

2025.12.19

한 학기 정리

<https://github.com/lgw323/ecolecapstonedesign>

학번: 2021663046

성명: 이건우

WEEK 1 오후

미팅해서 정보 뽑기

- 비전문가인 클라이언트의 말에서 필요한거 뽑기

뽑은거 다듬기

- 사이트 기능이 많아지면 목표에서 떨어짐
- 개발자 중심에서 사용자 중심으로 생각하기
- 내가 이런거 할 수 있음 → 기능 만들기가 아닌,
- 만들어야 하는 것 → 내가 할 수 있는게 있음. 이 방식으로 생각하기

프로젝트 기획서 만들기

- 난 기억 못하니까 만들어 둘 것
- 구조
 - 프로젝트 목표
 - 타겟층
 - 연령, 성별, 방문시간, 성향
 - 작업 계획표
 - 클라이언트 요청사항

UI/UX

외부 강사 강의

imager v1.7.2



Raspberry Pi

운영체제

저장소

운영체제 OS 선택

저장소 선택

쓰기

OS Setup

라즈베리 파이 이미저

WEEK 1 새벽

문제 8: 카드2 (BOJ 2164)

문제 설명
1부터 N까지의 카드를 쌓고, "맨 위"

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.util.LinkedList;
import java.util.Queue;

public class ecole007 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        Queue<String> queue = new LinkedList();
        int n = Integer.parseInt(br.readLine());
        for (int i = 1; i < n; i++) {
            queue.offer(i);
        }
        while (queue.size() > 1) {
            queue.poll();
            int num = queue.peek();
            queue.offer(num);
        }
        System.out.println(queue.poll());
    }
}
```

문제 7: 스택 수열 (BOJ 1874)

문제 설명
1부터 n까지의 수를 스택에 오름차순으로 쌓을 때

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Stack;

public class ecole006 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        Stack<Integer> stack = new Stack();
        int n = Integer.parseInt(br.readLine());
        for (int i = 1; i < n; i++) {
            stack.push(i);
        }
        while (stack.size() > 1) {
            stack.pop();
            int num = stack.peek();
            stack.push(num);
        }
        System.out.println(stack.pop());
    }
}
```

문제 6: 최솟값 찾기 (BOJ 11003)

문제 설명
N개의 수가 있을 때, 크기 L의 슬라이딩 윈도우

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.ArrayDeque;
import java.util.StringTokenizer;

public class ecole006 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int l = Integer.parseInt(st.nextToken());
        ArrayDeque<Integer> deque = new ArrayDeque();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = Integer.parseInt(br.readLine());
        }
        for (int i = 0; i < n; i++) {
            if (i < l - 1) {
                continue;
            }
            deque.addLast(arr[i]);
            if (deque.size() == l) {
                System.out.print(deque.peek() + " ");
            }
        }
    }
}
```

문제 5: 좋다 (BOJ 1253)

문제 설명
N개의 수 중에서, 어떤 수가 서로 다른 수들보다 더 좋다.

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.Arrays;
import java.util.StringTokenizer;

public class ecole005 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = Integer.parseInt(br.readLine());
        }
        Arrays.sort(arr);
        for (int i = 0; i < n; i++) {
            int count = 0;
            for (int j = i + 1; j < n; j++) {
                if (arr[i] < arr[j]) {
                    count++;
                }
            }
            if (count == n - i - 1) {
                bw.write(arr[i] + " ");
            }
        }
        bw.close();
    }
}
```

문제 4: 연속된 자연수의 합 (BOJ 2018)

문제 설명
자연수 N을 연속된 자연수들의 합으로 표기하는 경우

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

public class ecole004 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        long[] arr = new long[n];
        for (int i = 0; i < n; i++) {
            arr[i] = Long.parseLong(br.readLine());
        }
        for (int i = 0; i < n; i++) {
            Arrays.sort(arr);
            int count = 1;
            int sum = arr[i];
            int sidx = i;
            int eidx = i;
            while (sum < arr[n - 1]) {
                eidx++;
                sum += arr[eidx];
                if (eidx == n - 1) {
                    break;
                }
            }
            if (sum == arr[n - 1]) {
                for (int j = sidx; j <= eidx; j++) {
                    System.out.print(arr[j] + " ");
                }
                System.out.println();
            }
        }
    }
}
```

문제 3: 구간 합 구하기 4 (BOJ 11659)

문제 설명
N개의 수가 주어지고, M개의 구간을 찾는 문제

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

public class ecole003 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        long[] arr = new long[n];
        for (int i = 0; i < n; i++) {
            arr[i] = Long.parseLong(br.readLine());
        }
        int count = 1;
        int sum = 0;
        int sidx = 0;
        int eidx = 0;
        while (eidx != n) {
            if (sum == n) {
                for (int i = sidx; i <= eidx; i++) {
                    System.out.print(arr[i] + " ");
                }
                System.out.println();
                count++;
                sum = 0;
                sidx = eidx + 1;
                eidx++;
            } else if (sum < n) {
                eidx++;
                sum += arr[eidx];
            } else if (sum > n) {
                sum -= arr[sidx];
                sidx++;
            }
        }
    }
}
```

문제 2: 숫자의 합 (BOJ 11720)

