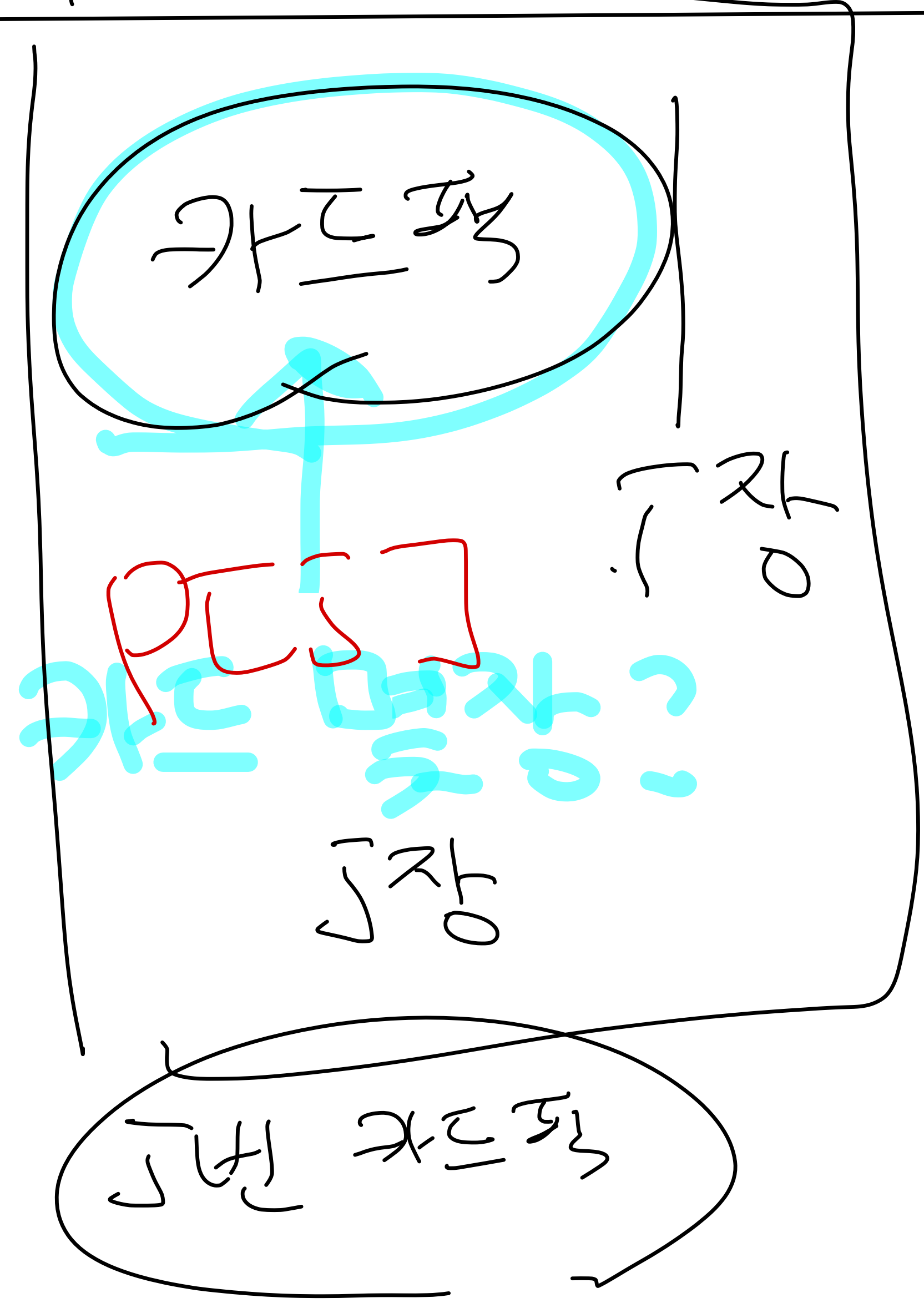
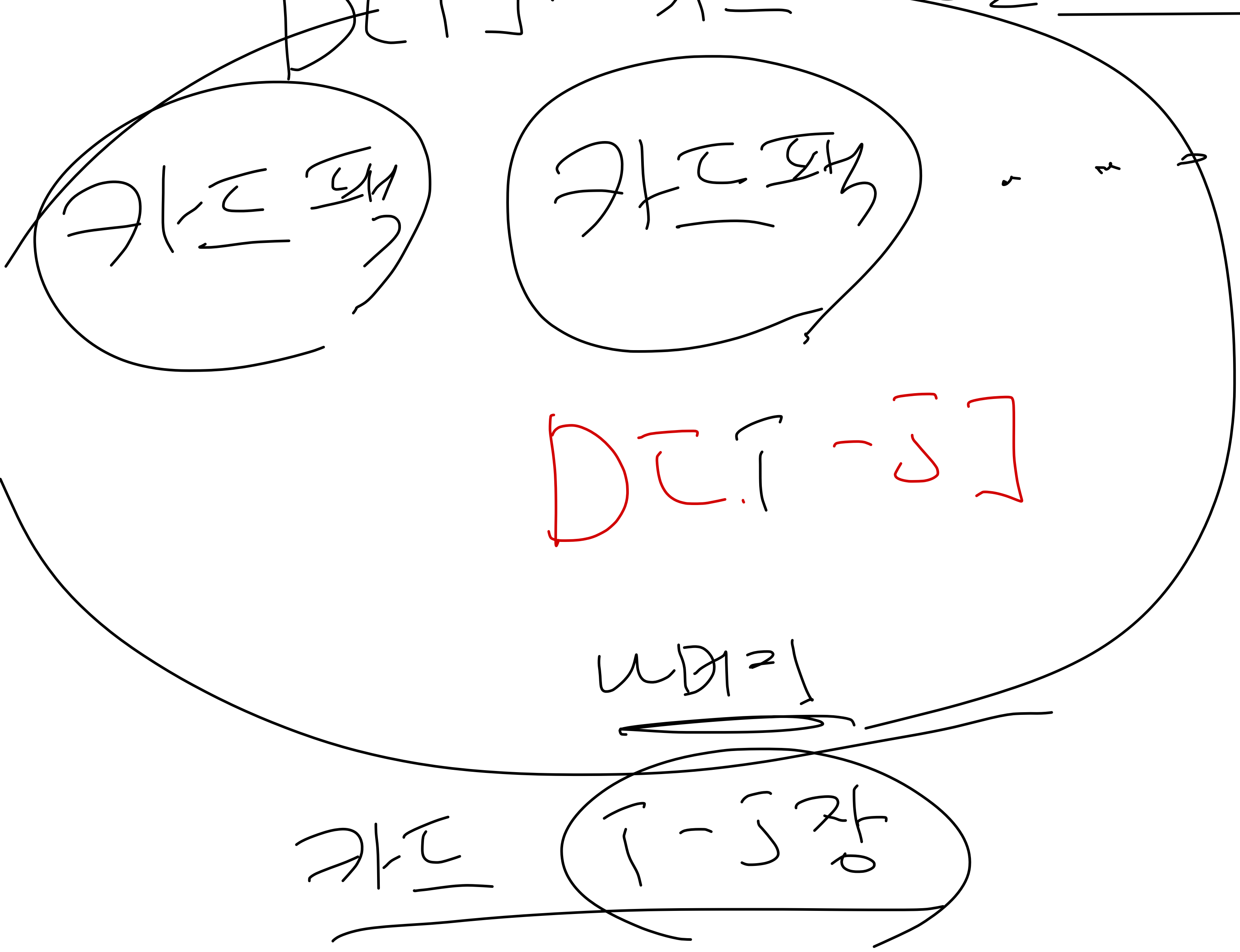
카드 구매

→ 카드가 들어있음

1번: PIN



DEF = 카드 - 장을 구매하는 3/4 4/5



DP[i] = 카드 i 장을 꾸미하는 최대 비용

$$= \max(DP[i-j] + p[j])$$

$(1 \leq j \leq i)$

$$\frac{(\text{문제의 수}) \times (\text{문제 1개를 꾸미는 시간})}{N} = O(N)$$

$$O(N^2)$$

Java

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int n = sc.nextInt();
6         int[] d = new int[n+1];
7         int[] p = new int[n+1];
8         for (int i=1; i<=n; i++) {
9             p[i] = sc.nextInt();
10        }
11        for (int i=1; i<=n; i++) {
12            for (int j=1; j<=i; j++) {
13                d[i] = Math.max(d[i], d[i-j]+p[j]);
14            }
15        }
16        System.out.println(d[n]);
