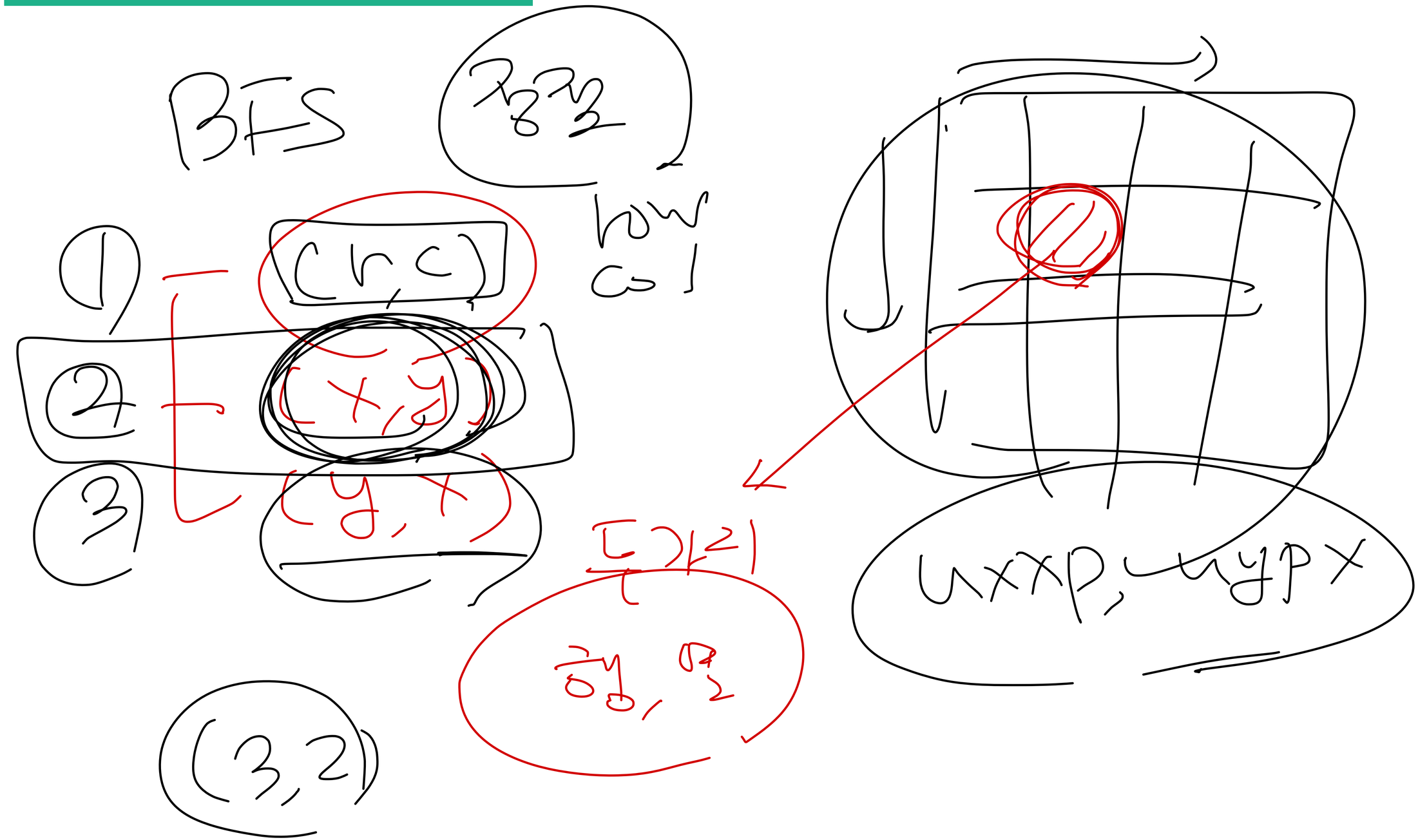
최백준 choi@startlink.io

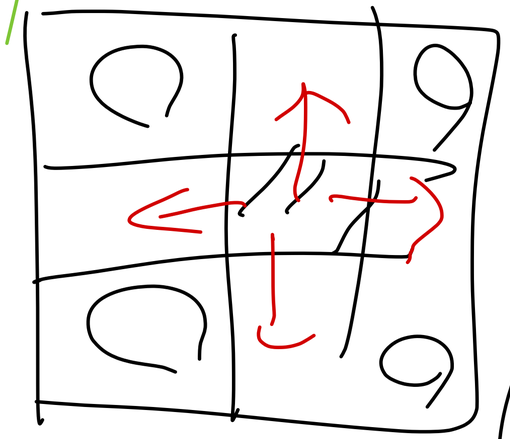
Java

```java
import java.util.*;

public class Main {
    static final int n = 33;
    static int[] score = {
        0,2,4,6,8,
        10,13,16,19,25,
        12,14,16,18,20,
        22,24,22,24,26,
        28,26,27,28,30,
        32,34,36,38,30,
        35,40,0
    };
    static final int t = 10;
    static int[] dice = new int[t];
    static int[][] a = new int[n][];
    static int get_next(int start, int k) {
        int now = start;
        for (int i=0; i<k; i++) {
            if (i == 0) {
                now = a[now][0];
            } else {
                now = a[now][1];
            }
        }
        return now;
    }
    static int go(int index, int[] horse, int sum) {
        if (index == t) return sum;
        int ans = 0;
        for (int i=0; i<4; i++) {
            int nxt = get_next(horse[i], dice[index]);
            boolean ok = true;
            if (nxt != n-1) {
                for (int j=0; j<4; j++) {
                    if (i == j) continue;
                    if (nxt == horse[j]) ok = false;
                }
            }
            if (ok) {
                int[] nhorse = horse.clone();
                nhorse[i] = nxt;
                int temp = go(index+1, nhorse, sum+score[nxt]);
                if (ans < temp) ans = temp;
            }
        }
        return ans;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        a[0] = new int[]{1,1};
        a[1] = new int[]{2,2};
        a[2] = new int[]{3,3};
        a[3] = new int[]{4,4};
        a[4] = new int[]{5,5};
        a[5] = new int[]{6,10};
        a[6] = new int[]{7,7};
        a[7] = new int[]{8,8};
        a[8] = new int[]{9,9};
        a[9] = new int[]{29,29};
        a[10] = new int[]{11,11};
        a[11] = new int[]{12,12};
        a[12] = new int[]{13,13};
        a[13] = new int[]{14,14};
        a[14] = new int[]{15,17};
        a[15] = new int[]{16,16};
        a[16] = new int[]{9,9};
        a[17] = new int[]{18,18};
        a[18] = new int[]{19,19};
        a[19] = new int[]{20,20};
        a[20] = new int[]{24,24};
        a[21] = new int[]{9,9};
        a[22] = new int[]{21,21};
        a[23] = new int[]{22,22};
        a[24] = new int[]{23,25};
        a[25] = new int[]{26,26};
        a[26] = new int[]{27,27};
        a[27] = new int[]{28,28};
        a[28] = new int[]{31,31};
        a[29] = new int[]{30,30};
        a[30] = new int[]{31,31};
        a[31] = new int[]{32,32};
        a[32] = new int[]{32,32};
        for (int i=0; i<t; i++) {
            dice[i] = sc.nextInt();
        }
        System.out.println(go(0, new int[]{0, 0, 0, 0}, 0));
    }
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	37968 KB	192 ms	2645 B


BFS
how
(r,c)
(x,y)
(y,x)
(3,2)



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

$$dx = [-1, 1, 0, 0]$$
$$dy = [0, 0, -1, 1]$$

$nx = x + dx[k]$

$ny = y + dy[k]$

$(x-1, y)$

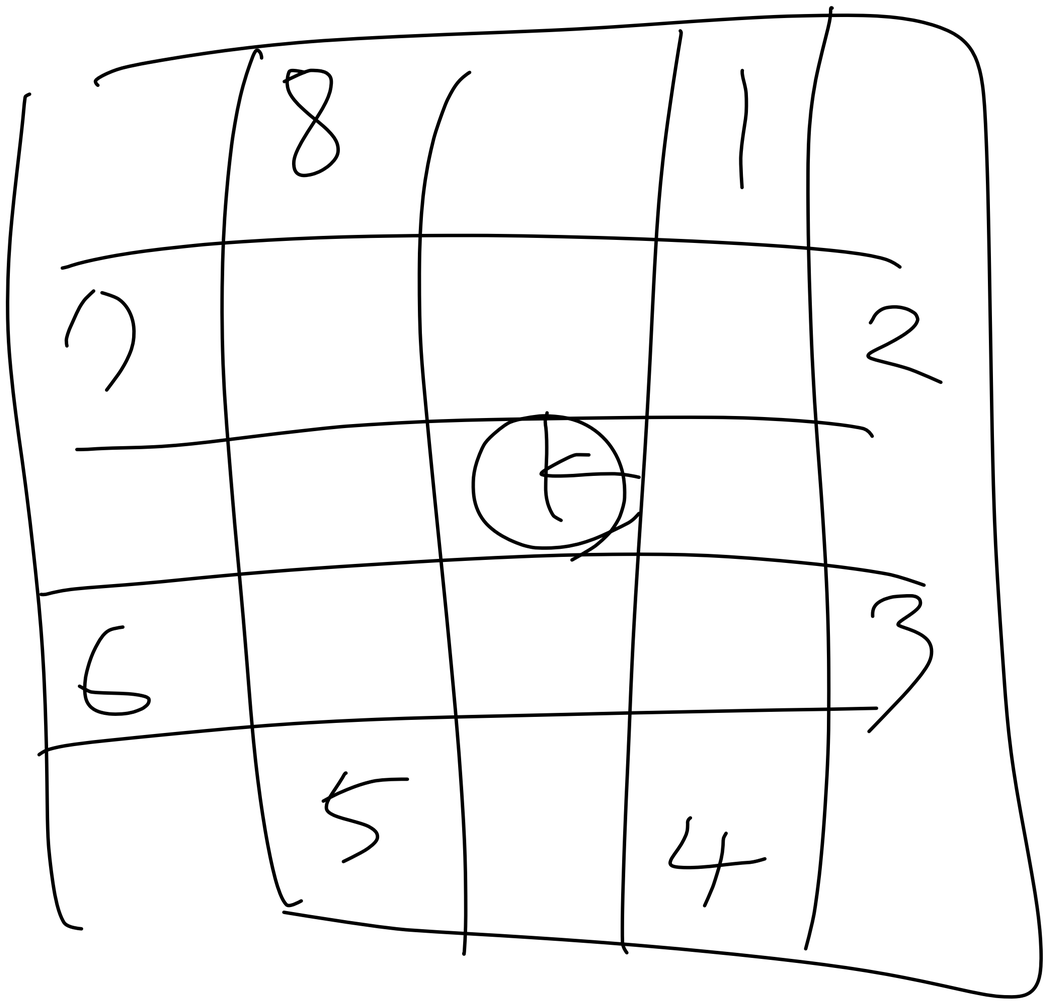
$(x, y-1)$ (x, y) $(x, y+1)$

$(x+1, y)$

$$dx = [1, -1, 0, 0, 1, -1, 1, -1]$$

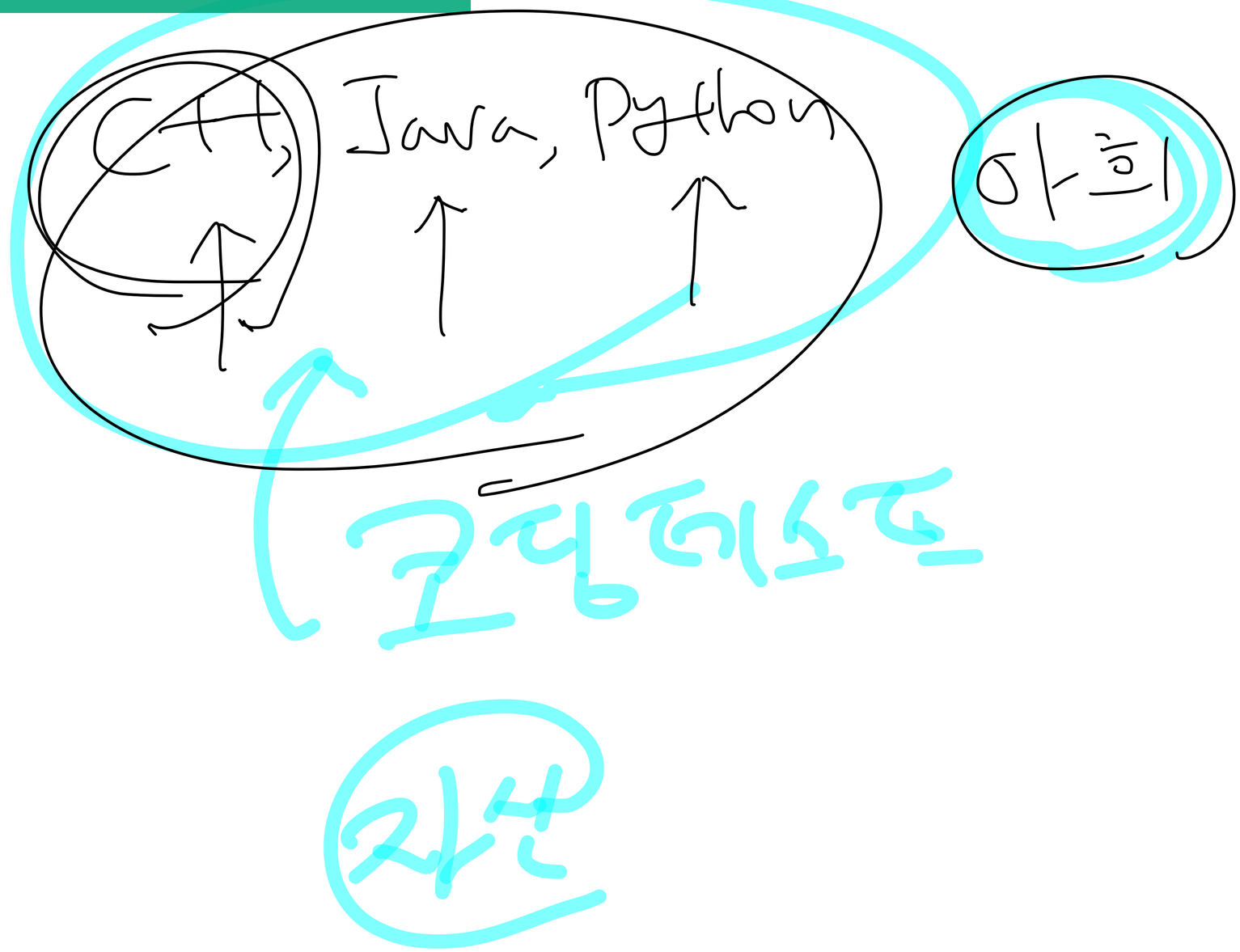
$$dy = [0, 0, 1, -1, 1, 1, -1, -1]$$

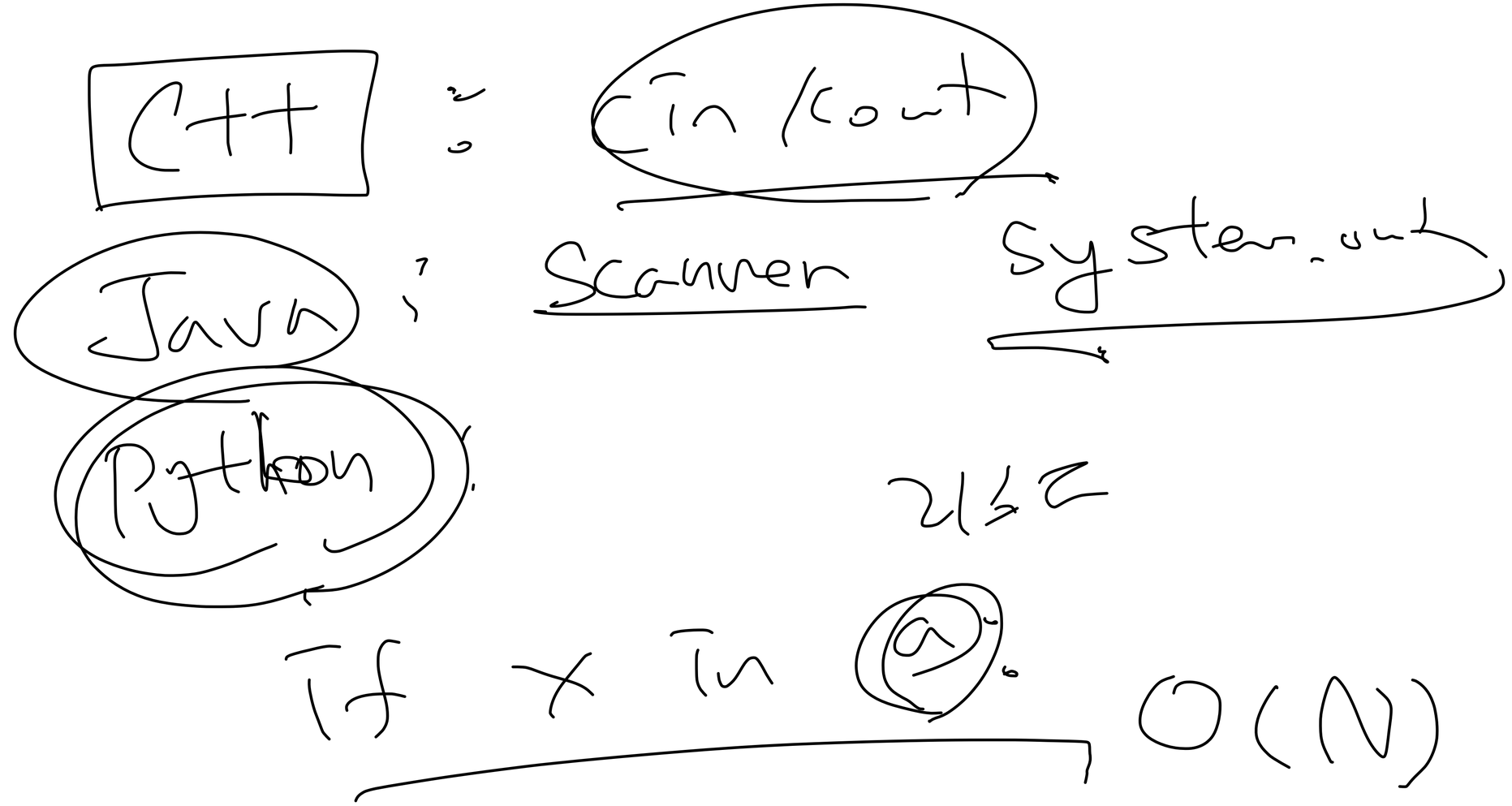
나이트 이동



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

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


C++,
Java, Python
아크릴


C++
Cin/cout
Java
Scanner
System.out
Python
21초
If x In @
O(N)

Python

a = [1, 2, 3]

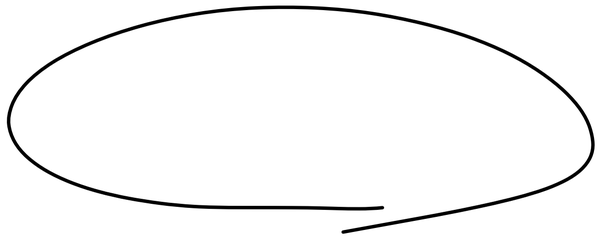
a[1:]

[2, 3]

a[1:5]

BFS Queue