문제 설명
N개의 숫자가 공백 없이 하나의 문자열로 주어지면 그 합을 출력하는 문제

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

public class ecole002 {
    public static void main()
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int result = 0;
        for (String string : st) {
            result += Integer.parseInt(string);
        }
        System.out.println(result);
    }
}
```

문제 1: 수 정렬하기 2 (BOJ 2751)

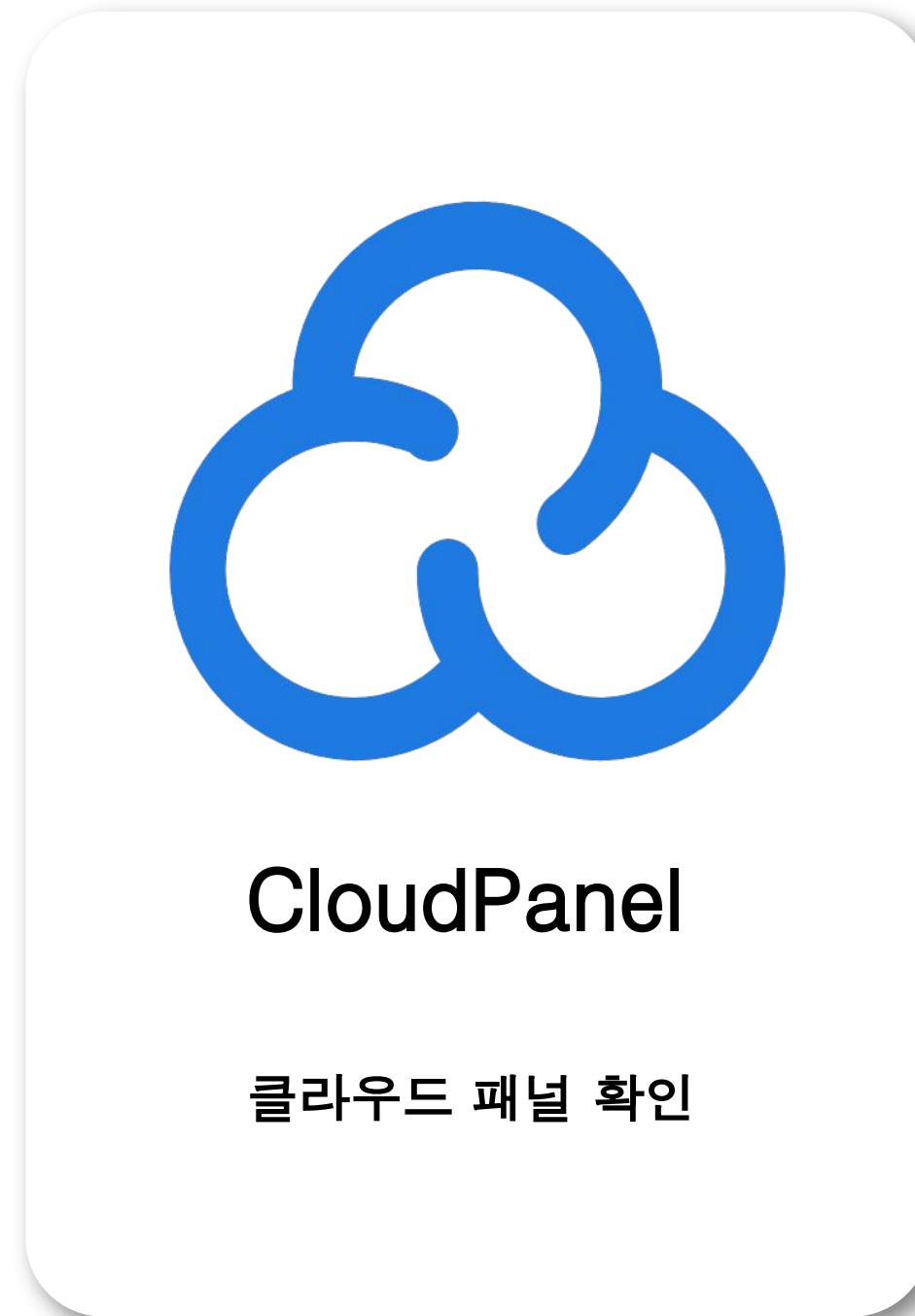
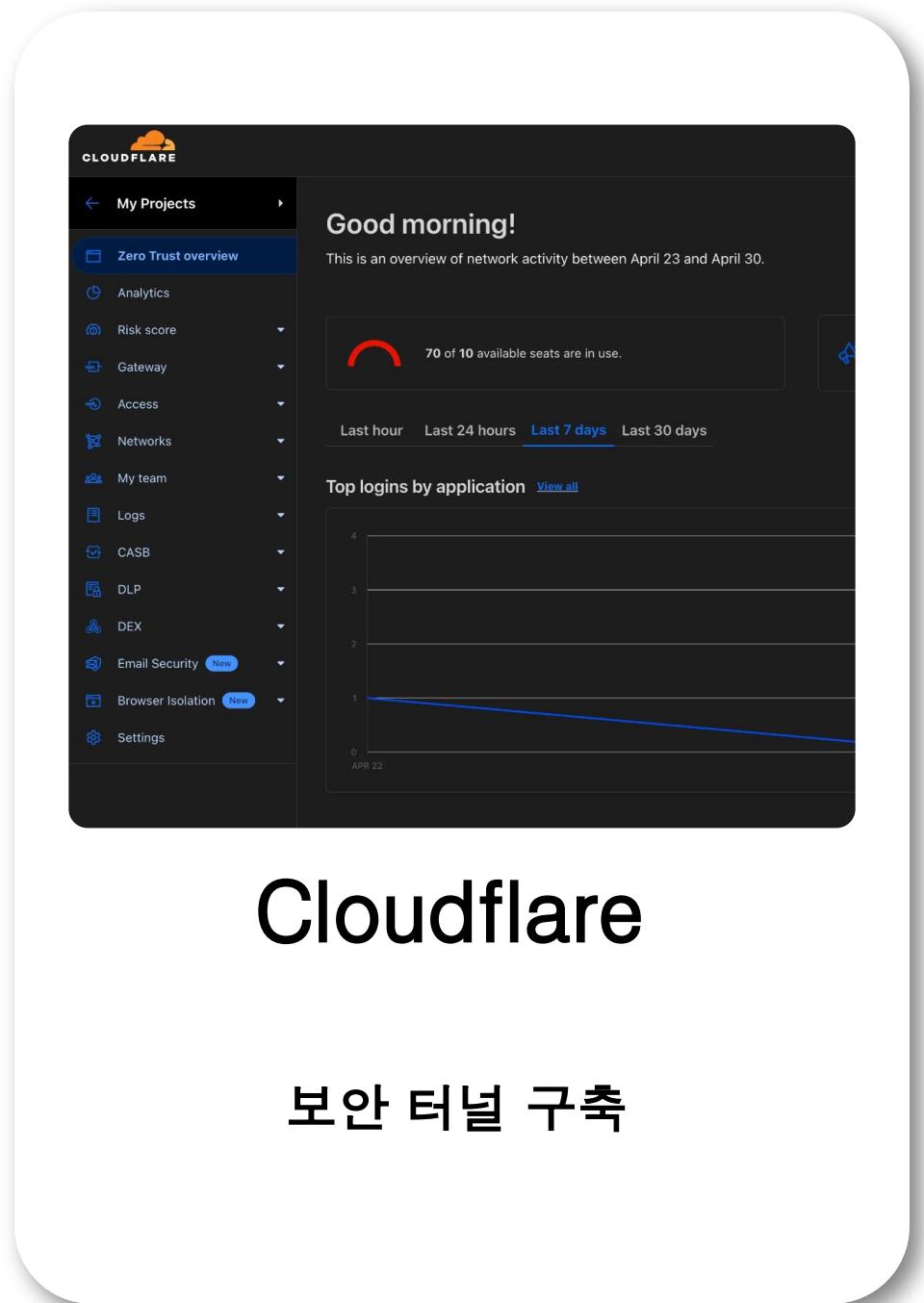
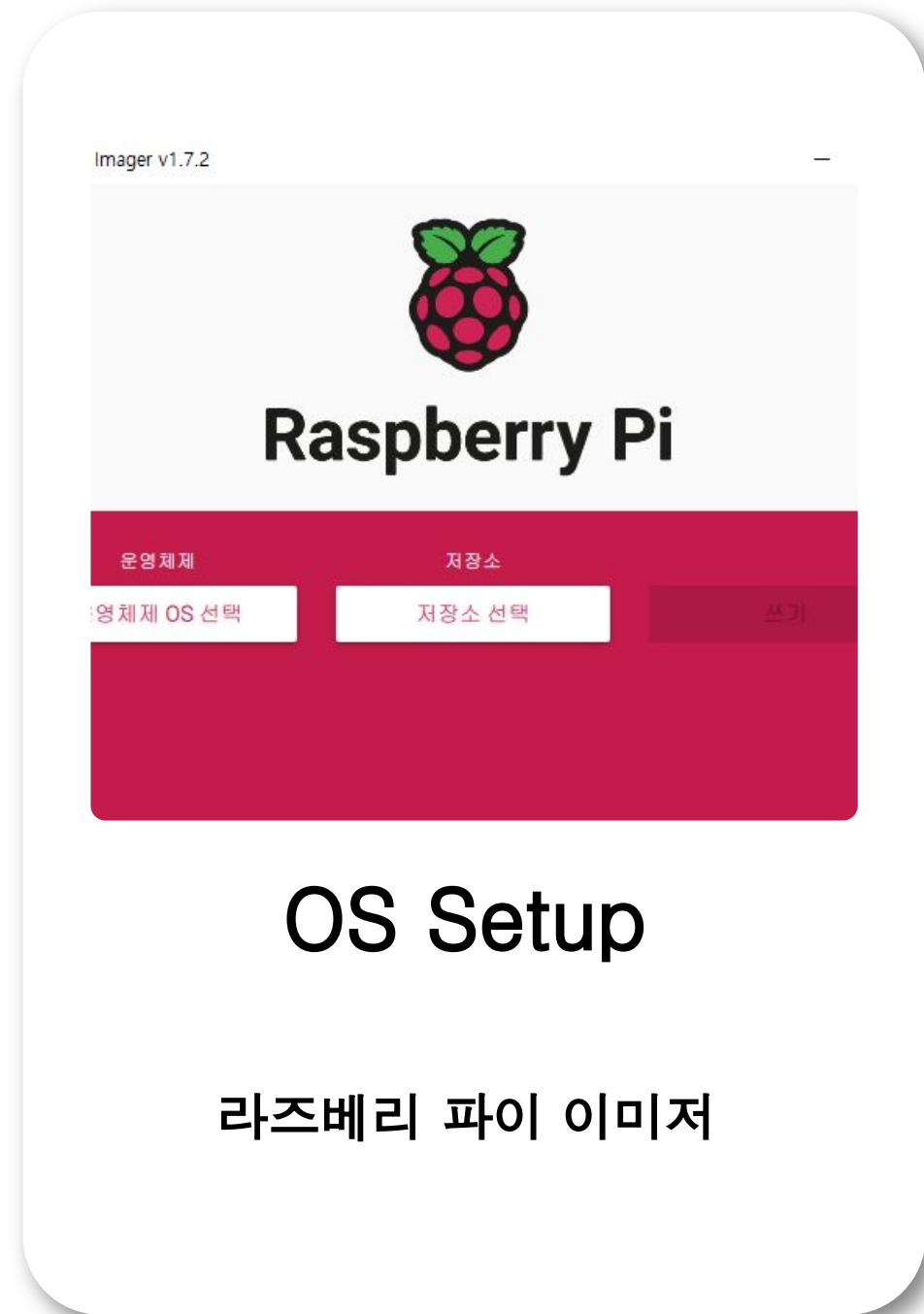
문제 설명
N개의 수가 주어졌을 때, 이 수들을 오름차순으로 정렬하는 문제입니다. 시간 복잡도가 중요합니다.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;

public class ecole001 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        int[] arr = new int[n];
        String lines[] = br.readLine().split(" ");
        for (String line : lines) {
            arr[Integer.parseInt(line)] = 1;
        }
        int result = 0;
        for (int i = 1; i < n; i++) {
            if (arr[i] == 1) {
                result += i;
            }
        }
        System.out.println(result);
    }
}

int mini, maxs;
for (int i = 0; i < n; i++) {
    st = new StringTokenizer(br.readLine());
    mini = Integer.parseInt(st.nextToken());
    maxs = Integer.parseInt(st.nextToken());
    if (mini == 0) {
        System.out.println(maxs);
    } else {
        System.out.println(mini);
    }
}
Arrays.sort(arr);
for (int i : arr) {
    System.out.println(i);
}
}
```

