

9월 3주차 스터디 발표 - 3A

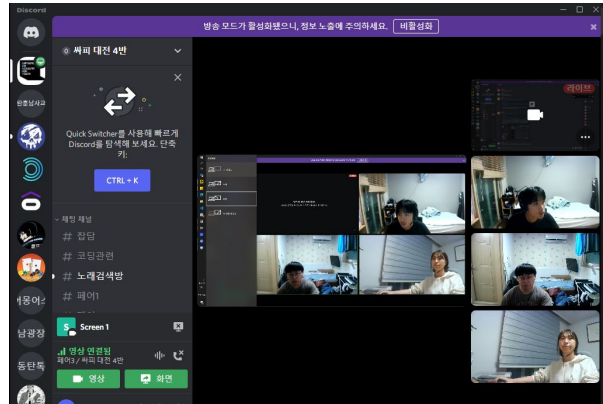
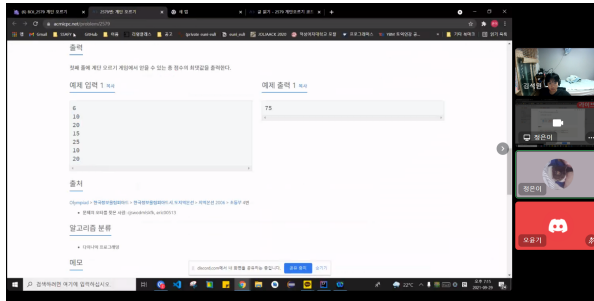


발표일자 : 2021년 9월 29일 수요일 19:00~ 20:00

발표 순서

1. 백준 9465 스티커 실버 2 김석원
2. 백준 2579 계단오르기 실버 3 정은이
3. 백준 12865 평범한 배낭 골드 5 오윤기

스터디 사진



1. 9465 스티커

- 발표자 김석원

```
package week9_3;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Scanner;
import java.util.StringTokenizer;

public class B_9465스티커 {

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int T = Integer.parseInt(br.readLine());
```

```

for(int t = 1; t <= T; t++ ) {
    int col = Integer.parseInt(br.readLine());

    int r1= 0;
    int r2 =0;
    int[][] arr= new int[2][col];
    int[][] dp= new int[2][col];

    for(int i = 0; i < 2; i++) {
        StringTokenizer st = new StringTokenizer(br.readLine());
        for(int j = 0; j < col; j++) {
            arr[i][j] = Integer.parseInt(st.nextToken());
        }
    } // end input
    //dp 초기화
    if(col == 1) {
        System.out.println(Integer.max(arr[0][0],arr[1][0]));
        continue;
    }
    dp[0][0] = arr[0][0];
    dp[1][0] = arr[1][0];
    dp[0][1] = dp[1][0] + arr[0][1];
    dp[1][1] = dp[0][0] + arr[1][1];
    if(col == 2) {
        System.out.println(Integer.max(dp[0][1],dp[1][1]));
        continue;
    }

    //점화식 dp[i][n] = dp[i+1%2][n-1] + arr[i][n]
    // dp[i][n] = max(dp[0][n-2],dp[1][n-2]) + arr[i][n];

    for(int c = 2; c < col; c++) {
        for(int i = 0; i < 2; i++) {
            r1 = dp[(i+1) %2][c-1] + arr[i][c];
            r2 = Integer.max(dp[0][c-2],dp[1][c-2]) + arr[i][c];
            dp[i][c] = Integer.max(r1, r2);
        }
    }

    System.out.println(Integer.max(dp[0][col-1], dp[1][col-1]));

} // end testcase
}
}

```

2. 2579 계단오르기

- 발표자 정은이



점화식 컨셉

dp[n] = 각 칸에 최고로 도달할 수 있는 점수

칸에 도달할 수 있는 조건

- 1칸 또는 2칸으로 오를 수 있음
- 연달아 1칸씩 오르는 것은 불가능

dp[n] 에 도달할 수 있는 경우의 수는 2 가지.

1. 2칸 전 $dp[n-2]$
2. 3칸전에서 1칸전으로 온 후 $dp[n-3] + (n-1\text{의 계단 점수})$

점화식

```
dp[i]=Math.max(stairs[i-1]+dp[i-3>0 ? i-3:0],dp[i-2])+stairs[i];
```

```
import java.util.Scanner;

public class Java_2579 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        int[] stairs = new int[T+1];
        int[] dp = new int[T+1]; // 각 칸에 최고로 도달할 수 있는 점수
        for (int i = 1; i <= T; i++) {
            stairs[i] = sc.nextInt();
        }

        dp[1] = stairs[1]; // 첫번째 칸 기록
        for (int i = 2; i <= T; i++) {
            dp[i]=Math.max(stairs[i-1]+dp[i-3>0 ? i-3:0],dp[i-2])+stairs[i];
            // dp[i-3>0 ? i-3:0] // i-3이 음수라면 dp[0] - 시작점
            // 도착점에는 2-1 칸, 2칸 으로 온 계단 점수중에 큰 점수 기록
        }
        System.out.print(dp[T]);
    }
}
```

3. 12865 평범한 배낭

- 발표자 오윤기

DP ver. 2차원 배열

```
package day0929;

import java.util.Arrays;
import java.util.Scanner;

public class Main_백준12865_평범한배낭 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int W = sc.nextInt();

        int[] weights = new int[N + 1];
        int[] values = new int[N + 1];

        for (int i = 1; i <= N; i++) {
            weights[i] = sc.nextInt();
            values[i] = sc.nextInt();
        }

        int[][] DP = new int[N + 1][W + 1];

        for (int i = 1; i <= N; i++) {
            for (int w = 1; w <= W; w++) {
                if (weights[i] <= w) { //가방에 넣을 수 있을 때
                    DP[i][w] = Math.max(DP[i - 1][w], values[i] + DP[i - 1][w - weights[i]]);
                    //해당 물건을 넣고 이전상태에서 남은무게의 가치의 합과 이전상황 무게에서의 가치 합 중 큰 것을 선택한다.
                } else { // 가방에 넣을 수 없을 때
                    DP[i][w] = DP[i - 1][w];
                }
            }
        }
        for(int i = 1; i<=N; i++) {
            System.out.println(i + "까지 넣었을 때의 상황: " +Arrays.toString(DP[i]));
        }
        System.out.println(DP[N][W]);
        sc.close();
    }
}
```

DP ver. 1차원 배열

```
package day0929;

import java.util.Arrays;
import java.util.Scanner;

public class Main_백준12865_평범한배낭2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```

int N = sc.nextInt();
int W = sc.nextInt();

int[] weights = new int[N + 1];
int[] profits = new int[N + 1];

for (int i = 1; i <= N; i++) {
    weights[i] = sc.nextInt();
    profits[i] = sc.nextInt();
}

int[] D = new int[W + 1];
// 가치판단할 때만 유리 , 어떤물건 있는지는 확인 힘듐
for (int i = 1; i <= N; i++) {
    for (int w = W; w >= weights[i]; w--) {
        D[w] = Math.max(D[w], profits[i] + D[w - weights[i]]);
    }
    System.out.println(i + "까지 넣었을 때의 상황: " + Arrays.toString(D));
}
System.out.println(D[W]);
sc.close();
}
}

```