```
q = [ ]
q.append(start)
while q:
    now = q[0]
    q = q[1:]
    for y in a[x]:
```



O(N)

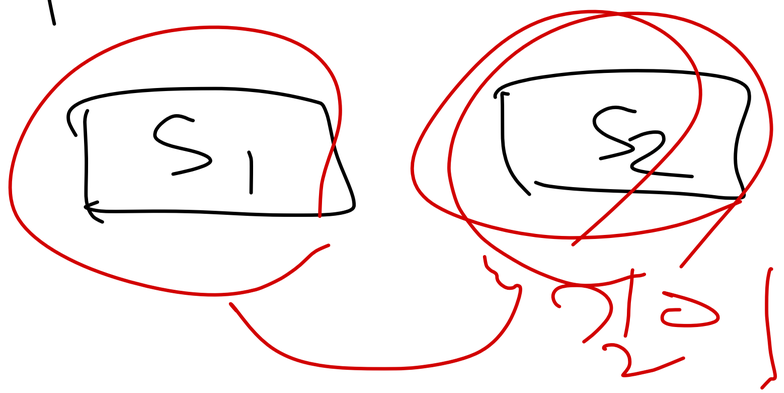
q.pop(0)

from collections import deque

Java

```
String S = "";
for (i=0; i<100000; ++i) {
    S += "abc";
}
```

C++



Java

Java

```java
import java.util.*;

class Pair {
    int first, second;
    Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }
}
public class Main {
    static final int[] dx = {0,0,1,-1};
    static final int[] dy = {1,-1,0,0};
    static void go(List<List<Integer>> a, int d, int k) {
        if (d == 0) {
            Collections.rotate(Arrays.asList(a[x]), k);
        } else {
            Collections.rotate(Arrays.asList(a[x]), (a[x].length-k));
        }
    }
    static boolean bfs(Integer[][] a) {
        int n = a.length-1;
        int m = a[1].length;
        boolean[][] d = new boolean[n+1][m];
        boolean ok = false;
        for (int i=1; i<=n; i++) {
            for (int j=0; j<m; j++) {
                if (d[i][j] == true) continue;
                if (a[i][j].intValue() == 0) continue;
                ArrayList<Pair> group = new ArrayList<>();
                int cnt = 0;
                Queue<Pair> q = new LinkedList<>();
                q.add(new Pair(i,j));
                group.add(new Pair(i,j));
                d[i][j] = true;
                while (!q.isEmpty()) {
                    Pair p = q.remove();
                    int x = p.first;
                    int y = p.second;
                    cnt += 1;
                    for (int k=0; k<4; k++) {
                        int nx = x+dx[k];
                        int ny = y+dy[k];
                        ny = (ny + m) % m;
                        if (1 <= nx && nx <= n && 0 <= ny && ny < m) {
                            if (d[nx][ny] == false && a[x][y].intValue() == a[nx][ny].intValue()) {
                                q.add(new Pair(nx,ny));
                                group.add(new Pair(nx,ny));
                                d[nx][ny] = true;
                            }
                        }
                    }
                }
                if (group.size() == 1) continue;
                ok = true;
                for (Pair p : group) {
                    int x = p.first;
                    int y = p.second;
                    a[x][y] = 0;
                }
            }
        }
        return ok;
    }
    static void adjust(Integer[][] a) {
        int n = a.length-1;
        int m = a[1].length;
        long sum = 0;
        long cnt = 0;
        for (int i=1; i<=n; i++) {
            for (int j=0; j<m; j++) {
                if (a[i][j].intValue() == 0) continue;
                sum += a[i][j];
                cnt += 1;
            }
        }
        if (cnt == 0LL) return;
        for (int i=1; i<=n; i++) {
            for (int j=0; j<m; j++) {
                if (a[i][j] == 0) continue;
                if (sum < (long long)a[i][j]*cnt) {
                    // sum/cnt < a[i][j] (-1)
                    a[i][j] -= 1;
                } else if (sum > (long long)a[i][j]*cnt) {
                    // sum/cnt > a[i][j] (+1)
                    a[i][j] += 1;
                }
            }
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int m = sc.nextInt();
        int t = sc.nextInt();
        Integer[][] a = new Integer[n+1][m];
        for (int i=1; i<=n; i++) {
            for (int j=0; j<m; j++) {
                a[i][j] = sc.nextInt();
            }
        }
        while (t-- > 0) {
            int x = sc.nextInt();
            int d = sc.nextInt();
            int k = sc.nextInt();
            for (int y=x; y<=n; y+=x) {
                go(a, y, d, k);
            }
            boolean ok = bfs(a);
            if (ok == false) {
                adjust(a);
            }
        }
        int ans = 0;
        for (int i=1; i<=n; i++) {
            for (int j=0; j<m; j++) {
                ans += a[i][j].intValue();
            }
        }
        System.out.println(ans);
    }
}
```

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

C++14