WEEK 2 오후



WEEK 2 새 봄

알고리즘: DFS와 BFS

알고리즘: DFS

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.*;

public class Main {
    static ArrayList<Integer> adjList = new ArrayList<Integer>();
    static boolean[] visited = new boolean[adjList.size()];
    static StringBuilder sb = new StringBuilder();

    public static void main() {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int m = Integer.parseInt(st.nextToken());
        int v = Integer.parseInt(st.nextToken());
        adjList = new ArrayList<Integer>(n);
        for (int i = 1; i <= n; i++) {
            adjList.add(i);
        }
        for (int i = 0; i < m; i++) {
            st = new StringTokenizer(br.readLine());
            int u = Integer.parseInt(st.nextToken());
            int w = Integer.parseInt(st.nextToken());
            adjList.get(u).add(w);
            adjList.get(w).add(u);
        }
    }
    for (int i = 1; i <= n; i++) {
        visited[i] = false;
    }
    dfs(v);
}

void dfs(int v) {
    visited[v] = true;
    sb.append(v + " ");
    for (int i : adjList.get(v)) {
        if (!visited[i]) {
            dfs(i);
        }
    }
}
```

알고리즘: 카운팅/기수 정렬

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

public class Main {
    static ArrayList<Integer> adjList = new ArrayList<Integer>();
    static boolean[] visited = new boolean[adjList.size()];
    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int m = Integer.parseInt(st.nextToken());
        for (int i = 0; i < n; i++) {
            adjList.add(i);
            visited[i] = false;
        }
        for (int i = 0; i < m; i++) {
            st = new StringTokenizer(br.readLine());
            int u = Integer.parseInt(st.nextToken());
            int v = Integer.parseInt(st.nextToken());
            adjList.get(u).add(v);
            adjList.get(v).add(u);
        }
    }
}

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.StringTokenizer;

public class Main {
    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));
        int n = Integer.parseInt(br.readLine());
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = i;
        }
        radixSort(arr, 5);
        for (int val : arr) {
            bw.write(val + " ");
        }
        bw.flush();
        bw.close();
    }
}
```

알고리즘: 병합 정렬

```
import java.io.*;
import java.io.*;
import java.io.*;
import java.io.*;
import java.io.*;

public class Main {
    private static void
    public static void main()
        BufferedReader
        BufferedReader
        int n =
        int[] arr =
        temp = r
        for (int i = 0; i < n; i++)
            mergeSort(arr, 0, n - 1);
        for (int i = 0; i < n; i++)
            bw.write(arr[i] + " ");
        bw.flush();
        bw.close();
        br.close();
    }
    private static void
        if (left == right)
            int mid =
            mergeSort(left, mid);
            mergeSort(mid + 1, right);
            merge(left, mid, right);
    }
}
```

고리즘: 퀵 선택

```
 알고리즘: 삽입 정렬

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.util.StringTokenizer;

public class Main {
    public static void main() {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int want = Integer.parseInt(st.nextToken());
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = Integer.parseInt(br.readLine());
        }
        quickSort(arr, 0, n - 1);
        System.out.println(arr[want - 1]);
    }

    public static void quickSort(int[] arr, int left, int right) {
        if (left >= right)
            return;
        int pivotIndex = partition(arr, left, right);
        if (want == pivotIndex)
            return;
        else if (pivotIndex < want)
            quickSort(arr, pivotIndex + 1, right);
        else if (pivotIndex > want)
            quickSort(arr, left, pivotIndex - 1);
    }

    private static int partition(int[] arr, int left, int right) {
        int pivot = arr[left];
        int j = left + 1;
        for (int i = left + 1; i <= right; i++) {
            if (arr[i] < pivot) {
                swap(arr, i, j);
                j++;
            }
        }
        swap(arr, left, j - 1);
        return j - 1;
    }

    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}
```