17    }
18 }
```

최소
d[i] = -1
d[i] = 10⁹
min

결과	메모리	시간	코드 길이
맞았습니다!!	18696 KB	192 ms	493 B

Java

```
1 import java.util.*;
2 public class Main {
3     static int n;
4     static int[] d, p;
5     static int go(int i) {
6         if (i == 0) return 0;
7         if (d[i] > 0) return d[i];
8         for (int j=1; j<=i; j++) {
9             d[i] = Math.max(d[i], go(i-j)+p[j]);
10        }
11        return d[i];
12    }
13    public static void main(String[] args) {
14        Scanner sc = new Scanner(System.in);
15        n = sc.nextInt();
16        d = new int[n+1];
17        p = new int[n+1];
18        for (int i=1; i<=n; i++) {
19            p[i] = sc.nextInt();
20        }
21        System.out.println(go(n));
22    }
23 }
```

결과

메모리

시간

코드 길이

채점 중

582 B

16987 제1항으로 제2항>1

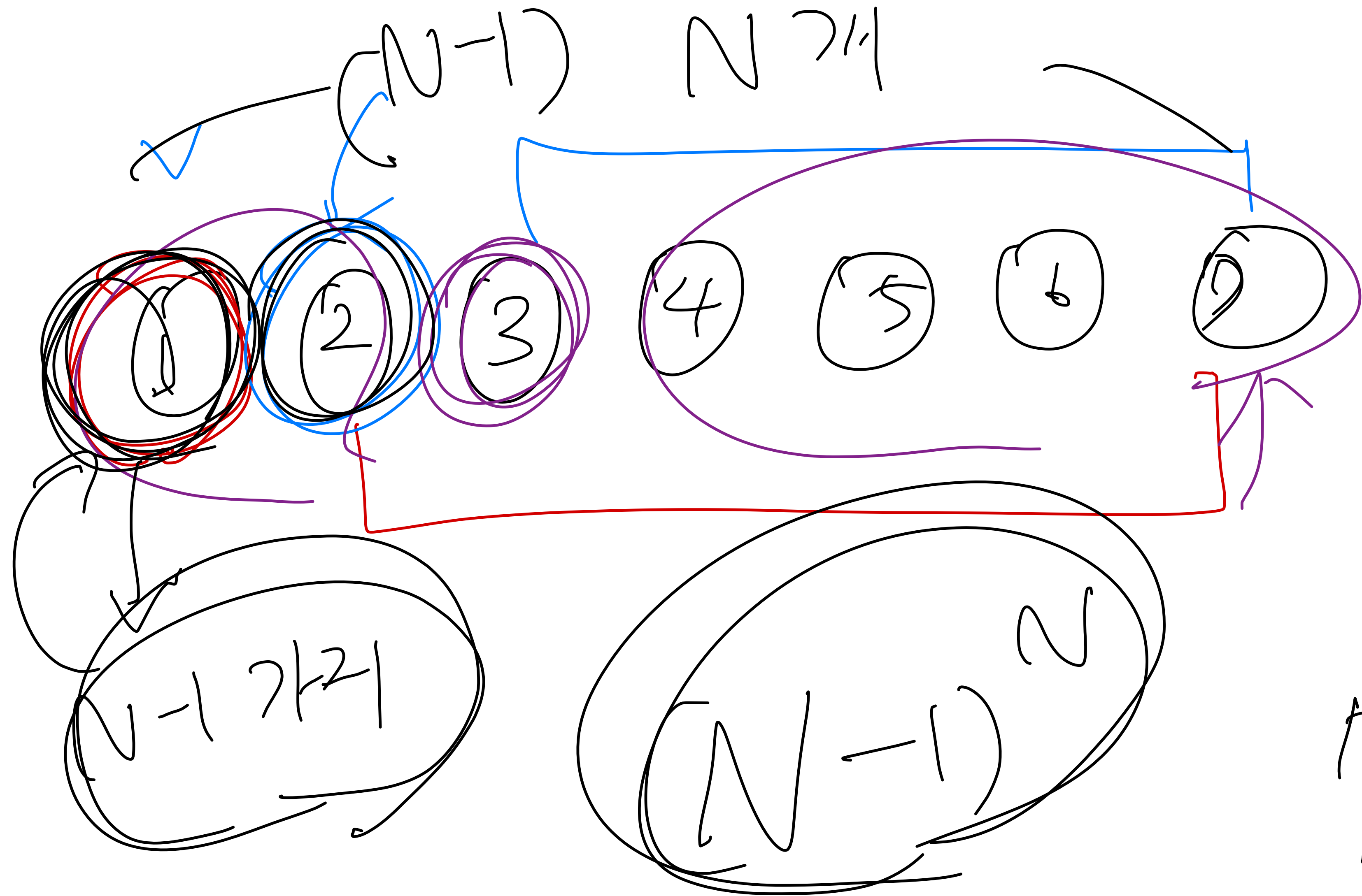
제1항 A3 제2항 B

제1항
제2항

B의 제1항 = A의 제2항

A의 제1항 = B의 제2항

제1항 $\leq 0 \Rightarrow$ 제2항 ≥ 0



$$N \leq 8$$

$$7^8 = 5764801$$

① 깨지리 _{이름} _{이름} 제_{이름}을 _{이름} 친다

② $\frac{1}{2}$ _{이름} _{이름} 제_{이름} _{이름} 깨_{이름} _{이름}

③ 깨_{이름} _{이름} _{이름} 제_{이름} _{이름} _{이름} _{이름}

Java

```
1 import java.util.*;
2 public class Main {
3     static int n; // 0 1 ... n-1 (n)
4     static int[] s; // 내구도
5     static int[] w; // 무게
6     static int ans = 0;
7     static void go(int index) {
8         if (index == n) {
9             // 깨진 계란의 수 계산
10            int cnt = 0;