```cpp
#include <iostream>
#include <map>
#include <queue>
#include <vector>
#include <algorithm>
using namespace std;
int dx[] = {0,0,1,-1};
int dy[] = {1,-1,0,0};
void go(vector<vector<int>> &a, int x, int d, int k) {
    if (d == 0) {
        rotate(a[x].rbegin(), a[x].rbegin()+k, a[x].rend());
    } else { // d == 1
        rotate(a[x].begin(), a[x].begin()+k, a[x].end());
    }
}
bool bfs(vector<vector<int>> &a) {
    int n = (int)a.size() - 1;
    int m = a[1].size();
    vector<vector<bool>> d(n+1, vector<bool>(m, 0));
    bool ok = false;
    for (int i=1; i<=n; i++) {
        for (int j=0; j<m; j++) {
            if (d[i][j] == true) continue;
            if (a[i][j] == 0) continue;
            vector<pair<int,int>> group;
            int cnt = 0;
            queue<pair<int,int>> q;
            q.push(make_pair(i,j));
            group.push_back(make_pair(i,j));
            d[i][j] = true;
            while (!q.empty()) {
                int x, y;
                tie(x, y) = q.front(); q.pop();
                cnt += 1;
                for (int k=0; k<4; k++) {
                    int nx = x+dx[k];
                    int ny = y+dy[k];
                    ny = (ny + m) % m;
                    if (1 <= nx && nx <= n && 0 <= ny && ny < m) {
                        if (d[nx][ny] == false && a[x][y] == a[nx][ny]) {
                            q.push(make_pair(nx,ny));
                            group.push_back(make_pair(nx,ny));
                            d[nx][ny] = true;
                        }
                    }
                }
            }
            if (group.size() == 1) continue;
            ok = true;
            for (auto &p : group) {
                int x, y;
                tie(x,y) = p;
                a[x][y] = 0;
            }
        }
    }
    return ok;
}
void adjust(vector<vector<int>> &a) {
    int n = (int)a.size() - 1;
    int m = a[1].size();
    long long sum = 0;
    long long cnt = 0;
    for (int i=1; i<=n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 0) continue;
            sum += a[i][j];
            cnt += 1;
        }
    }
    if (cnt == 0LL) return;
    for (int i=1; i<=n; i++) {
        for (int j=0; j<m; j++) {
            if (a[i][j] == 0) continue;
            if (sum < (long long)a[i][j]*cnt) {
                // sum/cnt < a[i][j] (-1)
                a[i][j] -= 1;
            } else if (sum > (long long)a[i][j]*cnt) {
                // sum/cnt > a[i][j] (+1)
                a[i][j] += 1;
            }
        }
    }
}
int main() {
    int n, m, t;
    cin >> n >> m >> t;
    vector<vector<int>> a(n+1, vector<int>(m));
    for (int i=1; i<=n; i++) {
        for (int j=0; j<m; j++) {
            cin >> a[i][j];
        }
    }
    while (t--) {
        int x, d, k;
        cin >> x >> d >> k;
        for (int y=x; y<=n; y+=x) {
            go(a, y, d, k);
        }
        bool ok = bfs(a);
        if (ok == false) {
            adjust(a);
        }
    }
    int ans = 0;
    for (int i=1; i<=n; i++) {
        for (int j=0; j<m; j++) {
            ans += a[i][j];
        }
    }
    cout << ans << '\n';
    return 0;
}
```

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

17425 약수의 합

N ≤ 10^12

f(A) = A의 B번 약수의 합

f(24) = 1+2+3+4+6+8+12+24

O(N)

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

N → g(N) → $O(N^2)$

24 f(1)+f(2)+--+f(24)

```
int f(int N)
{
    int ans=0;
    for(i=1; i<=N; i++)
    {
        if( N%i==0 )
            ans+=i;
    }
    return ans;
}
```

O(N)

① 77 = 1, 7, 11, 77

② 2 4 6 8 ---

f[2] += 2

f[4] += 2

③ f[6] += 2

3 6 9 12 ---

C++14

```cpp
#include <iostream>
#include <tuple>
#include <algorithm>
#include <vector>
using namespace std;
int dx[] = {0,0,-1,1};
int dy[] = {1,-1,0,0};
int opposite(int dir) {
    if (dir == 0) return 1;
    if (dir == 1) return 0;
    if (dir == 2) return 3;
    return 2;
}
void go(vector<vector<vector<pair<int,int>>>> &a, vector<pair<int,int>> &where, int x, int y, int nx, int ny) {
    for (auto &p : a[x][y]) {
        a[nx][ny].push_back(p);
        where[p.first] = make_pair(nx, ny);
    }
    a[x][y].clear();
}
void print(vector<vector<vector<pair<int,int>>>> &a) {
    int n = a.size();
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (a[i][j].size() == 0) {
                cout << "- ";
            } else {
                for (auto &p : a[i][j]) {
                    cout << p.first << ',' << p.second << ' ';
                }
            }
        }
        cout << '\n';
    }
}
int main() {
    int n, m;
    cin >> n >> m;
    vector<vector<int>> board(n, vector<int>(n));
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> board[i][j];
        }
    }
    vector<vector<vector<pair<int,int>>>> a(n, vector<vector<pair<int,int>>>(n));
    vector<pair<int,int>> where(m);
    for (int i=0; i<m; i++) {
        int x, y, dir;
        cin >> x >> y >> dir;
        a[x-1][y-1].push_back(make_pair(i, dir-1));
        where[i] = make_pair(x-1, y-1);
    }
    for (int turn=1; turn<=1000; turn++) {
        for (int k=0; k<m; k++) {
            int x, y;
            tie(x,y) = where[k];
            if (a[x][y][0].first == k) { // bottom
                int dir = a[x][y][0].second;
                int nx = x+dx[dir];
                int ny = y+dy[dir];
                if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
                    if (board[nx][ny] == 2) {
                        a[x][y][0].second = opposite(dir);
                    }
                } else {
                    a[x][y][0].second = opposite(dir);
                }
                dir = a[x][y][0].second;
                nx = x+dx[dir];
                ny = y+dy[dir];
                if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
                    if (board[nx][ny] == 0) {
                        go(a, where, x, y, nx, ny);
                    } else if (board[nx][ny] == 1) {
                        reverse(a[x][y].begin(), a[x][y].end());
                        go(a, where, x, y, nx, ny);
                    }
                    if (a[nx][ny].size() >= 4) {
                        cout << turn << '\n';
                        return 0;
                    }
                } else { // out
                }
            }
            //cout << "#" << turn << ' ' << k << '\n';
            //print(a);
        }
    }
    cout << -1 << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	1988 KB	0 ms	2835 B

C++14

```cpp
#include <iostream>
#include <cassert>
#include <tuple>
#include <algorithm>
#include <vector>
using namespace std;
int dx[] = {0,0,-1,1};
int dy[] = {1,-1,0,0};
int opposite(int dir) {
    if (dir == 0) return 1;
    if (dir == 1) return 0;
    if (dir == 2) return 3;
    return 2;
}
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) {
    for (int i=start; i<a[x][y].size(); i++) {
        auto &p = a[x][y][i];
        a[nx][ny].push_back(p);
        where[p.first] = make_tuple(nx, ny, (int)a[nx][ny].size()-1);
    }
    a[x][y].resize(start);
}
void print(vector<vector<vector<pair<int,int>>>> &a) {
    int n = a.size();
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if (a[i][j].size() == 0) {
                cout << "- ";
            } else {
                for (auto &p : a[i][j]) {
                    cout << p.first << ',' << p.second << ' ';
                }
            }
        }
        cout << '\n';
    }
}
int main() {
    int n, m;
    cin >> n >> m;
    vector<vector<int>> board(n, vector<int>(n));
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            cin >> board[i][j];
        }
    }
    vector<vector<vector<pair<int,int>>>> a(n, vector<vector<pair<int,int>>>(n));
    vector<tuple<int,int,int>> where(m);
    for (int i=0; i<m; i++) {
        int x, y, dir;
        cin >> x >> y >> dir;
        a[x-1][y-1].push_back(make_pair(i, dir-1));
        where[i] = make_tuple(x-1, y-1, (int)a[x-1][y-1].size()-1);
    }
    for (int turn=1; turn<=1000; turn++) {
        for (int k=0; k<m; k++) {
            int x, y, index;
            tie(x,y,index) = where[k];
            assert(a[x][y][index].first == k);
            int dir = a[x][y][index].second;
            int nx = x+dx[dir];
            int ny = y+dy[dir];
            if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
                if (board[nx][ny] == 2) {
                    a[x][y][index].second = opposite(dir);
                }
            } else {
                a[x][y][index].second = opposite(dir);
            }
            dir = a[x][y][index].second;
            nx = x+dx[dir];
            ny = y+dy[dir];
            if (0 <= nx && nx < n && 0 <= ny && ny < n) { // in
                if (board[nx][ny] == 0) {
                    go(a, where, x, y, nx, ny, index);
                } else if (board[nx][ny] == 1) {
                    reverse(a[x][y].begin()+index, a[x][y].end());
                    go(a, where, x, y, nx, ny, index);
                }
                if (a[nx][ny].size() >= 4) {
                    cout << turn << '\n';
                    return 0;
                }
            } else { // out
                }
                //cout << "#" << turn << ' ' << k << '\n';
                //print(a);
            }
        }
    }
    cout << -1 << '\n';
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	1988 KB	0 ms	2908 B

Java

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.StringTokenizer;

public class Main {
    static int N, K;
    static final int WHITE = 0;
    static final int RED = 1;
    static final int BLUE = 2;
    static int[][] map;
    static ArrayList<Integer>[][] arr;
    static ArrayList<Horse> horse_list;
    static int[] di = { 0, 0, -1, 1 };
    static int[] dj = { 0, 1, 0, -1 };

    public static void main(String[] args) throws IOException {
        init();
        solution();
    }

    private static void init() throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        N = Integer.parseInt(st.nextToken());
        K = Integer.parseInt(st.nextToken());
        map = new int[N + 2][N + 2];
        arr = new ArrayList[N + 2][N + 2];
        horse_list = new ArrayList<>();
        for (int j = 0; j < N + 2; j++) {
            map[0][j] = BLUE;
            map[j][0] = BLUE;
            map[N + 1][j] = BLUE;
            map[j][N + 1] = BLUE;
        }
        // 맵에 대한 정보와 2차원 어레이리스트 배열 초기화
        for (int i = 1; i <= N; i++) {
            st = new StringTokenizer(br.readLine());
            for (int j = 1; j <= N; j++) {
                arr[i][j] = new ArrayList<>();
                map[i][j] = Integer.parseInt(st.nextToken());
            }
        }
        // 말 리스트 초기화
        horse_list.add(new Horse(0, -1, -1, -1)); //의미없는말 추가(인덱스맞추기)
        for (int i = 1; i <= K; i++) {
            st = new StringTokenizer(br.readLine());
            int ni = Integer.parseInt(st.nextToken());
            int nj = Integer.parseInt(st.nextToken());
            int dir = Integer.parseInt(st.nextToken());
            Horse h = new Horse(i, ni, nj, dir);
            horse_list.add(h);
            arr[ni][nj].add(i); // [ni][nj] 번째는 i번째 말이 들어간다.
        }

    }

    private static void solution() {
        int turn = 0;
        while (true) {
            turn++;
            boolean isend = false;
            for (int h = 1; h <= K; h++) {
                Horse now = horse_list.get(h); // 현재 말을 받아 움직일려는 쪽이어딘지봐야함.
                int nexti = now.i + di[now.dir];
                int nextj = now.j + dj[now.dir];
                int color = map[nexti][nextj]; //움직일려는 칸의 색깔.
                switch (color) {
                case WHITE: //파란색이면,
                    white(now);
                    break;
                case RED:
                    red(now);
                    break;
                case BLUE:
                    nexti = now.i - di[now.dir];
                    nextj = now.j - dj[now.dir];
                    //파란색이면 방향만 반대방향으로 바꾸고,
                    if (map[nexti][nextj] == WHITE) {
                        now.dir = dirReverse(now.dir);
                        white(now);
                    } else if (map[nexti][nextj] == RED) {
                        now.dir = dirReverse(now.dir);
                        red(now);
                    }else {//또 블루면 이동 안하고 방향만 바꿔주기.
                        now.dir = dirReverse(now.dir);
                    }
                    break;
                }
                if (size()>=4) {
                    System.out.println(turn);
                    isend=true;
                    break;
                }
            }
            if(isend) {
                break;
            }

            if (turn > 1000) {
                System.out.println(-1);
                break;
            }
        }
    }

    private static int size() {
        int cnt = 0;
        for (int i = 1; i < horse_list.size(); i++) {
            cnt = Math.max(arr[horse_list.get(i).i][horse_list.get(i).j].size(), cnt);
        }
        return cnt;
    }

    private static int dirReverse(int dir) {
        if (dir == 1) {
            return 2;
        } else if (dir == 2) {
            return 1;
        } else if (dir == 3) {
            return 4;
        } else {
            return 3;
        }
    }

    private static void red(Horse horse) {
        ArrayList<Integer> temp = new ArrayList<>();
        //현재 위치의 배열에서  horse 인덱스번째를 찾는다.
        int size = arr[horse.i][horse.j].size();
        for (int i = size - 1; i >= 0; i--) {
            int removeIDX = arr[horse.i][horse.j].remove(i);
            if (removeIDX == horse.idx) {
                temp.add(removeIDX);
                break;
            } else {
                temp.add(removeIDX);
            }
        }
        int nexti = horse.i + di[horse.dir];
        int nextj = horse.j + dj[horse.dir];
        for (int i = 0; i < temp.size(); i++) {
            int idx = temp.get(i);
            arr[nexti][nextj].add(idx);
            horse_list.get(idx).i = nexti;
            horse_list.get(idx).j = nextj;
        }

    }

    private static void white(Horse horse) {
        //내가 움직일려는 nexti,nextj
        int nexti = horse.i + di[horse.dir];
        int nextj = horse.j + dj[horse.dir];
        ArrayList<Integer> temp = new ArrayList<>();
        int size = arr[horse.i][horse.j].size();
        for (int i = size - 1; i >= 0; i--) {
            int removeIDX = arr[horse.i][horse.j].remove(i);
            if (removeIDX == horse.idx) {
                temp.add(removeIDX);
                break;
            } else {
                temp.add(removeIDX);
            }
        }
        for (int i = 0; i < temp.size(); i++) {
            int idx = temp.get(temp.size() - i - 1);
            arr[nexti][nextj].add(idx);
            horse_list.get(idx).i = nexti;
            horse_list.get(idx).j = nextj;
        }
    }


    static class Horse {
        int idx, i, j, dir;

        Horse(int idx, int i, int j, int dir) {
            this.idx = idx;
            this.i = i;
            this.j = j;
            this.dir = dir;
        }
    }
}
```