고리즘: 선택 정렬

```
rt java.io.BufferedReader;
rt java.io.IOException;
rt java.io.InputStreamReader;

ic class Main {
public static void main()
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    String lines = br.readLine();
    int[] arr = new int[lines];
    for (int i = 0; i < lines; i++)
        arr[i] = Integer.parseInt(br.readLine());
}

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
    public static void main(String args[])
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        int[] arr = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = Integer.parseInt(br.readLine());
        for (int i = 0; i < n - 1; i++)
            System.out.print(arr[i] + " ");
        System.out.println(arr[n - 1]);
}
```

Java Python JavaScript

```
main {
    BufferedReader br;
    String[] lines = br.readLine().split(" ");
    int[] arr = new int[lines.length];
    for (int i = 0; i < lines.length; i++) {
        arr[i] = Integer.parseInt(lines[i]);
    }
    for (int i = 0; i < arr.length - 1; i++) {
        for (int j = 0; j < arr.length - 1 - i; j++) {
            if (arr[j] > arr[j + 1]) {
                int tmp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = tmp;
            }
        }
    }
    for (int i : arr) {
        System.out.println(i);
    }
}
```

WEEK 3 오후

▼ Web Server

웹 플랫폼(Web Platform) 구축의 이해

교수님께서 설명해주신 웹 서버 구성은 단순히 기술 요소를 나열하는 것을 넘어, 우리가 만들 '플랫폼'의 목적과 역할을 정의하는 과정입니다.

🌐 웹의 기본 구성 요소

웹 플랫폼은 크게 세 가지 파트와 이를 지원하는 기반으로 이루어집니다.

- **프론트엔드 (Front-end)**: 사용자가 직접 보고 상호작용하는 화면. (예: 웹사이트의 디자인, 버튼, 레이아웃)
- **백엔드 (Back-end)**: 사용자 눈에 보이지 않는 서버 뒤편의 영역. 데이터 처리, 로직 계산 등 실제 기능이 이루어지는 곳입니다.
- **데이터베이스 (DB)**: 회원 정보, 게시글 등 모든 데이터를 체계적으로 저장하고 관리하는 참고입니다.

⚠️ 플랫폼을 지원하는 기반 시설

위 요소들이 동작하기 위해 반드시 필요한 외부 환경입니다.

- **서버 (Server)**: 위 모든 구성 요소가 실제로 설치되고 24시간 실행되는 컴퓨터.
- **도메인 (Domain)**: google.com처럼 사용자가 서버에 쉽게 찾아올 수 있도록 만드는 인터넷 주소.
- **네트워크 (Network)**: 사용자의 컴퓨터와 서버를 연결해주는 통신망.

웹 플랫폼

웹 플랫폼 구성요소 강의



워드 프레스

클라우드 패널에 추가

WEEK 3 새 봄

알고리즘·화자 유크리드 허제버

알고리즘: 유크리드 호제법

```
import java.io
import java.io
import java.io
import java.util

public class b {
    public static void main( String[] args ) {
        BufferedReader br = new BufferedReader( new InputStreamReader( System.in ) );
        String strLine;
        long a = 0;
        long b = 0;
        long c = 0;

        long[] d = new long[10];
        long g = 0;
        long x = 0;

        if (c > 0) {
            /
        } else {
            long s = 0;
            System.out.println( " " );
        }
    }

    private static void f( long b = 0 ) {
        long c = 0;
        long d = 0;
        long e = 0;
        long f = 0;
        long g = 0;
        long h = 0;
        long i = 0;
        long j = 0;
        long k = 0;
        long l = 0;
        long m = 0;
        long n = 0;
        long o = 0;
        long p = 0;
        long q = 0;
        long r = 0;
        long s = 0;
        long t = 0;
        long u = 0;
        long v = 0;
        long w = 0;
        long x = 0;
        long y = 0;
        long z = 0;
    }
}
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.util.StringBuilder;

public class bj9_1 {
    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int t = Integer.parseInt(br.readLine());
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < t; i++) {
            String str = br.readLine();
            int a = Integer.parseInt(str.substring(0, str.length() / 2));
            int b = Integer.parseInt(str.substring(str.length() / 2));
            long result = a * b;
            sb.append(result).append("\n");
        }
        System.out.println(sb);
    }
}

private static void swap(int[] arr, int i, int j) {
    while (i < j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        i++;
        j--;
    }
}
```

```
import java.io.  
import java.io.  
import java.io.  
  
public class bj  
    public stat  
        Buffere  
        long n  
        long ph  
  
        for (lo  
            if  
  
                }  
            }  
  
            if (n >  
                phi  
            }  
  
            System.  
        }  
    }
```

```
import java.io
import java.io
import java.io
import java.util

public class b
    public static void main(String[] args) {
        int m = 5;
        int n = 10;
        boolean result = true;
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                if (i * j == 10) {
                    result = false;
                }
            }
        }
        System.out.println(result);
    }
}
```

```
 알고리즘

import java.io.B
import java.io.I
import java.io.I

public class bj6
    public static
        BufferedReader
        String[]
        int resu

        for (int
            int
            Stri
            for

        }

        if (
        }

        } el
        }

    }
System.o
    }

}
```

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;
import java.util.Arrays;
import java.util.stream.Collectors;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int m = scanner.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        List<List<Integer>> result = new ArrayList<List<Integer>>(n);
        for (int i = 0; i < n; i++) {
            result.add(new ArrayList<Integer>());
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                result.get(i).add(arr[i] * arr[j]);
            }
        }
        System.out.println(result.stream().map(List::toString).collect(Collectors.joining("\n")));
    }
}
```

```
import java.io.*;
import java.io.*;
import java.io.*;
import java.util.*;

public class b {
    public static void main(BufferedReader br) {
        int n = Integer.parseInt(br.readLine());
        PriorityQueue<Long> pq = new PriorityQueue<Long>(n);
        for (int i = 0; i < n; i++) {
            pq.add(Long.parseLong(br.readLine()));
        }
        long sum = 0;
        while (pq.size() > 1) {
            long l = pq.poll();
            long r = pq.poll();
            long sum2 = l + r;
            sum += sum2;
            pq.add(sum2);
        }
        System.out.println(sum);
    }
}
```

```
rt java.io.B
rt java.io.I
rt java.io.L
rt java.util

ic class bj3
public static
    Buffered
    StringTo

    int n =
    int k =

    int[] co
    for (int
        coin
    }

    int cou
    for (int
        if (
            }
        }
    }
    System.c
}
```

