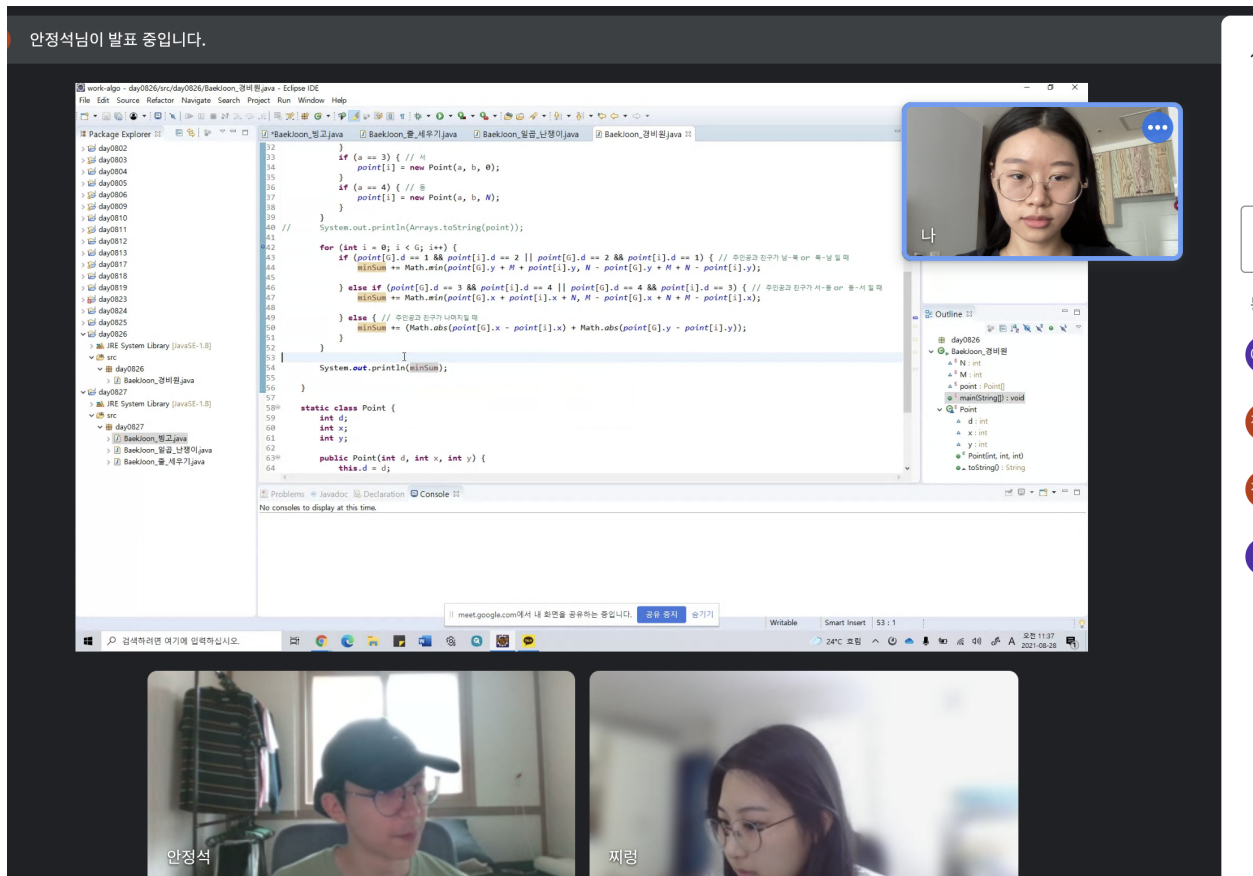


5조 결과물 제출



- 백준 IM 대비 어려운 문제 4개씩 리뷰
- 서로 다른 로직에 대해 부가 설명++
- stackoverflow 날 수 있는 상황 설명
- 시간 낭비했던 부분, 코드 줄일 수 있는 부분 공유

예인

8월 4째주 스터디

+ :: <https://www.acmicpc.net/problem/2309>

일곱난쟁이

1. 9개 중 2개를 뽑는 조합 → 조건 맞으면 출력
2. 브루투포스

실수했던점

- 재귀 호출. 한번 출력이 되고 난 후 재귀로 돌아가게 되는데,, 다음번 조건맞은 재귀에서 걸리게 되면 또 print() 가 되서 출력이 오버될 수 있음.

```
*boj2309_조합.java X *boj2309_브루투포스... boj2605.java *boj2578.java *bo
1 package algo;
2
3 import java.util.Arrays;
4
5
6 public class boj2309_조합 {
7     static int[] arr;
8     static boolean[] check;
9     static boolean flag;
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        arr = new int[9];
13        for(int i=0; i<9; i++) {
14            arr[i] = sc.nextInt();
15        } //=====>input end
16
17        Arrays.sort(arr); //먼저 정렬해줌
18        check = new boolean[9];
19        comb(0,0); //현재 인덱스
20    }
21    private static void comb(int idx, int cnt) {
22        if(cnt == 2) {
23            //조건맞는 경우 출력
24            for(int i=0; i<9; i++) {
25                if(check[i]) {
26                    System.out.print(arr[i] + " ");
27                }
28            }
29            System.out.println();
30            flag = true;
31        }
32        if(idx == 9) {
33            return;
34        }
35        check[idx] = true;
36        comb(idx+1, cnt+1);
37        check[idx] = false;
38        comb(idx+1, cnt);
39    }
40}
```