결과	메모리	시간	코드 길이
채점 중			4840 B

Dynamic programming

14501번. 퇴사

1일 ~ N일

상담

N+1 일 퇴사

1일

T[i]일 기간

P[i] 원

1 ≤ N ≤ 15

최댓값

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

각각의 상담은 상담을 완료하는데 걸리는 기간 T_i와 상담을 했을 때 받을 수 있는 금액 P_i로 이루어져 있다.

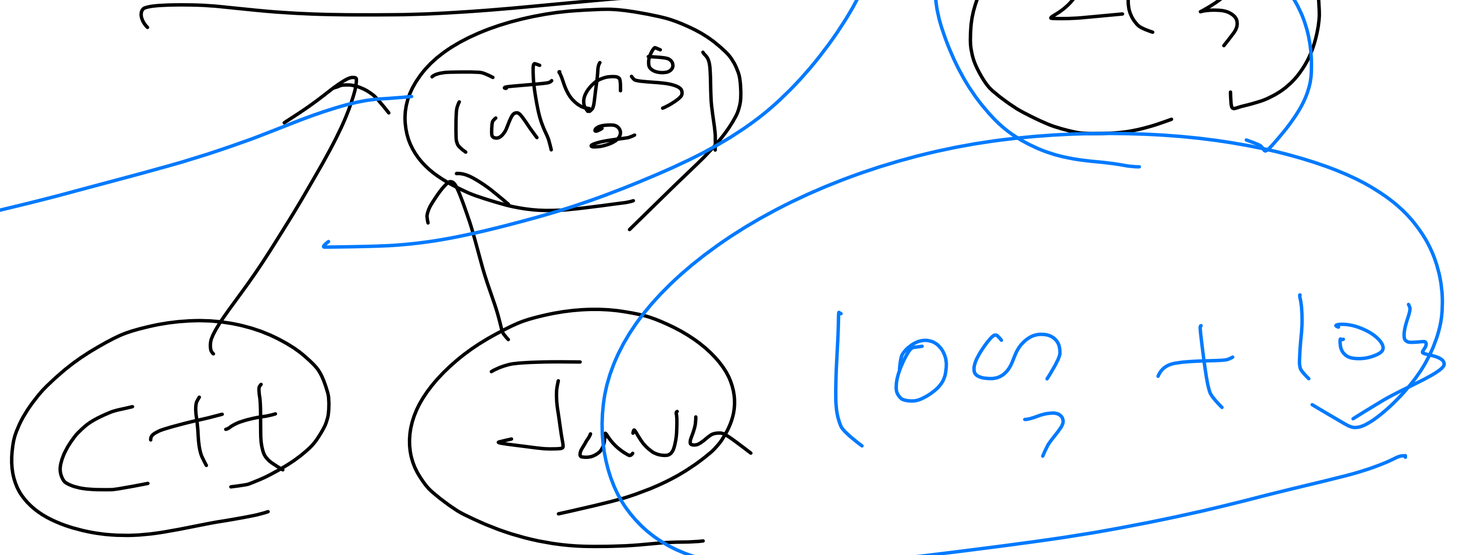
N = 7인 경우에 다음과 같은 상담 일정표를 보자.

	1일	2일	3일	4일	5일	6일	7일
T_i	3	5	1	1	2	4	2
P_i	10	20	10	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 1일에 상담을 하게 되면, 2일, 3일에 있는 상담은 할 수 없게 된다. 2일에 있는 상담을 하게 되면, 3, 4, 5, 6일

$2^{31} - 1 = 2147483647$



$2^{63} - 1 = ?$

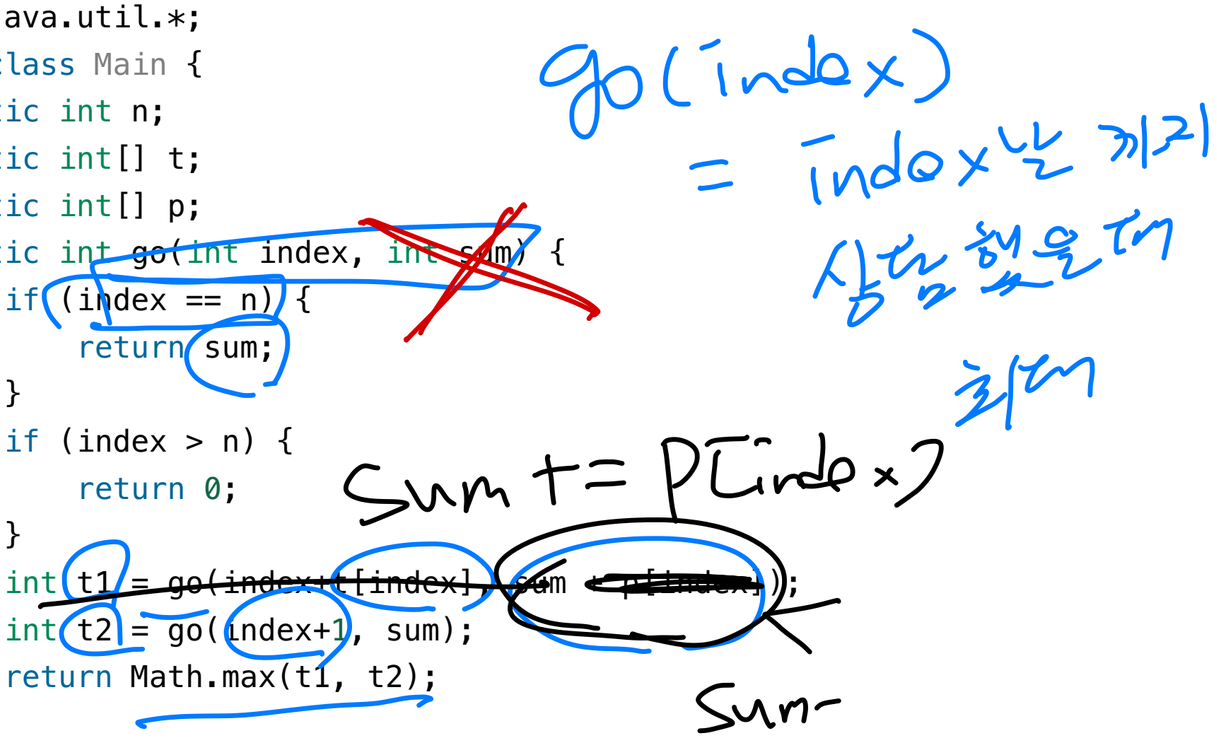
Java

```java
import java.util.*;
public class Main {
    static int n;
    static int[] t;
    static int[] p;
    static int ans=0;
    static void go(int index, int sum) {
        if (index == n) {
            if (ans < sum) {
                ans = sum;
            }
            return;
        }
        if (index > n) {
            return;
        }
        go(index+t[index], sum + p[index]);
        go(index+1, sum);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        t = new int[n];
        p = new int[n];
        for (int i=0; i<n; i++) {
            t[i] = sc.nextInt();
            p[i] = sc.nextInt();
        }
        go(0,0);
        System.out.println(ans);
    }
}
```

결과	메모리	시간	코드 길이
채점 중			749 B

Java

```java
import java.util.*;
public class Main {
    static int n;
    static int[] t;
    static int[] p;
    static int go(int index, int sum) {
        if (index == n) {
            return sum;
        }
        if (index > n) {
            return 0;
        }
        int t1 = go(index+t[index], sum + p[index]);
        int t2 = go(index+1, sum);
        return Math.max(t1, t2);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        t = new int[n];
        p = new int[n];
        for (int i=0; i<n; i++) {
            t[i] = sc.nextInt();
            p[i] = sc.nextInt();
        }
        System.out.println(go(0,0));
    }
}
```