```
import java.io.*;
import java.io.*;
import java.io.*;
import java.util.*;

public class b1 {
    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        int m = Integer.parseInt(br.readLine());

        int[] arr = new int[m];
        StringTokenizer st = new StringTokenizer(br.readLine());
        long sum = 0;
        int max = 0;
        for (int i = 0; i < m; i++) {
            arr[i] = Integer.parseInt(st.nextToken());
            sum += arr[i];
            if (arr[i] > max) {
                max = arr[i];
            }
        }

        long len = n * m;
        while (len > 0) {
            long r = len % m;
            len /= m;
            if (r > max) {
                System.out.println("NO");
                return;
            }
        }
        System.out.println("YES");
    }
}
```

라메트리 고리즘: 이

알고리즘·O이코 피 하스

알고리즘 · 알고리즘의 종류와 특징

알고리즘·그리드(저력)

알고리즘·그리기(저력)

알고리즘·그리디 (으)서스의 크'

알고리즘 · 그림자

알고리즘: 파라메트린 서치

알고리즘: 이진 탐색

Java Python JavaScript

WEEK 4 오후



워드 프레스

플러그인 설명 및
새 페이지 제작

WEEK 4 새벽

알고리즘: 다익스트라 변형

알고리즘: 다익스트라 (Dijkstra)

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

public class bj7_1852 {
    static class Node implements Comparable {
        int node;
        int weight;

        Node(int node, int weight) {
            this.node = node;
            this.weight = weight;
        }

        @Override
        public int compareTo(Node o) {
            return this.weight - o.weight;
        }
    }

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int m = Integer.parseInt(st.nextToken());
        ArrayList<Node> adjList = new ArrayList<Node>(n + 1);
        for (int i = 1; i <= n; i++) {
            adjList.add(new Node(i, Integer.MAX_VALUE));
        }

        for (int i = 0; i < m; i++) {
            st = new StringTokenizer(br.readLine());
            int u = Integer.parseInt(st.nextToken());
            int v = Integer.parseInt(st.nextToken());
            int w = Integer.parseInt(st.nextToken());
            adjList.get(u).weight = Math.min(adjList.get(u).weight, w);
        }

        PriorityQueue<Node> pq = new PriorityQueue<Node>();
        pq.add(new Node(1, 0));
        while (!pq.isEmpty()) {
            Node cur = pq.poll();
            if (cur.weight == Integer.MAX_VALUE)
                break;
            for (Node next : adjList.get(cur.node)) {
                if (next.weight > cur.weight + next.weight) {
                    next.weight = cur.weight + next.weight;
                    pq.add(next);
                }
            }
        }
    }
}
```

알고리즘: 위상 정렬 + DP

알고리즘: 위상 정렬 (Topological Sort)

알고리즘: Union-Find (분리 집합)

알고리즘: BFS 또는 DFS

알고리즘: BFS (너비 우선 탐색)

Java Python JavaScript

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

public class bj2_1712 {
    static ArrayList<ArrayList<Integer>> adjList;
    static int[] parent;
    static final int INFINITE = 1000000000;
    static final int INF = 1000000000000000000;

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int m = Integer.parseInt(st.nextToken());
        adjList = new ArrayList<ArrayList<Integer>>(n + 1);
        parent = new int[n];
        for (int i = 0; i < n; i++) {
            parent[i] = i;
        }

        for (int i = 0; i < m; i++) {
            st = new StringTokenizer(br.readLine());
            int a = Integer.parseInt(st.nextToken());
            int b = Integer.parseInt(st.nextToken());
            adjList.get(a).add(b);
            adjList.get(b).add(a);
        }

        for (int i = 0; i < n; i++) {
            StringBuilder sb = new StringBuilder();
            sb.append(i);
            for (int j : adjList.get(i)) {
                sb.append(" " + j);
            }
            System.out.println(sb);
        }
    }

    public static void dfs(int v) {
        Queue<Integer> q = new LinkedList<Integer>();
        q.add(v);
        while (!q.isEmpty()) {
            int cur = q.poll();
            for (int next : adjList.get(cur)) {
                if (parent[next] != cur) {
                    parent[next] = cur;
                    q.add(next);
                }
            }
        }
    }

    public static void union(int a, int b) {
        int pa = find(a);
        int pb = find(b);
        if (pa != pb) {
            parent[pa] = pb;
        }
    }

    public static int find(int v) {
        if (parent[v] != v) {
            parent[v] = find(parent[v]);
        }
        return parent[v];
    }
}
```

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

public class bj1_18352 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int m = Integer.parseInt(st.nextToken());
        int k = Integer.parseInt(st.nextToken());
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < k; i++) {
            sb.append(st.nextToken());
        }
        String op = sb.toString();

        ArrayList<ArrayList<Integer>> adjList = new ArrayList<ArrayList<Integer>>(n + 1);
        for (int i = 1; i <= n; i++) {
            adjList.add(new ArrayList<Integer>());
        }

        adjList.get(0).add(1);
        adjList.get(1).add(0);

        for (int i = 2; i <= n; i++) {
            adjList.get(i).add(i - 1);
            adjList.get(i - 1).add(i);
        }

        for (int i = 0; i < m; i++) {
            st = new StringTokenizer(br.readLine());
            int u = Integer.parseInt(st.nextToken());
            int v = Integer.parseInt(st.nextToken());
            int e = Integer.parseInt(st.nextToken());
            adjList.get(u).add(v);
            adjList.get(v).add(u);
            adjList.get(u).add(e);
            adjList.get(e).add(u);
            adjList.get(v).add(e);
            adjList.get(e).add(v);
        }

        Queue<Integer> q = new LinkedList<Integer>();
        q.add(0);
        int[] groups = new int[n + 1];
        int group = 1;
        while (!q.isEmpty()) {
            int cur = q.poll();
            for (int next : adjList.get(cur)) {
                if (groups[next] == 0) {
                    groups[next] = group;
                    q.add(next);
                }
            }
        }
    }
}
```

WEEK 5 오후

▼ 25.11.07 - 웹디자인 ux

1. 기본원칙

- 사용자는 나보다 조금 덜 아는 사람
- 내가 원하는것과 사용자가 원하는것은 다르다
- 사용자는 내 의도와 다르게 행동할 수 있다
- 사용자를 불안하게 하지 마라
- 모바일을 먼저 설계하라
- UX를 카피하라
 - 디자인은 그리는게 아니라 설계하는 것
- 브랜드가 아니라 사용자를 위해 고민하라
- 참조 - 휴먼 인터페이스 가이드라인(HIG)

1. 시작 튜토리얼은 넣지말기

- 넣는다면 스kip 버튼 제공하기.

2. 가이드는 텍스트 없이

- 사용자는 텍스트를 읽지 않는다.