11            for (int i=0; i<n; i++) {
12                if (s[i] <= 0) {
13                    cnt += 1;
14                }
15            }
16            if (ans < cnt) {
17                ans = cnt;
18            }
19            return;
20        }
21        if (s[index] <= 0) {
22            go(index+1);
23            return;
24        }
25        boolean ok = false;
26        for (int i=0; i<n; i++) {
27            // index -> i
28            if (index == i) continue;
29            if (s[i] <= 0) continue;
30            s[index] -= w[i];
31            s[i] -= w[index];
32            go(index+1);
33            s[i] += w[index];
34            s[index] += w[i];
35            ok = true;
36        }
37        if (!ok) go(index+1);
38    }
39    public static void main(String[] args) {
40        Scanner sc = new Scanner(System.in);
41        n = sc.nextInt();
42        s = new int[n];
43        w = new int[n];
44        for (int i=0; i<n; i++) {
45            s[i] = sc.nextInt();
46            w[i] = sc.nextInt();
47        }
48        go(0);
49        System.out.println(ans);
50    }
51 }
```

결과

메모리

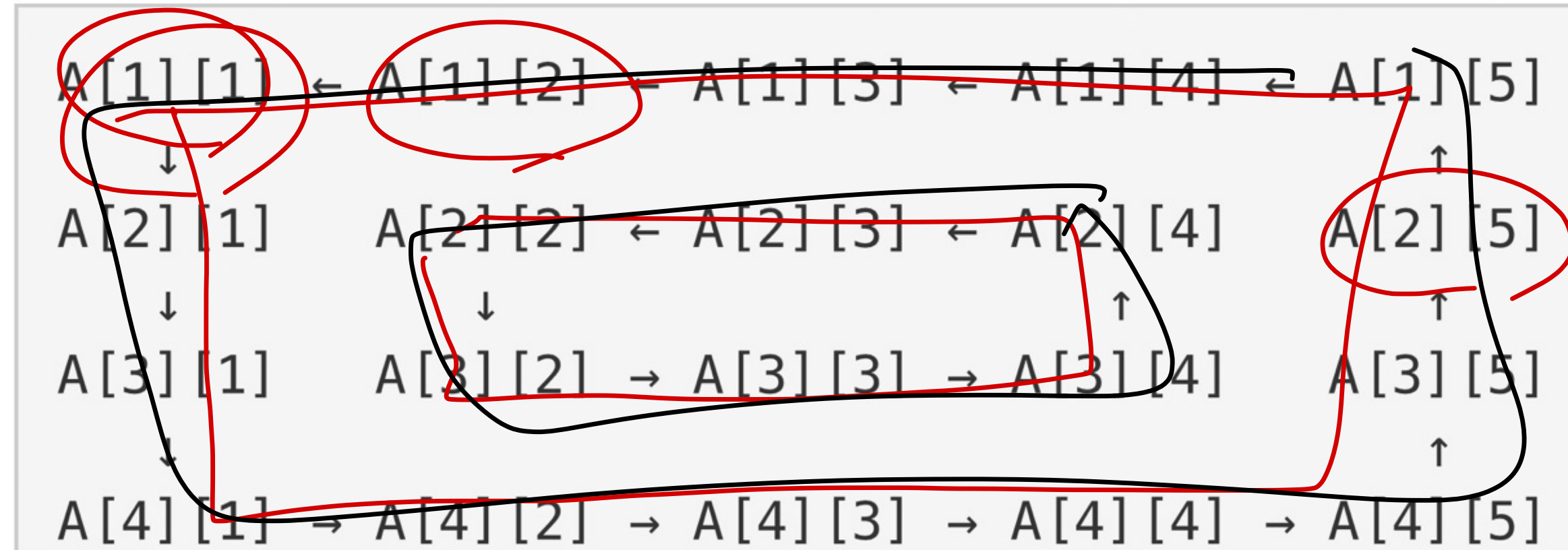
시간

코드 길이

문제

16426
2 2/4

크기가 $N \times M$ 인 배열이 있을 때, 배열을 돌려보려고 한다. 배열은 다음과 같이 반시계 방향

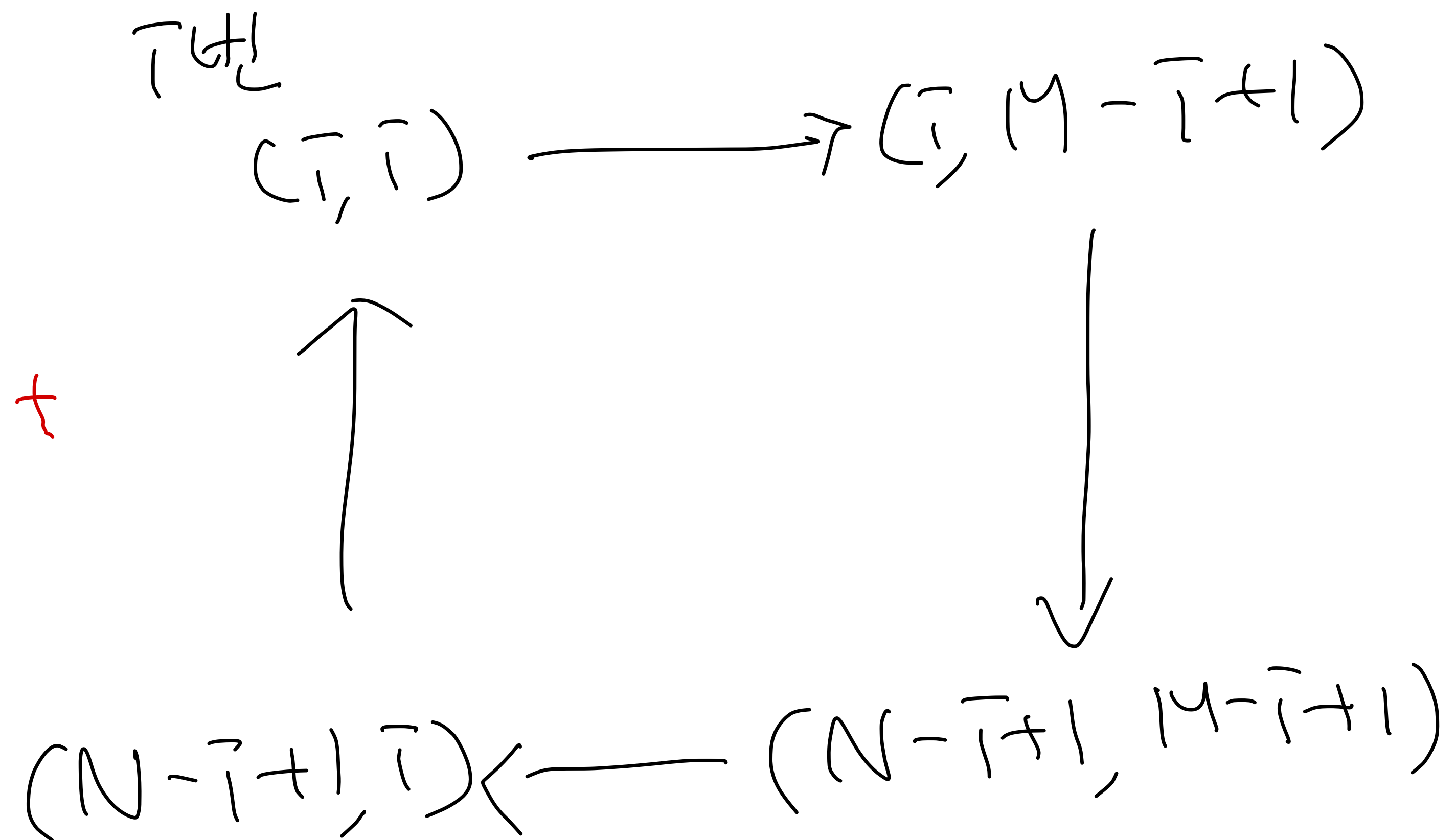
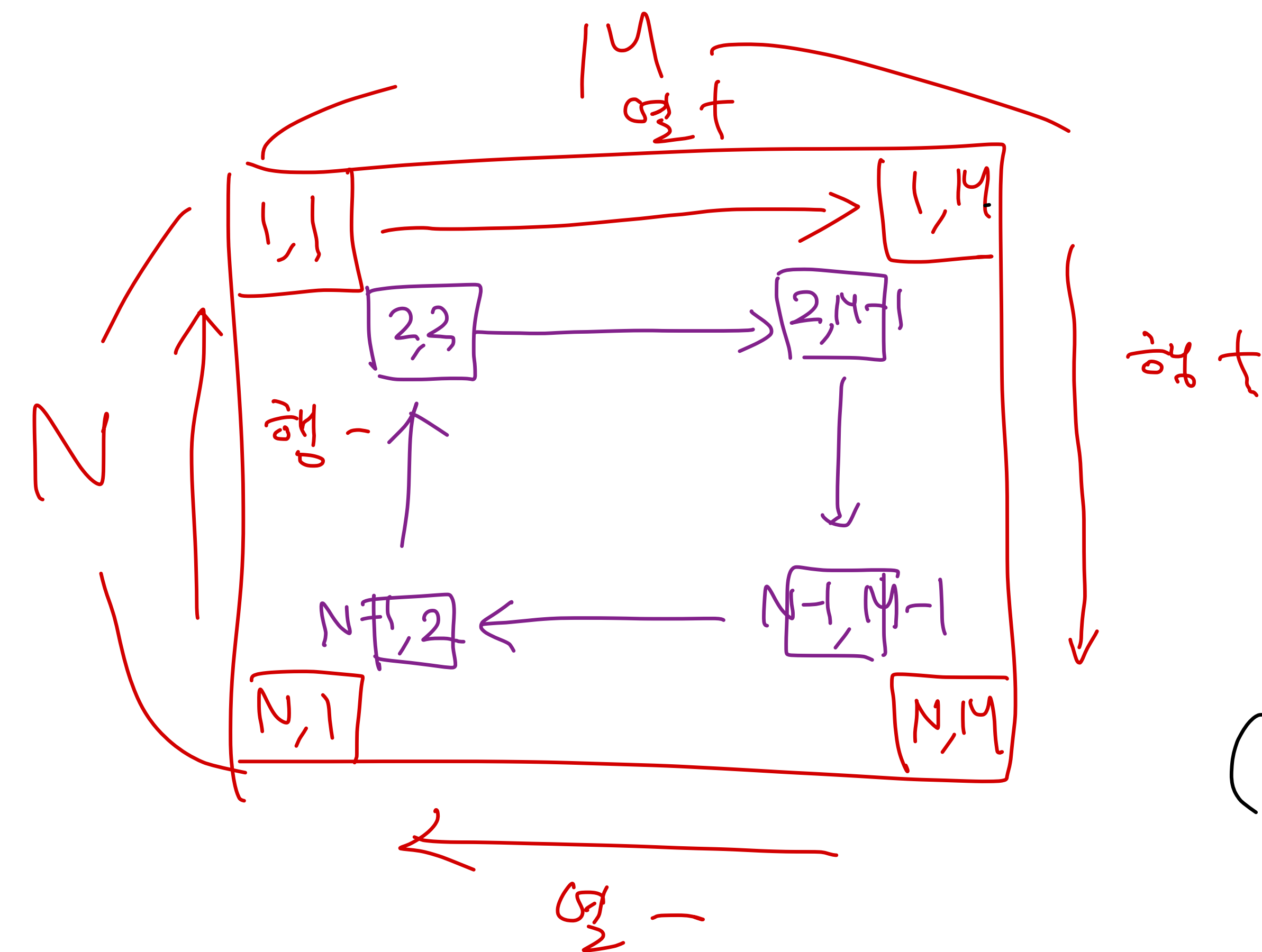


예를 들어, 아래와 같은 배열을 2번 회전시키면 다음과 같이 변하게 된다.

1 2 3 4		2 3 4 8		3 4 8 6
5 6 7 8		1 7 7 6		2 7 8 2