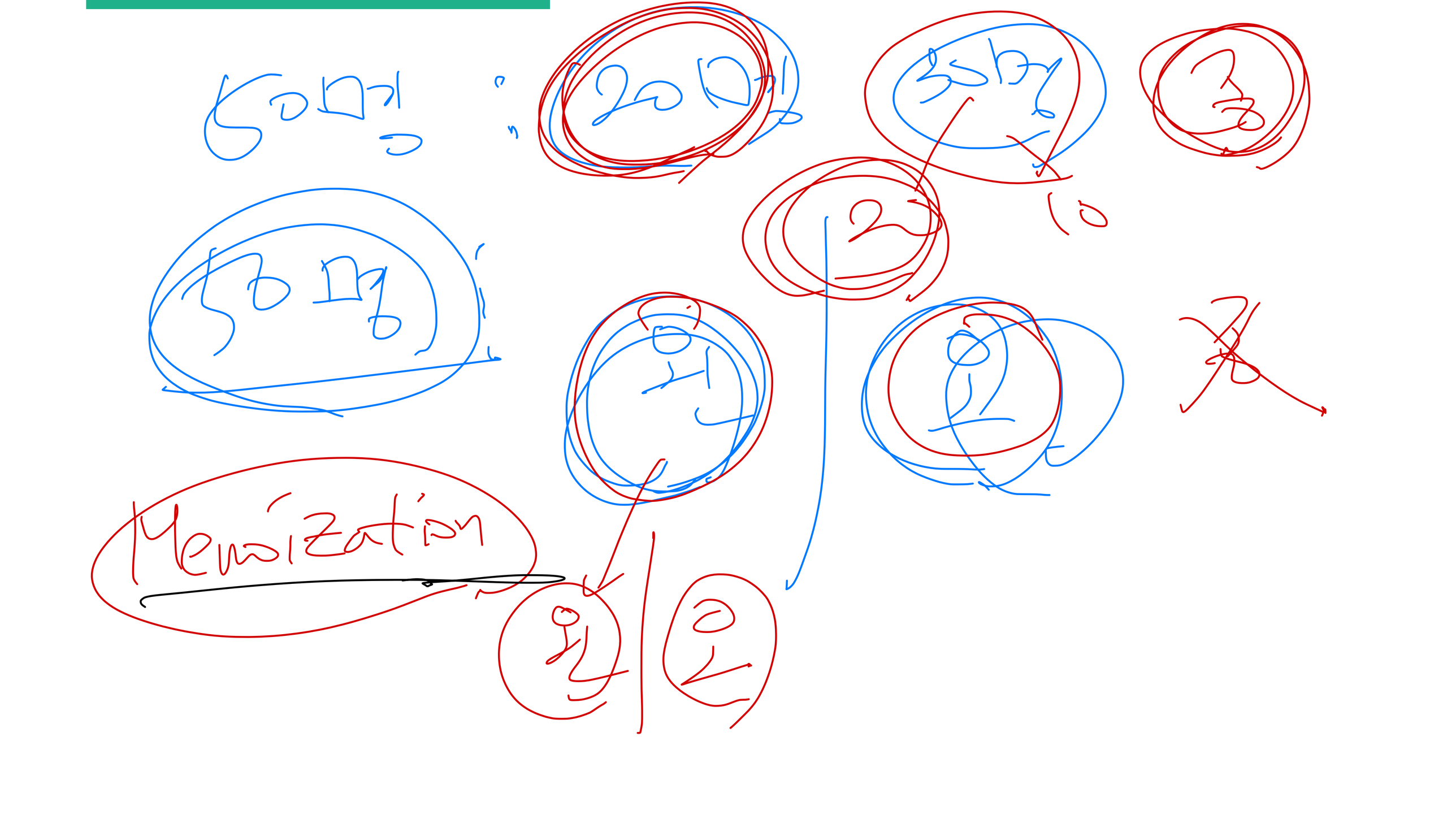
결과	메모리	시간	코드 길이
채점 중			700 B

Java

```java
import java.util.*;
public class Main {
    static int n;
    static int[] t;
    static int[] p;
    // 0~(n-1), 퇴사: n일
    // go(index) = index일부터 얻을수 있는 최대 수익
    static int go(int index) {
        if (index == n) { // 퇴사하는 날
            return 0;
        }
        if (index > n) { // 조건 위배 (퇴사날짜 넘김)
            return -1000000000;
        }
        // index index+t[index] ... n
        // -------------- -------------
        //.    p[index]   go(index+t[index])
        int t1 = p[index] + go(index+t[index]);
        int t2 = go(index+1);
        return Math.max(t1, t2);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        t = new int[n];
        p = new int[n];
        for (int i=0; i<n; i++) {
            t[i] = sc.nextInt();
            p[i] = sc.nextInt();
        }
        System.out.println(go(0));
    }
}
```