3. 스플래시는 컨텐츠를 뺀 레이아웃으로

UI/UX

외부 강사 강의 2차시



Gaming controller



Web

appwrite

앱라이트 기초 강의

WEEK 5 새 봄

알고리즘: 트리 순회 (재귀)

알고리즘: 해시 집합 (HashSet)

```
import java.io.BufferedReader  
import java.io.IOException  
import java.io.InputStream  
import java.util.StringTokenizer  
  
public class bj7_1991 {  
    static class Node {  
        static Node[] tree;  
        static StringTokenizer st;  
  
        public static void main(String[] args) {  
            BufferedReader br = new BufferedReader(new  
                InputStreamReader(System.in));  
            int N = Integer.parseInt(br.readLine());  
            tree = new Node[N];  
            for (int i = 0; i < N; i++)  
                tree[i] = null;  
            for (int i = 0; i < N; i++)  
                st = new StringTokenizer(br.readLine());  
                char p = st.nextToken().charAt(0);  
                char l = st.nextToken().charAt(0);  
                char r = st.nextToken().charAt(0);  
                if (l != '.')  
                    tree[p - 'A'].left = tree[l - 'A'];  
                if (r != '.')  
                    tree[p - 'A'].right = tree[r - 'A'];  
            }  
            preorder('A');  
        }  
        void preorder(char c) {  
            System.out.print(c);  
            if (tree[c - 'A'].left != null)  
                preorder(tree[c - 'A'].left);  
            if (tree[c - 'A'].right != null)  
                preorder(tree[c - 'A'].right);  
        }  
        void inorder(char c) {  
            if (tree[c - 'A'].left != null)  
                inorder(tree[c - 'A'].left);  
            System.out.print(c);  
            if (tree[c - 'A'].right != null)  
                inorder(tree[c - 'A'].right);  
        }  
        void postorder(char c) {  
            if (tree[c - 'A'].left != null)  
                postorder(tree[c - 'A'].left);  
            if (tree[c - 'A'].right != null)  
                postorder(tree[c - 'A'].right);  
            System.out.print(c);  
        }  
    }  
}
```

```
import java.io.BufferedReader  
import java.io.IOException  
import java.io.InputStreamReader  
import java.util.HashSet  
import java.util.StringTokenizer  
  
public class bj6_14425 {  
    public static void main(String[] args) {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        StringTokenizer st = new StringTokenizer(br.readLine());  
        int N = Integer.parseInt(st.nextToken());  
        int M = Integer.parseInt(st.nextToken());  
  
        // 1. 집합 S를 HashSet<String>로 만든다.  
        HashSet<String> setS = new HashSet<String>();  
        for (int i = 0; i < N; i++)  
            setS.add(br.readLine());  
  
        int count = 0;  
        // 2. M개의 문자열을 받아서 S에 있는지 확인한다.  
        for (int i = 0; i < M; i++) {  
            String str = br.readLine();  
            if (setS.contains(str))  
                count++;  
        }  
        System.out.println(count);  
    }  
}
```

알고리즘: DFS

알고리즘: DFS 또는 BFS

알고리즘: 크루스칼 (Kruskal) + Union-Find

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Arrays;
import java.util.PriorityQueue;
import java.util.StringTokenizer;

public class bj3_1197 {
    static class Edge {
        int start, end;
        public Edge(int s, int e) {
            this.start = s;
            this.end = e;
        }
        @Override
        public int compareTo(Edge o) {
            return this.start - o.start;
        }
    }
    static int[] parent;
    static int find(int x) {
        if (parent[x] == -1)
            return parent[x];
        else
            return parent[x] = find(parent[x]);
    }
    static void union(int x, int y) {
        int rootX = find(x);
        int rootY = find(y);
        if (rootX != rootY) {
            if (rootX < rootY)
                parent[rootY] = rootX;
            else
                parent[rootX] = rootY;
        }
    }
}

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

public class bj2_11404 {
    static final int INF = 1000000000;

    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int n = Integer.parseInt(st.nextToken());
        int m = Integer.parseInt(st.nextToken());
        int[][] dist = new int[n][m];

        for (int i = 0; i < n; i++)
            Arrays.fill(dist[i], INF);
        dist[0][0] = 0;

        for (int i = 0; i < m; i++) {
            st = new StringTokenizer(br.readLine());
            int a = Integer.parseInt(st.nextToken());
            int b = Integer.parseInt(st.nextToken());
            int c = Integer.parseInt(st.nextToken());
            dist[a][b] = Math.min(dist[a][b], c);
        }