9 8 7 6	→	5 6 8 2	→	1 7 6 3
5 4 3 2		9 5 4 3		5 9 5 4
<시작>		<회전1>		<회전2>

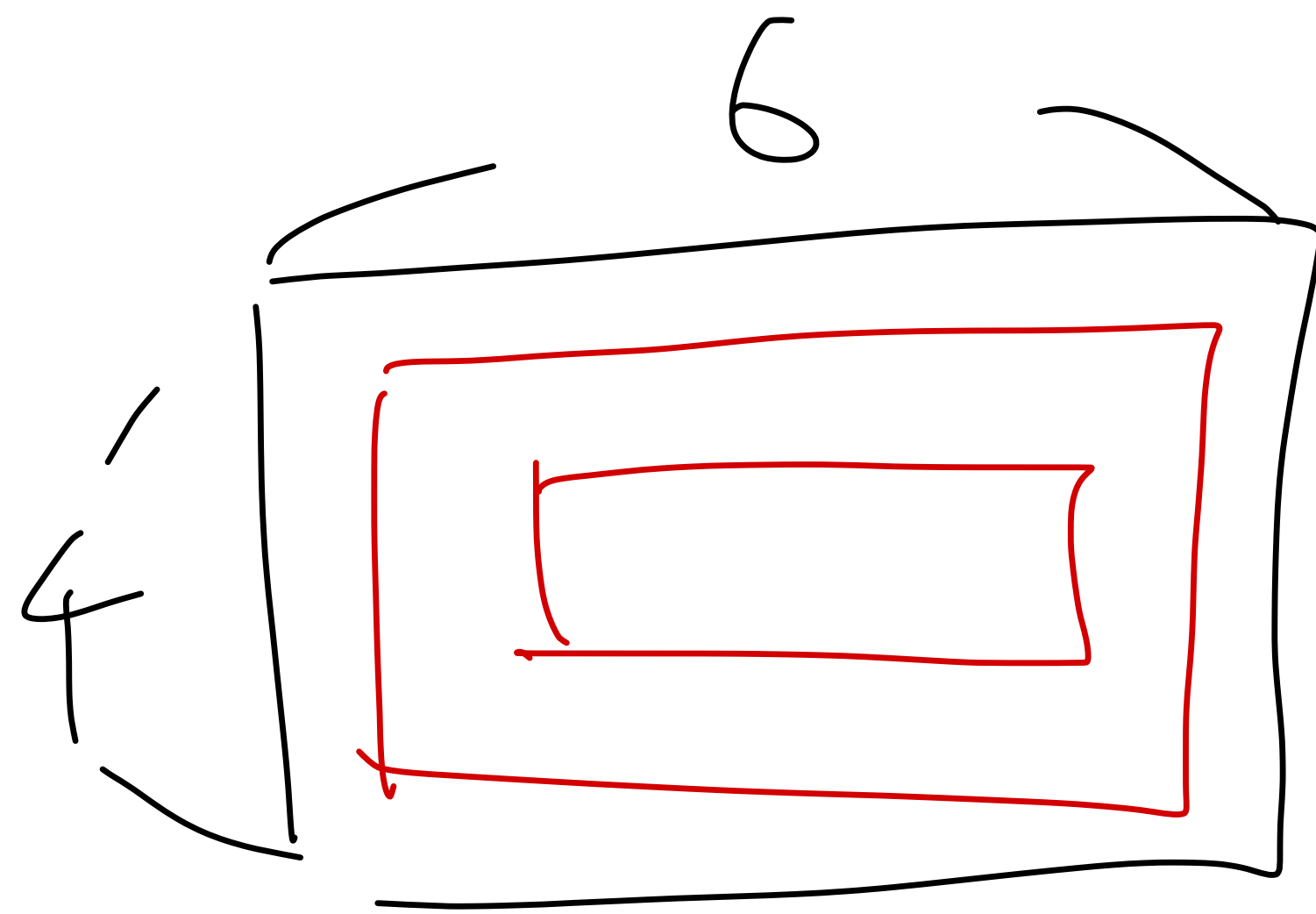
배열의 크기 $N \times M$ 이 주어질 때 배열을 N 번 회전시키게 되는 구조를 구하라

$(1,1); (1,2) (1,3) (1,4) (1,5) (2,5) (3,5)$



Java

```
1 import java.util.*;
2 public class Main {
3     static int n,m,r; // n*m, r
4     static int[][] a;
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         n = sc.nextInt();
8         m = sc.nextInt();
9         r = sc.nextInt();
10        a = new int[n+1][m+1];
11        for (int i=1; i<=n; i++) {
12            for (int j=1; j<=m; j++) {
13                a[i][j] = sc.nextInt();
14            }
15        }
16        int groups = Math.min(n,m)/2;
17        for (int g=1; g<=groups; g++) {