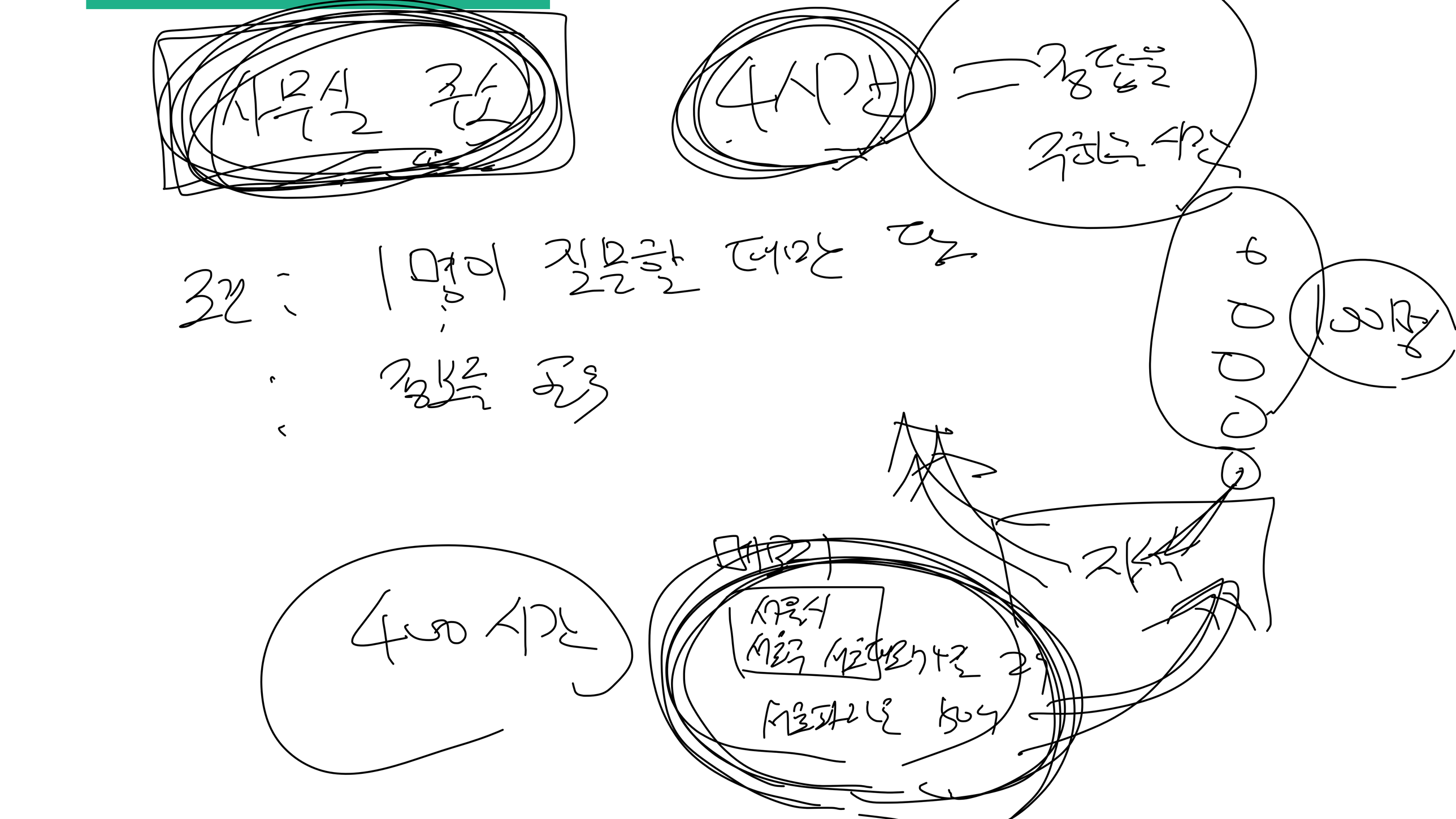
결과	메모리	시간	코드 길이
채점 중			961 B

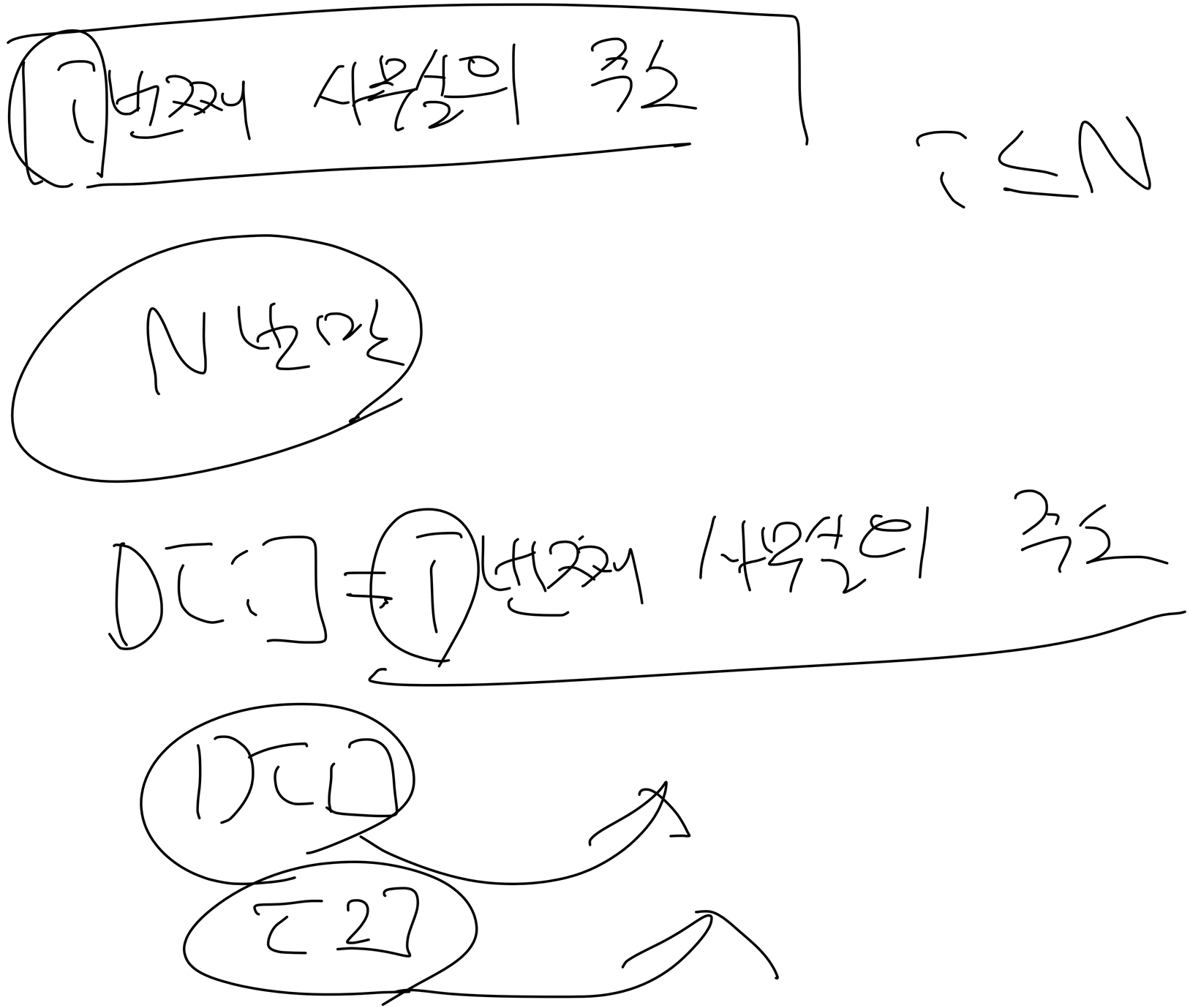

김밥

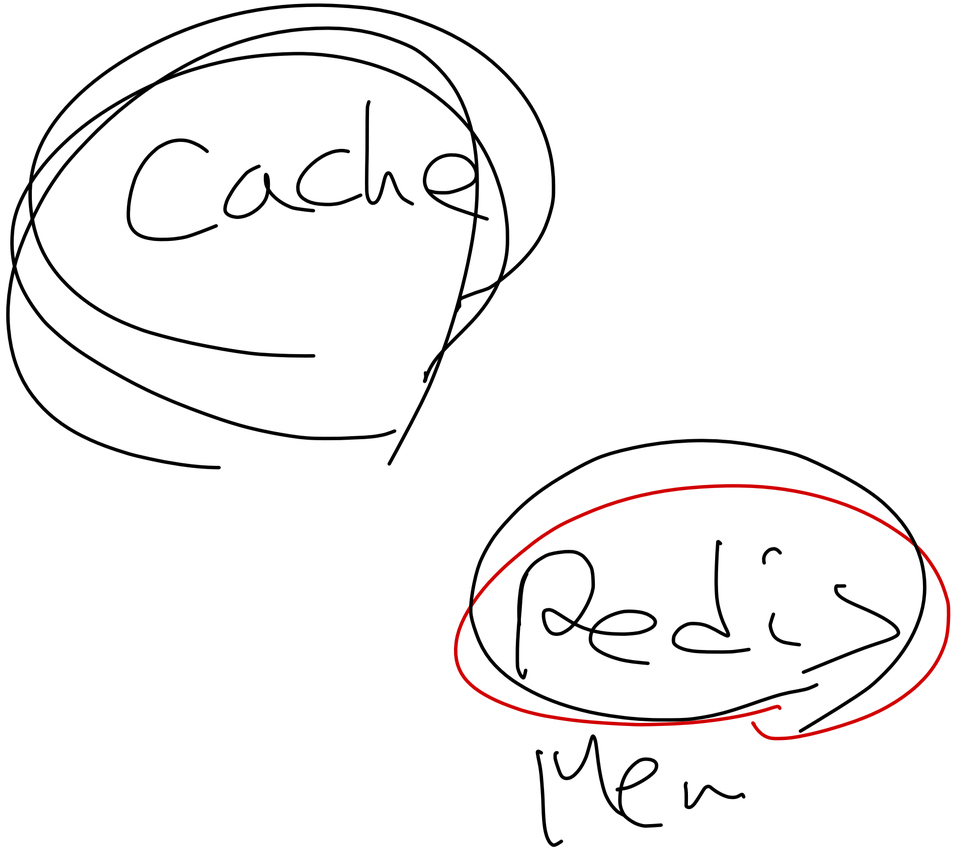

Memoization

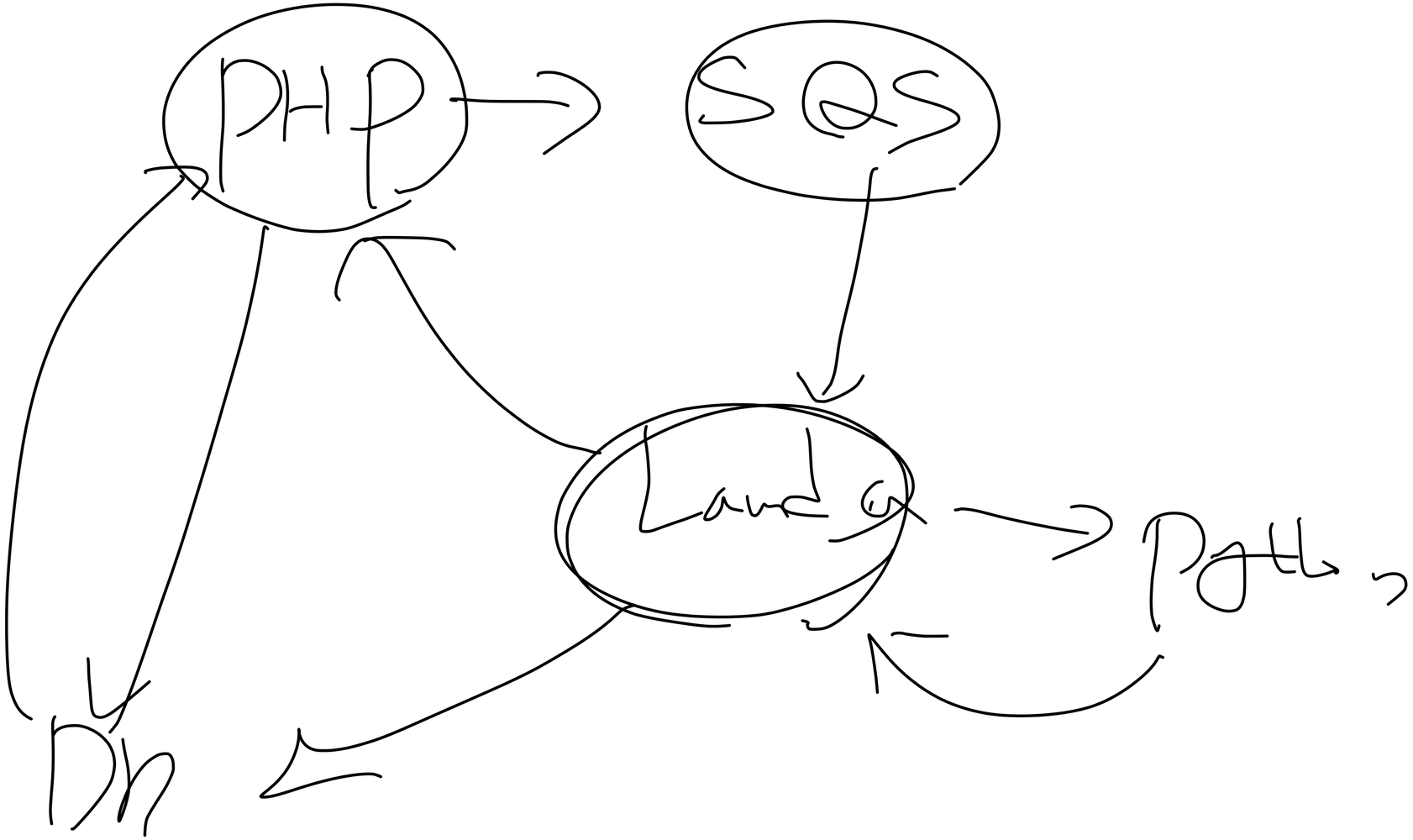
모델의 장점이

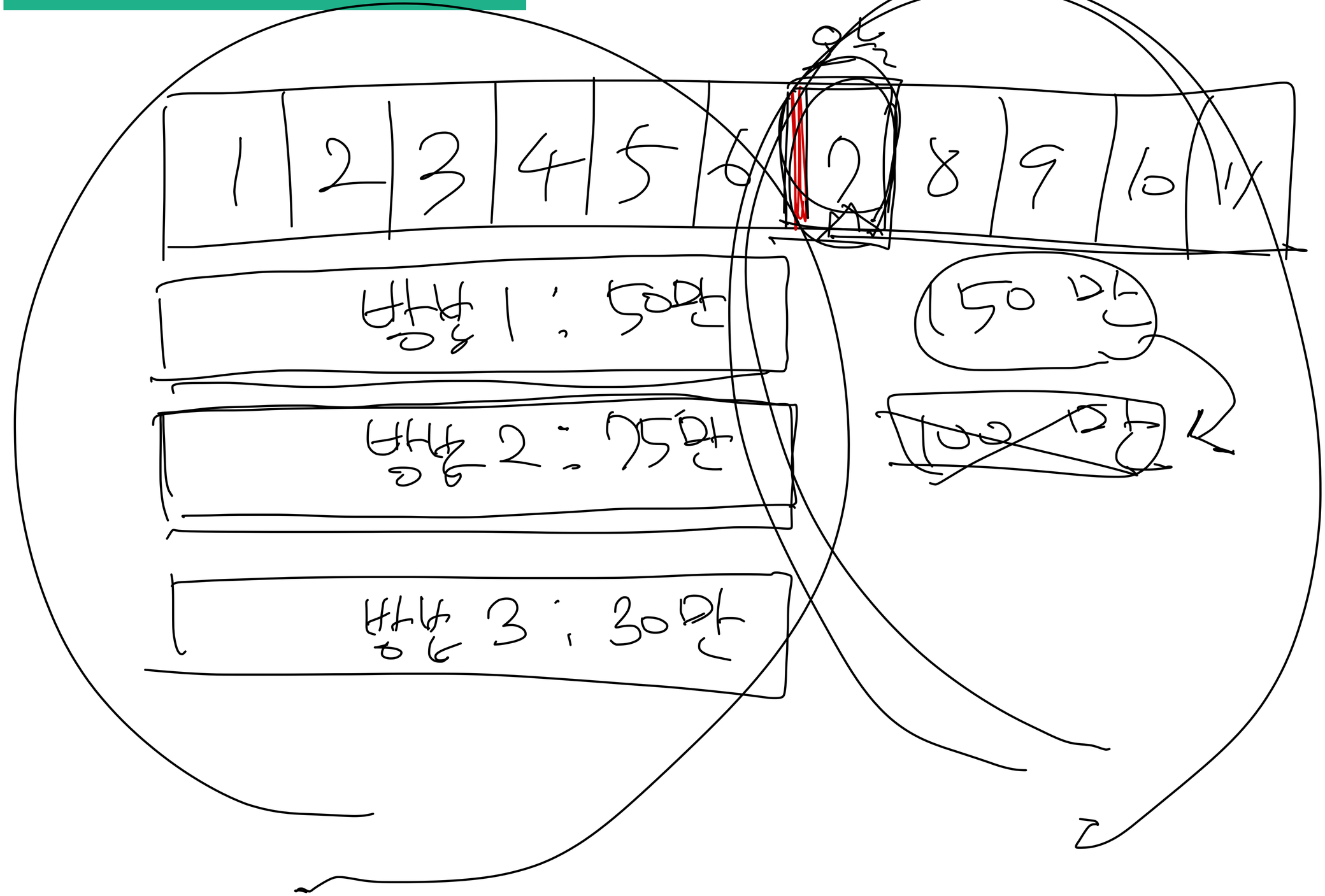
언제 구해도 같아야 함






Cache
Redis
Mem


PHP
SQS
Laravel
Python
DB


1
2
3
4
5
6
7
8
9
10
11
버퍼 1 : 50만
버퍼 2 : 75만
버퍼 3 : 30만
50 만
100 만

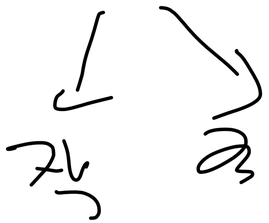
Java

```java
import java.util.*;
import java.io.*;
public class Main {
    static int n;
    static int[] t;
    static int[] p;
    static int[] d;
    // 0~(n-1), 퇴사: n일
    // go(index) = index일부터 얻을수 있는 최대 수익
    static int go(int index) {
        if (index == n) { // 퇴사하는 날
            return 0;
        }
        if (index > n) { // 조건 위배 (퇴사날짜 넘김)
            return -1000000000;
        }
        if (d[index] >= 0) return d[index];
        // index index+t[index] ... n
        // ------------ ------------
        //.    p[index]   go(index+t[index])
        int t1 = p[index] + go(index+t[index]);
        int t2 = go(index+1);
        d[index] = Math.max(t1, t2);
        return d[index];
    }
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
        n = Integer.parseInt(bf.readLine());
        t = new int[n];
        p = new int[n];
        d = new int[n];
        for (int i=0; i<n; i++) {
            String[] temp = bf.readLine().split(" ");
            t[i] = Integer.parseInt(temp[0]);
            p[i] = Integer.parseInt(temp[1]);
            d[i] = -1;
        }
        System.out.println(go(0));
    }
}
```