        for (int k = 1; k < n; k++) {
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < m; j++) {
                    if (dist[i][j] > dist[i][k] + dist[k][j])
                        dist[i][j] = dist[i][k] + dist[k][j];
                }
            }
        }
        System.out.println(dist[n-1][m-1]);
    }
}
```

알고리즘: 플로이드-워셜 (Floyd-Warshall)

알고리즘: 벨만-포드 (Bellman-Ford)

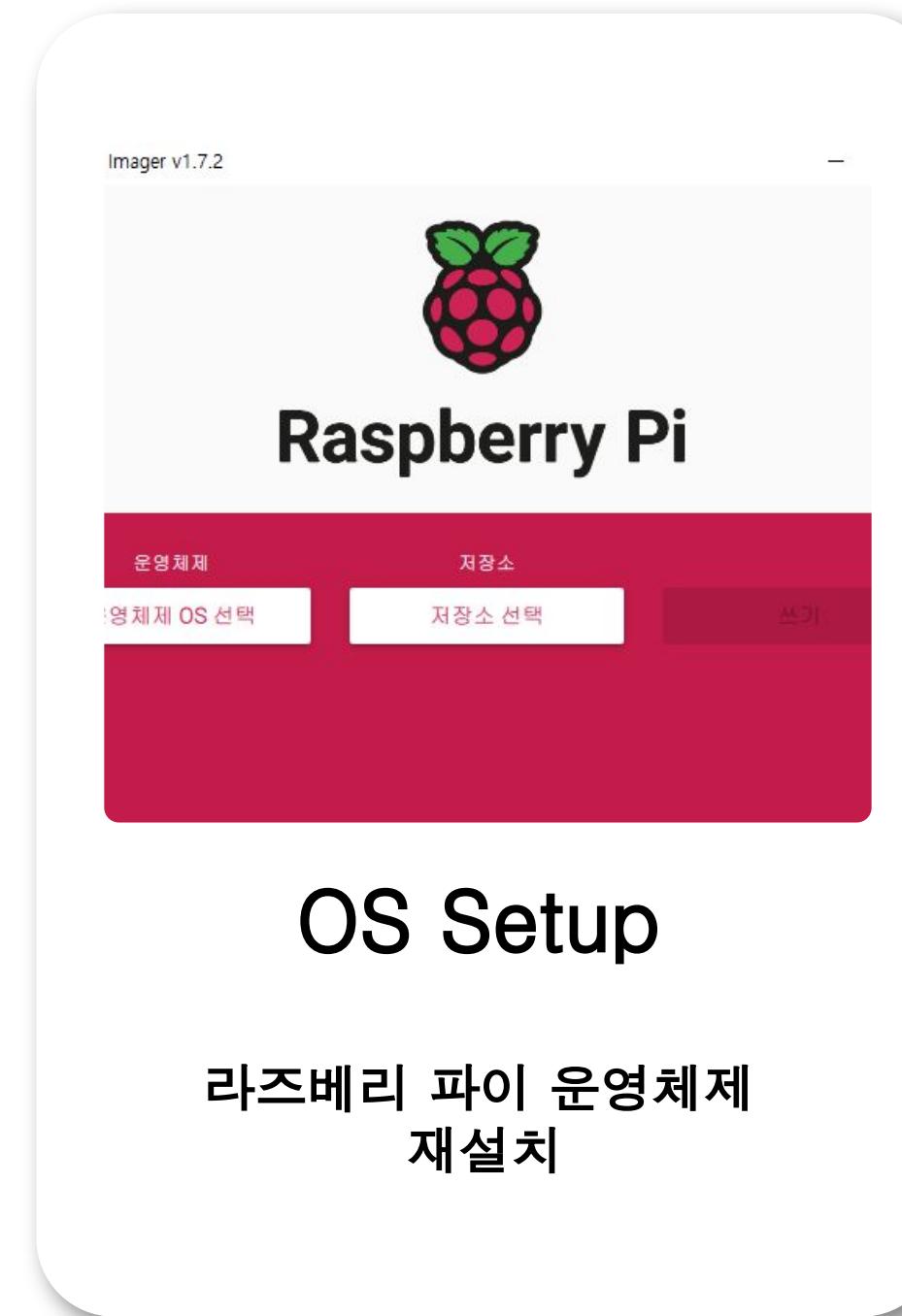
Java Python JavaScript

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

public class bj1_11657 {
    static class Edge {
        int start, end, time;
        public Edge(int start, int end, int time) {
            this.start = start; this.end = end; this.time = time;
        }
    }
    static final long INF = Long.MAX_VALUE;

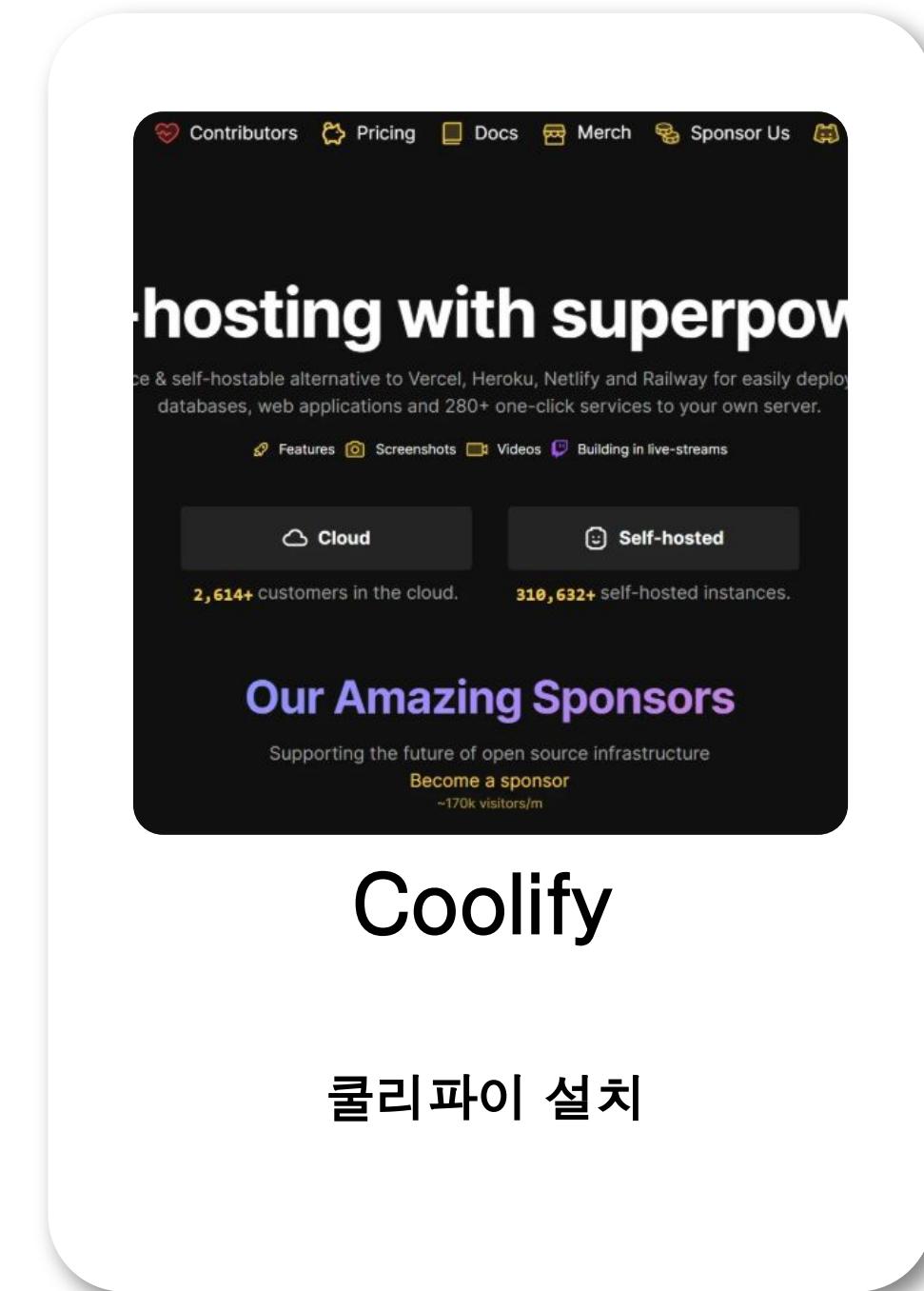
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(br.readLine());
        int N = Integer.parseInt(st.nextToken());
        int M = Integer.parseInt(st.nextToken());
        ArrayList<Edge> edges = new ArrayList<>();
        long[] dist = new long[N + 1];
        for (int i = 0; i < M; i++) {
            st = new StringTokenizer(br.readLine());
            edges.add(new Edge(Integer.parseInt(st.nextToken()), Integer.parseInt(st.nextToken())));
        }
        Arrays.fill(dist, INF);
    }
}
```

WEEK 6 오후



OS Setup

라즈베리 파이 운영체제
재설치



Coolify

쿨리파이 설치

WEEK 6 새벽

알고리즘: DP + 조합론

[Java](#) [Python](#) [Node.js](#)

알고리즘: 수학 (팩토리얼)

```
import java
import java

public clas:
public :
    Buf:
Str:
int
int
long
long
for
```

알고리즘: DP (조합)

```
import java.
import java.

public class
public s
Buf
int
Str
int
long
fact
for
boo
int[
```

알고리즘: DP (파스칼의 삼각형)

```
import java.i
import java.u
public class
public st
Buffe
int t
int[]
for (
d
d
}
for (
f
f
```

알고리즘: 수학 (팩토리얼)

```
import java.io
import java.ut
public class b
public sta
Buffer
String
int n
int k
int[][]
for (i
fo
}
```

알고리즘: 수학 LCA ($O(\log N)$)

```
import java
import java
public clas
public
Buf
Str
int
int
Sys
}
static
if
ret
}
```

알고리즘: 최소 배열 LCA ($O(\log N)$)

```
import java.io.*;
import java.util.*;

public class bj1_2042 {
    static long[] arr;
    static long[] tree;

    public :
        Buf
        int
        adj
        par
        dept
        vis:
```