18            // (g,g) -> ... -> (g,m-g+1)
19            // ^
20            // |
21            // |
22            // (n-g+1,g) <- ... <- (n-g+1, m-g+1)
23            ArrayList<Integer> group = new ArrayList<>();
24            for (int j=g; j<=m-g+1; j++) {
25                group.add(a[g][j]);
26            }
27            for (int i=g; i<=n-g+1; i++) {
28                group.add(a[i][m-g+1]);
29            }
30            for (int j=m-g+1; j>g; j--) {
31                group.add(a[n-g+1][j]);
32            }
33            for (int i=n-g+1; i>g; i--) {
34                group.add(a[i][g]);
35            }
36            Collections.rotate(group, (int)group.size()-r);
37            int index = 0;
38            for (int j=g; j<=m-g+1; j++) {
39                a[g][j] = group.get(index++);
40            }
41            for (int i=g; i<=n-g+1; i++) {
42                a[i][m-g+1] = group.get(index++);
43            }
44            for (int j=m-g+1; j>g; j--) {
45                a[n-g+1][j] = group.get(index++);
46            }
47            for (int i=n-g+1; i>g; i--) {
48                a[i][g] = group.get(index++);
49            }
50        }
51        for (int i=1; i<=n; i++) {
52            for (int j=1; j<=m; j++) {
53                System.out.print(a[i][j] + " ");
54            }
55            System.out.println();
56        }
57    }
58 }
```

224

$$\frac{\min(N, M)}{2}$$