결과	메모리	시간	코드 길이
채점 중			1274 B

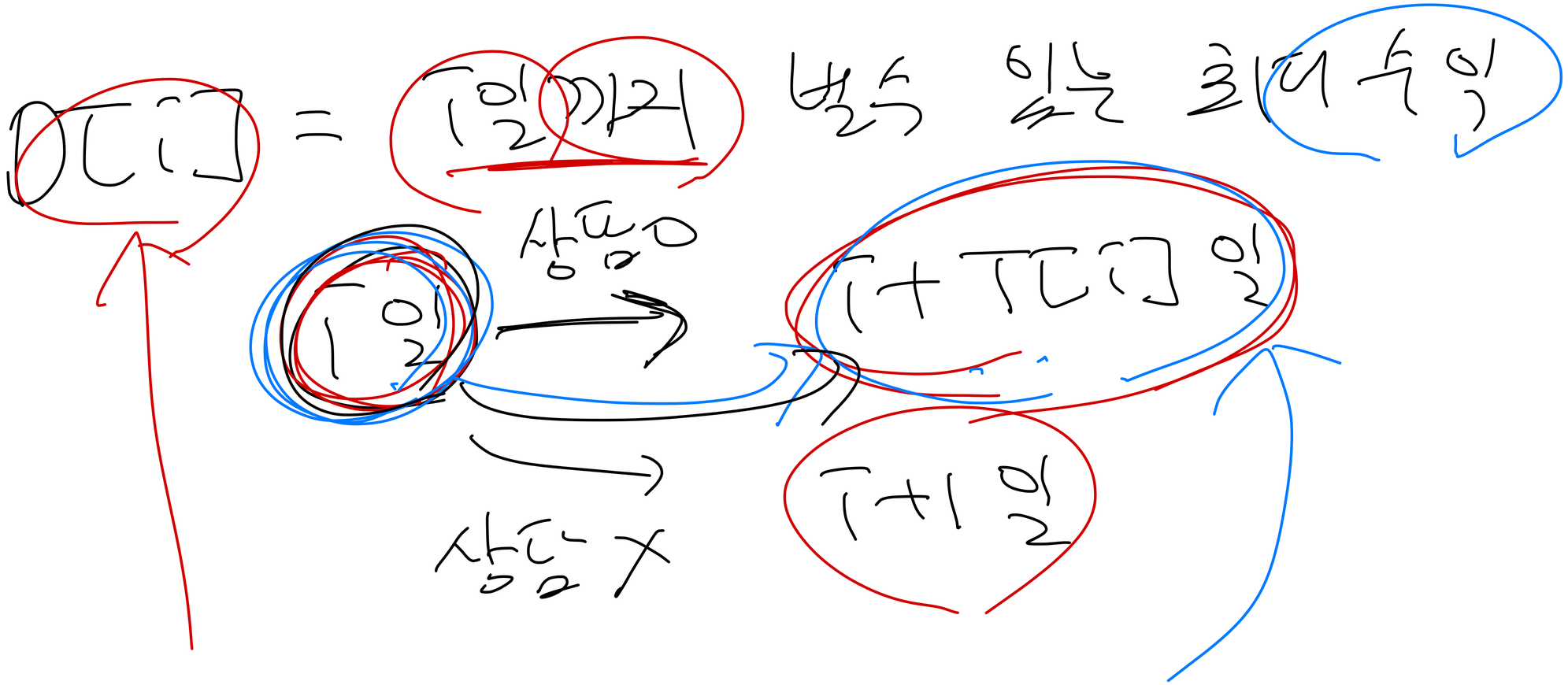
Dynamic Programming

분할



재귀 Top Down

반복 Bottom UP



$$D[i+T[i]] = D[i] + p[i]$$

$$D[i+1] = D[i]$$

Java

```java
import java.util.*;
import java.io.*;
public class Main {
    static int n;
    static int[] t;
    static int[] p;
    static int[] d;
    public static void main(String[] args) throws IOException {
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
        n = Integer.parseInt(bf.readLine());
        t = new int[n];
        p = new int[n];
        d = new int[n+1000];
        for (int i=0; i<n; i++) {
            String[] temp = bf.readLine().split(" ");
            t[i] = Integer.parseInt(temp[0]);
            p[i] = Integer.parseInt(temp[1]);
        }
        for(int i=0; i<n; i++) {
            d[i+t[i]] = Math.max(d[i+t[i]], d[i]+p[i]);
            d[i+1] = Math.max(d[i+1], d[i]);
        }
        System.out.println(d[n]);
    }
}
```

결과	메모리	시간	코드 길이
채점 중			779 B

11052번 : 카드 구매하기

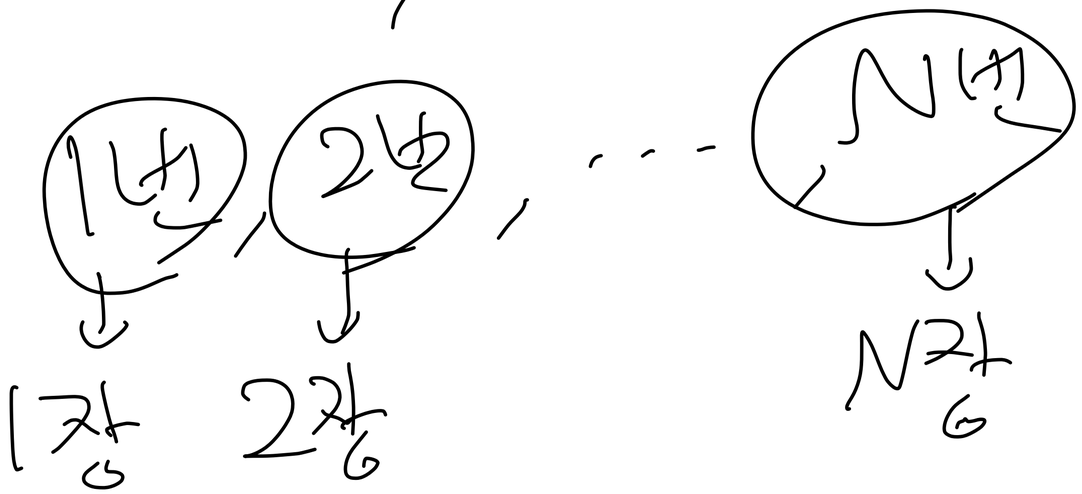
카드를 N개 구매 최대비용

N개지

카드팩 구매

→ 카드가 들어있는

개념 pdf



DP[i] = 카드 i장을 구매하는 최대 비용
카드팩
카드팩
DP[i-j]
나머지
카드 i-j장
i장
카드팩
P[j]
카드 몇장?
j장
j번 카드팩

DC[i] = 카드 i 장을 구매하는 최대 비용

$= \max(DC[i-j] + p[j])$ $(1 \le j \le i)$

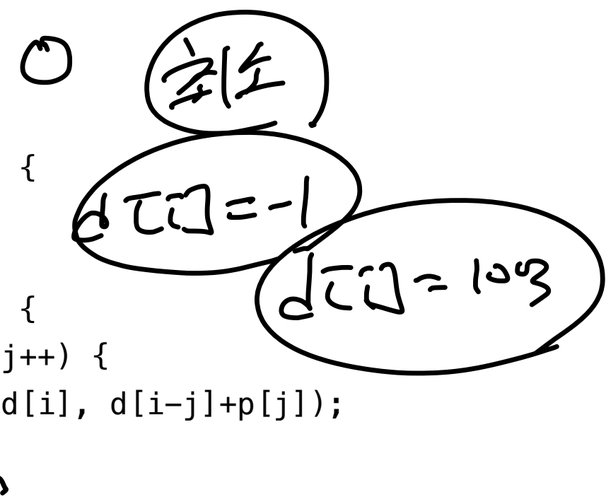
(문제의 수) × (문제 1개를 구하는 시간)

N × O(N)

$O(N^2)$

Java

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] d = new int[n+1];
        int[] p = new int[n+1];
        for (int i=1; i<=n; i++) {
            p[i] = sc.nextInt();
        }
        for (int i=1; i<=n; i++) {
            for (int j=1; j<=i; j++) {
                d[i] = Math.max(d[i], d[i-j]+p[j]);
            }
        }
        System.out.println(d[n]);
    }
}
```



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

Java

```java
import java.util.*;
public class Main {
    static int n;
    static int[] d, p;
    static int go(int i) {
        if (i == 0) return 0;
        if (d[i] > 0) return d[i];
        for (int j=1; j<=i; j++) {
            d[i] = Math.max(d[i], go(i-j)+p[j]);
        }
        return d[i];
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        d = new int[n+1];
        p = new int[n+1];
        for (int i=1; i<=n; i++) {
            p[i] = sc.nextInt();
        }
        System.out.println(go(n));
    }
}
```

결과	메모리	시간	코드 길이
채점 중			582 B

16987 계란으로 계란치기

계란 A로 계란 B

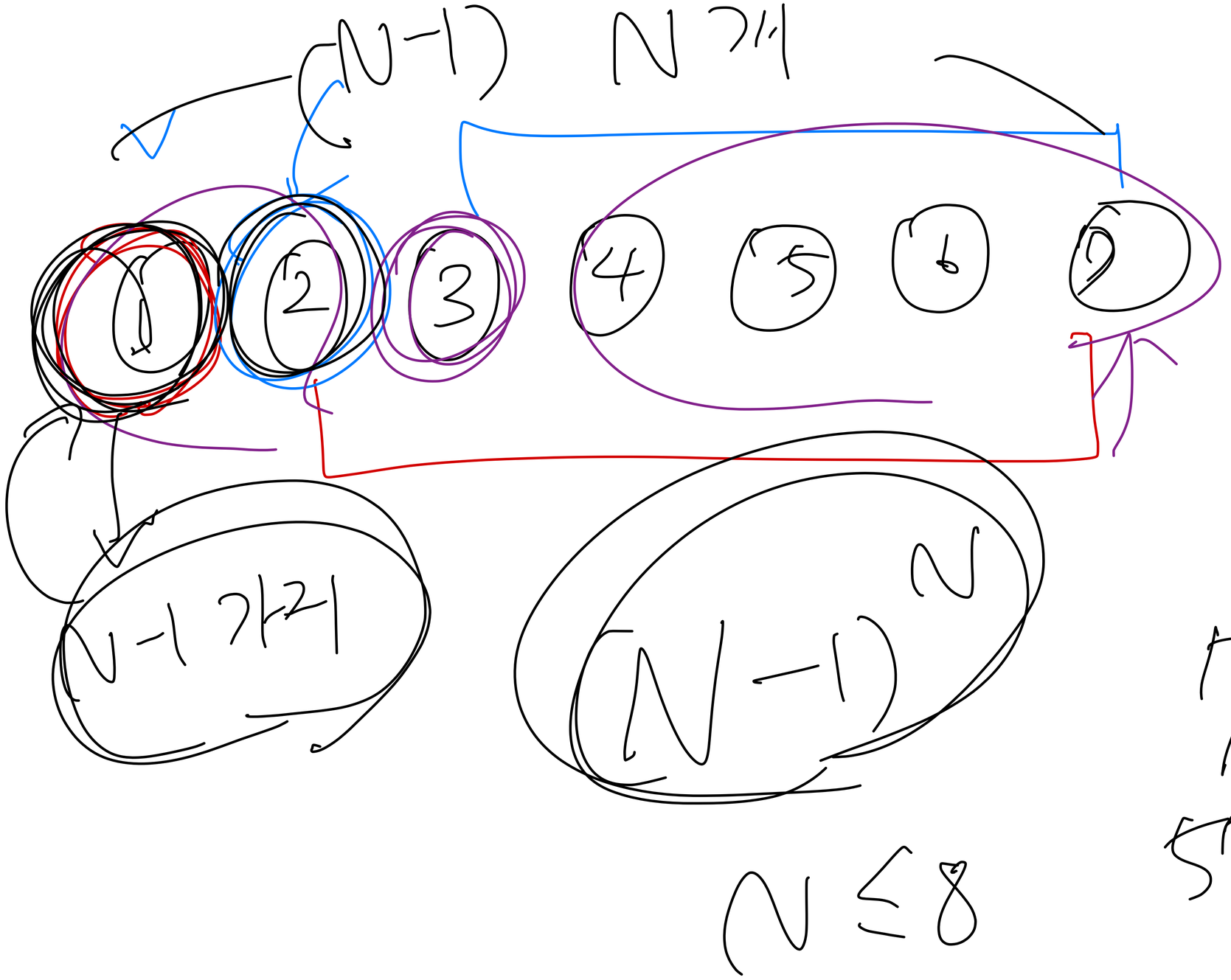
B의 내구도 -= A의 무게

A의 내구도 -= B의 무게

내구도

무게

내구도 ≤ 0 => 계란은 깨짐


(N-1) N 가지
1
2
3
4
5
6
7
N-1 가지
(N-1)^N
N ≤ 8
7^8 =
5764801

① 깨지지 않은 계란을 친다

② 볼에는 거품이 깨질

③ 깨지지 않은 계란이 있는 한

16987번 - 계란으로 계란치기

baekjoon

Java