알고리즘: 세그먼트 트리

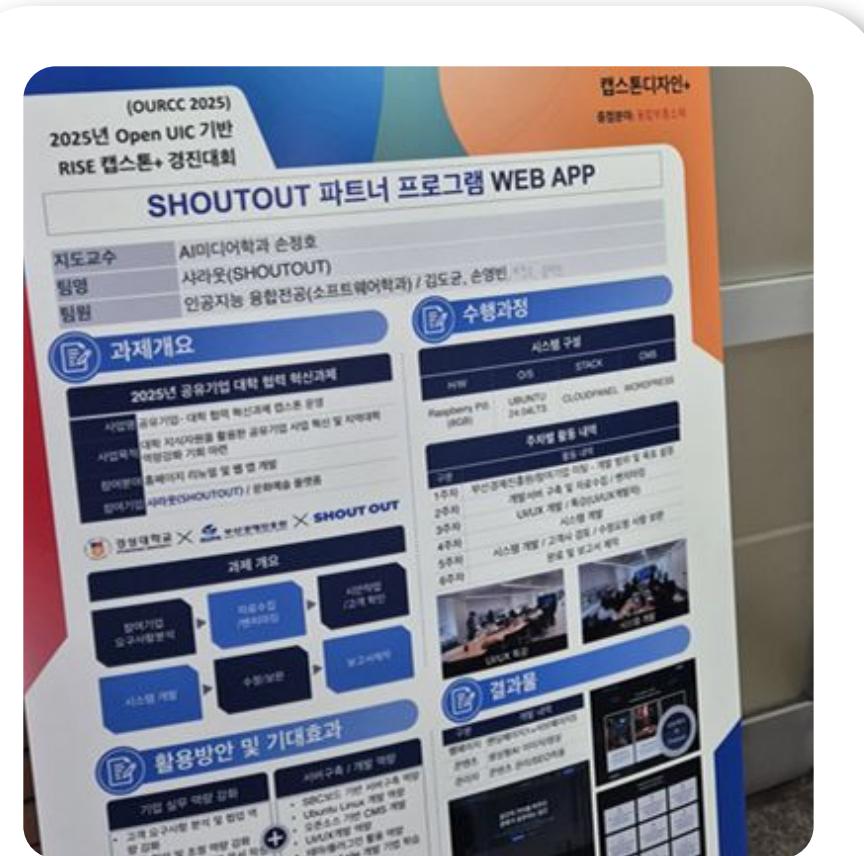
```
import java.io.*;
import java.util.*;

public static void main(String[] args) throws IOException {
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    StringTokenizer st = new StringTokenizer(br.readLine());

    int n = Integer.parseInt(st.nextToken());
    int m = Integer.parseInt(st.nextToken());
    int k = Integer.parseInt(st.nextToken());

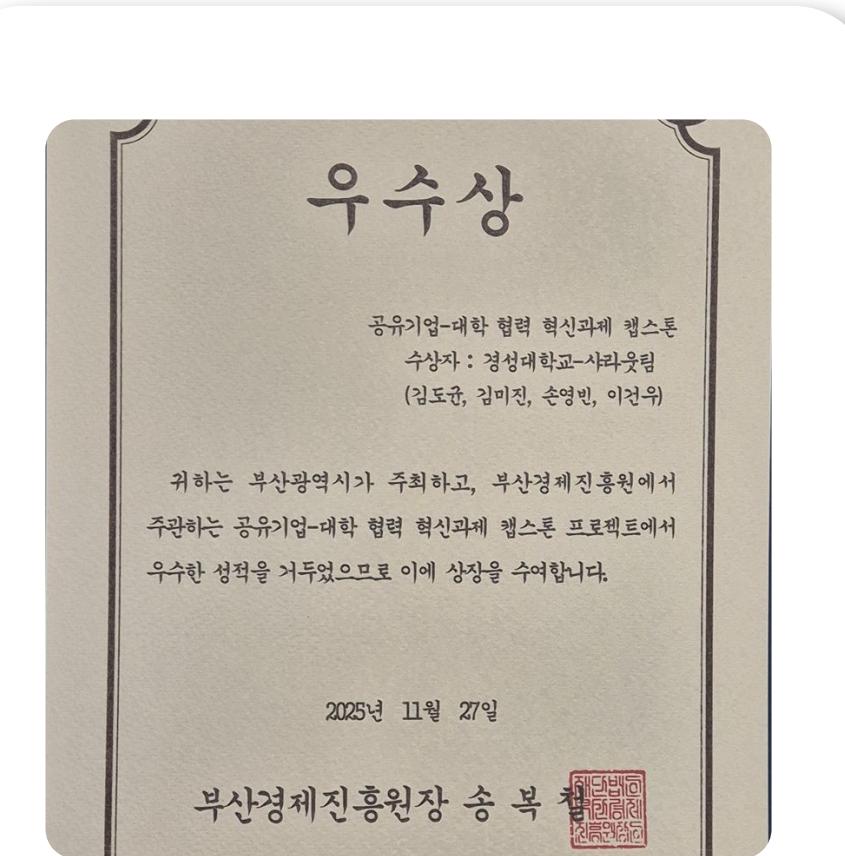
    arr = new long[n + 1];
    for (int i = 1; i <= n; i++) {
```

WEEK 7 오후



캡스톤 디자인

캡스톤 디자인 발표



우수상

야호

WEEK 7 새벽

13. 다각형의 면적 (코드)

```
i
i
p
```

12. 선분 그룹 (코드)

```
import
import
public
```

11. 선분 교차 2 (코드)

```
import
import
public
```

10. CCW (코드)

```
import
import
public
```

9. LIS 5 (코드)

```
import
import
public
```

8. 5 (코드)

```
import
import
public
```

6. 가장 큰 정사각형 (코드)

```
import
import
public
```

5. LCS 2 (코드)

```
import
import
public
```

4. 2×n 타일링 (코드)

```
import
import
public
```

3. 이친수 (코드)

```
import
import
public
```

2. 퇴사 (코드)

```
import
import
public
```

1. 1로 만들기 (코드)

```
import java.io.*;
public class bj1_1463 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        int[] dp = new int[n + 1];
        dp[1] = 0;
        for (int i = 2; i <= n; i++) {
            dp[i] = dp[i - 1] + 1;
            if (i % 2 == 0) dp[i] = Math.min(dp[i], dp[i / 2] + 1);
            if (i % 3 == 0) dp[i] = Math.min(dp[i], dp[i / 3] + 1);
        }
        System.out.println(dp[n]);
    }
}
```

깃허브 레포지토리

ecolecapstonedesign Public

Pin Watch Fork Star

main 2 Branches 0 Tags Go to file Add file Code About

Igw323 update: week7 ppt update a74c080 · 2 weeks ago 33 Commits

WEEK1	전체 업로드	2 months ago
WEEK2	add&delete	2 months ago
WEEK3	update	2 months ago
WEEK4	final update	2 months ago
WEEK5	refactor: 레이아웃 수정	last month
WEEK6	update: 내용 업데이트	last month
WEEK7	update: week7 ppt update	2 weeks ago
.gitignore	전체 업로드	2 months ago

Activity 0 stars 0 watching 0 forks

No description, website, or topics provided.

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

HTML 45.9%	JavaScript 30.9%
Java 13.9%	Python 5.1%
CSS 4.1%	Haskell 0.1%

README

Add a README

<https://github.com/Igw323/ecolecapstonedesign>