지언

```

// 2309. 일곱 난쟁이
public class Test2309 {
    static int[] input, answer;

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        input = new int[9];
        answer = new int[7];
        for (int i = 0; i < 9; i++) {
            input[i] = Integer.parseInt(br.readLine());
        } // input end
        combination(0, 0, 0);

        Arrays.sort(answer);
        for (int i : answer) {
            System.out.println(i);
        }
    } // main end

    private static boolean combination(int cnt, int start, int sum) {
        if (cnt == 7) {
            if (sum == 100) {
                return true;
            }
            return false;
        }
        for (int i = start; i < input.length; i++) {
            answer[cnt] = input[i];
            if (combination(cnt + 1, i + 1, sum + input[i])) {
                return true;
            }
        }
        return false;
    }
}

```

정석

```

import java.io.BufferedReader;

public class BaekJoon_밀급_난쟁이 {

    static int arr[];
    static boolean used[];
    static ArrayList<Integer> finalArr;

    public static void main(String[] args) throws NumberFormatException, IOException {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        arr = new int[9];
        finalArr = new ArrayList<Integer>();
        for (int i = 0; i < 9; i++) {
            arr[i] = Integer.parseInt(br.readLine());
        }
        Arrays.sort(arr);
        used = new boolean[9];
        comb(0, 0);
        for (int i = 0; i < 7; i++) {
            if (finalArr.get(i) != 0) {
                System.out.println(finalArr.get(i)); // true가 된 애들이 자극자극 저장되어 있기때문에 처음 들어왔던 7개만 출력해준다.
            }
        }

    }

    static void comb(int target, int cnt) {
        if (cnt == 7) {
            int sum = 0;
            for (int i = 0; i < used.length; i++) {
                if (used[i]) {
                    sum += arr[i];
                }
            }
            if (sum == 100) {
                for (int i = 0; i < used.length; i++) {
                    if (used[i]) {
                        System.out.println(arr[i]); // 100이 되는 모든 쌍을 찾으면 안되고 한쌍을 찾는 순간 끝내야한다. 따라서 배열을 하나 만들어서 최종으로 저장해줄거다.
                        finalArr.add(arr[i]);
                    }
                }
            }
            return;
        }
        if (target > 8) {
            return;
        }

        used[target] = true;
        comb(target + 1, cnt + 1);
        used[target] = false;
        comb(target + 1, cnt);
    }
}

```

```

package day0827;

import java.io.BufferedReader;

public class BaekJoon_빙고 {

    static int map[][];
    static int bingoCnt;
    static int numCnt;

    public static void main(String[] args) throws IOException {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        map = new int[5][5];

        for (int i = 0; i < 5; i++) {
            StringTokenizer st = new StringTokenizer(br.readLine());
            for (int j = 0; j < 5; j++) {
                map[i][j] = Integer.parseInt(st.nextToken());
            }
        }

        Loop: for (int i = 0; i < 5; i++) {
            StringTokenizer st = new StringTokenizer(br.readLine());
            for (int j = 0; j < 5; j++) {
                numCnt++;
                delete(Integer.parseInt(st.nextToken())); // 입력값 하나 하나 삭제하면서 합수로 전체 검색
                rowBingo();
                columnBingo();
                rightDiagonalBingo();
                leftDiagonalBingo();
                if (bingoCnt >= 3) {
                    System.out.println(numCnt);
                    break Loop;
                }

                bingoCnt = 0; // 빙고의 숫자가 중복되지 않게 탐색이 끝나면 0으로 초기화
            }
        }

        static void delete(int a) {
            Loop: for (int i = 0; i < 5; i++) {
                for (int j = 0; j < 5; j++) {
                    if (map[i][j] == a) {
                        map[i][j] = 0;
                        break Loop;
                    }
                }
            }
        }
    }
}

```

```

    static void rowBingo() {
        int cnt = 0;
        for (int i = 0; i < 5; i++) {
            cnt = 0;
            for (int j = 0; j < 5; j++) {
                if (map[i][j] == 0) {
                    cnt++;
                }
                if (cnt == 5) {
                    bingoCnt++;
                }
            }
        }
    }

    static void columnBingo() {
        int cnt = 0;
        for (int j = 0; j < 5; j++) {
            cnt = 0;
            for (int i = 0; i < 5; i++) {
                if (map[i][j] == 0) {
                    cnt++;
                }
                if (cnt == 5) {
                    bingoCnt++;
                }
            }
        }
    }

    static void rightDiagonalBingo() {
        int cnt = 0;
        int a = 0;
        for (int i = 0; i < 5; i++) {
            if (map[a][4 - a] == 0) {
                cnt++;
            }
            a++;
        }
        if (cnt == 5) {
            bingoCnt++;
        }
    }

    static void leftDiagonalBingo() {
        int cnt = 0;
        int a = 0;
        for (int i = 0; i < 5; i++) {
            if (map[a][a] == 0) {
                cnt++;
            }
            a++;
        }
        if (cnt == 5) {
            bingoCnt++;
        }
    }
}

```