```java
import java.util.*;
public class Main {
    static int n; // 0 1 ... n-1 (n)
    static int[] s; // 내구도
    static int[] w; // 무게
    static int ans = 0;
    static void go(int index) {
        if (index == n) {
            // 깨진 계란의 수 계산
            int cnt = 0;
            for (int i=0; i<n; i++) {
                if (s[i] <= 0) {
                    cnt += 1;
                }
            }
            if (ans < cnt) {
                ans = cnt;
            }
            return;
        }
        if (s[index] <= 0) {
            go(index+1);
            return;
        }
        boolean ok = false;
        for (int i=0; i<n; i++) {
            // index -> i
            if (index == i) continue;
            if (s[i] <= 0) continue;
            s[index] -= w[i];
            s[i] -= w[index];
            go(index+1);
            s[i] += w[index];
            s[index] += w[i];
            ok = true;
        }
        if (!ok) go(index+1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        s = new int[n];
        w = new int[n];
        for (int i=0; i<n; i++) {
            s[i] = sc.nextInt();
            w[i] = sc.nextInt();
        }
        go(0);
        System.out.println(ans);
    }
}
```

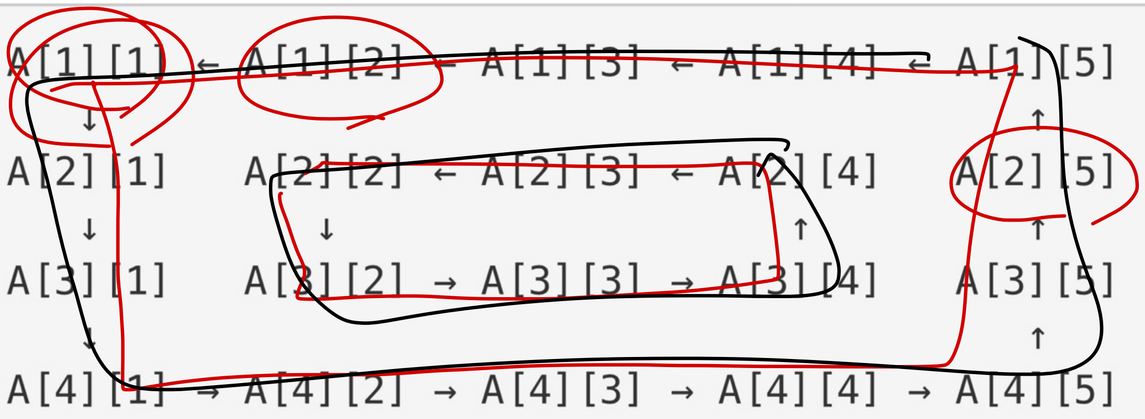
결과	메모리	시간	코드 길이
채점 중			1313 B

16926

그림

문제

크기가 N×M인 배열이 있을 때, 배열을 돌려보려고 한다. 배열은 다음과 같이 반시계 방향

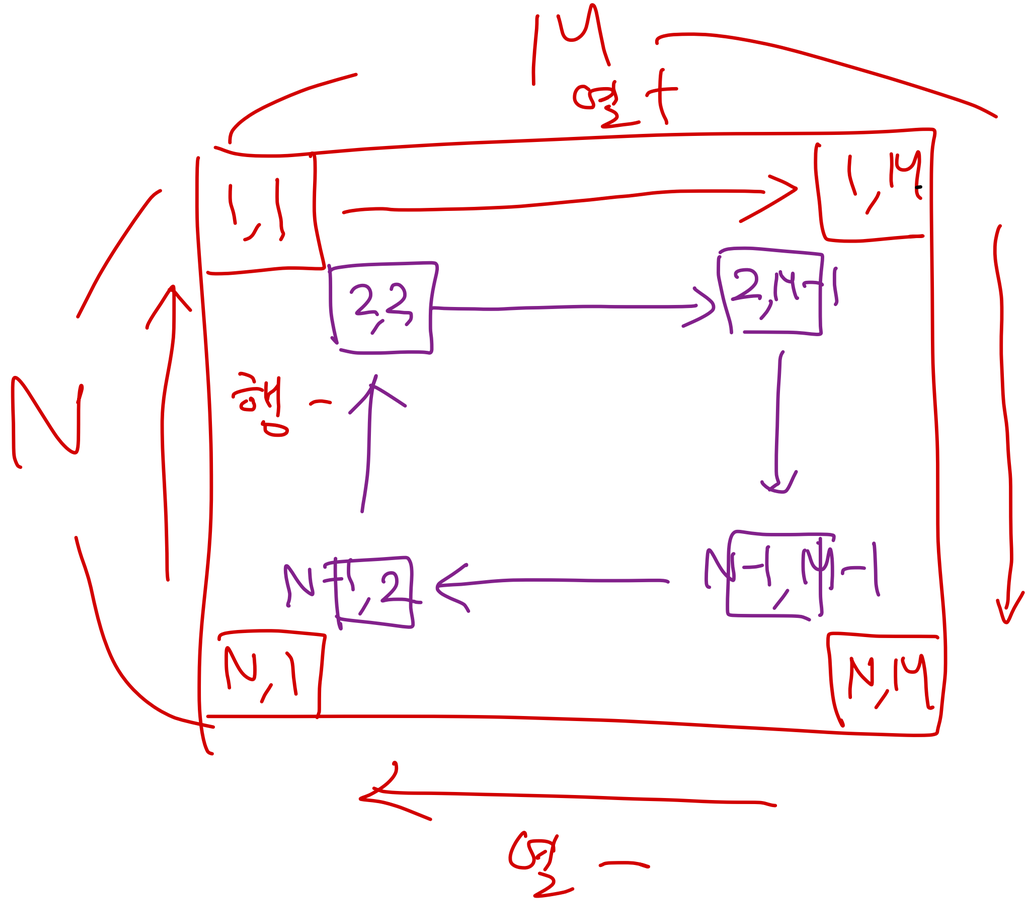


예를 들어, 아래와 같은 배열을 2번 회전시키면 다음과 같이 변하게 된다.

```
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>
```

배열과 정수 R이 주어졌을 때, 배열을 R번 회전시킨 결과를 구해보자.

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



T번째

$(i, i) \rightarrow (i, M-i+1)$

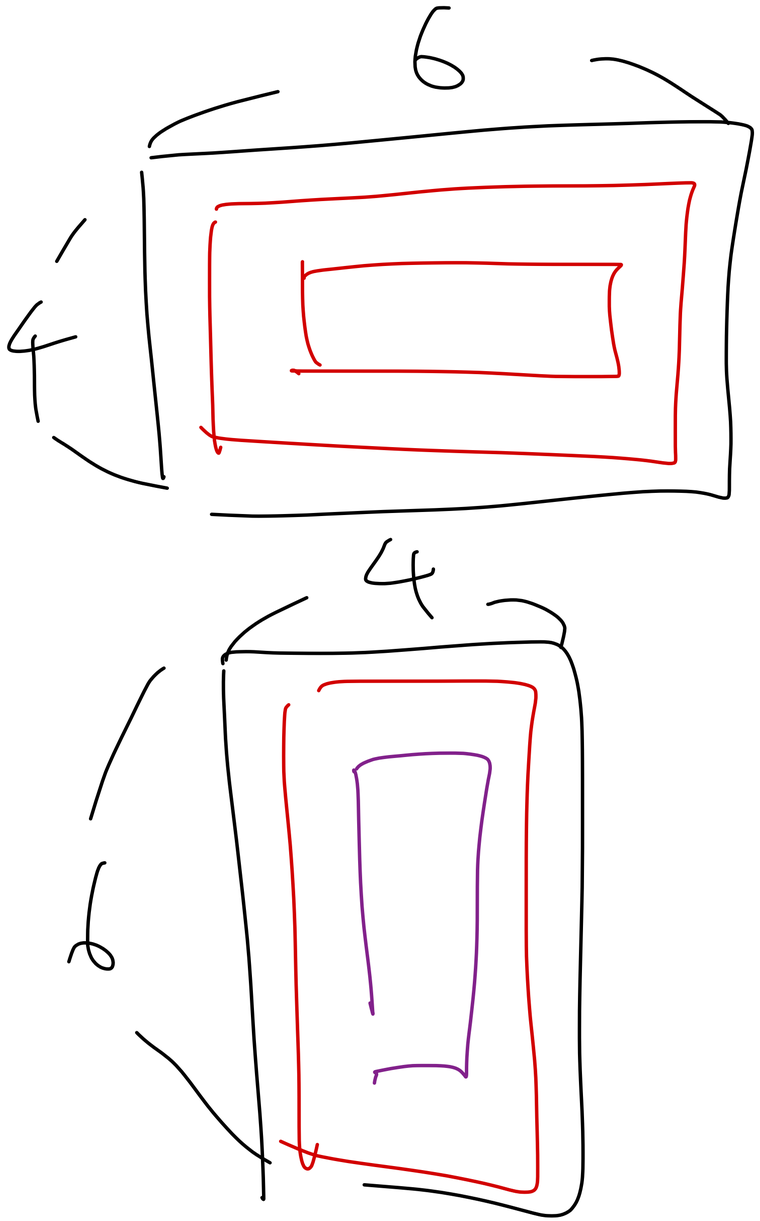
$\uparrow$ $\downarrow$

$(N-i+1, i) \leftarrow (N-i+1, M-i+1)$

Java

```java
import java.util.*;
public class Main {
    static int n,m,r; // n*m, r
    static int[][] a;
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        m = sc.nextInt();
        r = sc.nextInt();
        a = new int[n+1][m+1];
        for (int i=1; i<=n; i++) {
            for (int j=1; j<=m; j++) {
                a[i][j] = sc.nextInt();
            }
        }
        int groups = Math.min(n,m)/2;
        for (int g=1; g<=groups; g++) {
            // (g,g)   ->   ...   -> (g,m-g+1)
            //  ^                      |
            //  |                      |
            //  |                      v
            // (n-g+1,g) <- ... <- (n-g+1, m-g+1)
            ArrayList<Integer> group = new ArrayList<>();
            for (int j=g; j<m-g+1; j++) {
                group.add(a[g][j]);
            }
            for (int i=g; i<n-g+1; i++) {
                group.add(a[i][m-g+1]);
            }
            for (int j=m-g+1; j>g; j--) {
                group.add(a[n-g+1][j]);
            }
            for (int i=n-g+1; i>g; i--) {
                group.add(a[i][g]);
            }
            Collections.rotate(group, (int)group.size()-r);
            int index = 0;
            for (int j=g; j<m-g+1; j++) {
                a[g][j] = group.get(index++);
            }
            for (int i=g; i<n-g+1; i++) {
                a[i][m-g+1] = group.get(index++);
            }
            for (int j=m-g+1; j>g; j--) {
                a[n-g+1][j] = group.get(index++);
            }
            for (int i=n-g+1; i>g; i--) {
                a[i][g] = group.get(index++);
            }
        }
        for (int i=1; i<=n; i++) {
            for (int j=1; j<=m; j++) {
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

결과	메모리	시간	코드 길이
채점 중			1870 B



224

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