

포팅매뉴얼



re:code 프로젝트의 Gitlab 소스를 클론한 이후
빌드 및 배포할 수 있도록 정리한 문서.

환경 설정

1. 개발 환경

A. Frontend (수정 필요)

- **Node.js** : ^20.11.1
- **react**: ^18.2.0
- **npm** : ^10.2.4
- **axios**: ^1.6.8
- **npm** : ^10.2.4
- **firebase** : ^10.11.1
- **million**: ^3.0.6
- **recharts**: ^2.12.6
- **zustand**: ^4.5.2
- **eslint**: ^8.57.0
- **msw**: ^2.2.13
- **prettier**: ^3.2.5
- **tailwindcss**: ^3.4.3
- **typescript**: ^5.2.2
- **vite**: ^5.2.0
- **@tanstack/react-query**: ^5.32.0

B. Backend

- **Server** : 20.04.6 LTS
- **JDK** : 21.0.2
- **Spring Boot** : 3.2.3
- **QueryDSL** : 5.0.0
- **DB**
 - **MySQL** : 8.0.36
 - **Redis** : 5.0.7

C. CI/CD

- **AWS EC2**
- **Docker** : 26.1.0
- **Jenkins** : 2.454

D. 협업 툴

- **GitLab** : 형상관리
- **Jira** : 이슈관리
- **Mattermost, notion** : 소통
- **Figma** : 디자인

2. 설정 파일 및 환경 변수

▼ .env

위치 `jenkins_credentials`

Dockerfile 실행에 필요한 환경변수.

```
# 'development' 'server'
VITE_NODE_ENV='server'
VITE_BASE_URL=https://www.recode-d210.com/api

VITE_FIREBASE_API_KEY=AIzaSyCD-dm8gYbrvTkpqzitKKYbjpyV5EuPno4
VITE_FIREBASE_AUTH_DOMAIN=recode-43ead.firebaseio.com
VITE_FIREBASE_PROJECT_ID=recode-43ead
VITE_FIREBASE_STORAGE_BUCKET=recode-43ead.appspot.com
VITE_FIREBASE_MESSAGING_SENDER_ID=19941674840
VITE_FIREBASE_APP_ID=1:19941674840:web:43538b7853c17f12aa9f2d
VITE_FIREBASE_MEASUREMENT_ID=G-7XSJSFS2ET
VITE_FIREBASE_VAPID_KEY=BI2mS0DCnZzielXaVmImIRF0SqWsYAep2Uy5eiGuBta9bHhNpj0tx7i-pDFUciwi1
OpbxEYtLoake8AC-iPZDc
```

▼ application.yml

위치 `./back/src/main/resources/application.yml`

```
server:
  servlet:
    context-path: /api

spring:
  profiles:
    include:
      - secret
  jpa:
    database-platform: org.hibernate.dialect.MySQL8Dialect
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        default_batch_fetch_size: 1000
        format_sql: true
        rewriteBatchedStatements: true
      jdbc:
        time_zone: Asia/Seoul
        order_updates: true
        order_inserts: true
    show-sql: true
    defer-datasource-initialization: true

    open-in-view: false
```

```

security:
  oauth2:
    client:
      registration:
        github:
          client-name: github
          client-id: ${oauth.client.id}
          client-secret: ${oauth.client.secret}
          redirect-uri: ${oauth.redirect.uri}
          authorization-grant-type: authorization_code
#          scope: no-scope

datasource:
#   url: jdbc:mysql://${MYSQL_HOST:localhost}:3306/recode?serverTimezone=UTC&useUnicode=yes&characterEncoding=UTF-8
   url: jdbc:mysql://${db.server_public_ip}:3306/recode?serverTimezone=UTC&useUnicode=yes&characterEncoding=UTF-8

   username: ${db.username}
   password: ${db.password}
   driver-class-name: com.mysql.cj.jdbc.Driver
   hikari:
     connection-timeout: 30000 # 커넥션 타임아웃 설정 (ms)
     maximum-pool-size: 10 # 커넥션 풀의 최대 크기 설정
     minimum-idle: 5 # 유휴 커넥션을 유지할 최소한의 커넥션 수 설정
     idle-timeout: 600000 # 유휴 커넥션의 최대 유지 시간 설정 (ms)
     pool-name: MyHikariCP # 커넥션 풀의 이름 설정
     validation-timeout: 5000 # 커넥션 유효성 검사 타임아웃 설정 (ms)
     auto-commit: true # 자동 커밋 모드 설정

redis:
  host: ${db.redis.host}
  port: ${db.redis.port}
  password: ${db.redis.password}

openai:
  secret-key: ${openai.secret-key}

springdoc: #swagger
  packages-to-scan: com.clover.recode
  default-consumes-media-type: application/json;charset=UTF-8
  default-produces-media-type: application/json;charset=UTF-8
  swagger-ui:
    path: /record.html # Swagger UI ?? => localhost:8080/api/record.html
    tags-sorter: alpha # alpha: ??? ? ?? ??, method: HTTP Method ? ??
    operations-sorter: alpha # alpha: ??? ? ?? ??, method: HTTP Method ? ??
  api-docs:
    path: /api-docs/json
    groups:
      enabled: true
  cache:
    disabled: true

firebase:
  config-path: ${FIREBASE_CONFIG_PATH}

```

```
management:
  endpoints:
    web:
      exposure:
        include: prometheus, health, info
  metrics:
    tags:
      application: ${spring.application.name}
```

▼ application-key.yml

위치 `./back/src/main/resources/application-secret.yml`

```
db:
  server_public_ip: 13.209.10.226
  username: root
  password: fpzhemWkdWkd777!
  redis:
    host: www.recode-d210.com
    port: 6379
    password: fpzhemWkdWkd777!

jwt:
  secret: SSAYFekemfwkfehoTdmauswhgrpTekgodqhrqksvmfhwprxmrlrksdlehlrlf20240425

oauth:
  client:
    id: 8f3a7cf3f8cc9caafecb
    secret: a6490817cdd56cf2dc53f2115bba756d2fa1f5ef
  redirect:
    uri: https://www.recode-d210.com/api/login/oauth2/code/github

openai:
  secret-key: sk-proj-XTpvJRswMb304AWMr0PaT3BlbkFJLCdFZ8F35D9gP0Kkj9Vd

firebase:
  config-path: firebase/recode-43ead-firebase-adminsdk-vhyqn-37b7e61941.json
```

빌드 및 배포

1. 준비하기

Jenkins - Pipeline script

```
node {
  try {
    stage('Clone') {
      git branch: 'develop', credentialsId: 'gitlab', url: 'https://lab.ssafy.com/s10-final/S10P31D210.git'

      dir('back/src/main/resources') {
        withCredentials([file(credentialsId: 'backend_secret', variable: 'SECRETS_FILE')]) {
        }
```

```

        // SECRETS_FILE 변수를 사용하여 파일을 프로젝트 디렉토리로 복사
        sh 'chmod -R 755 .'
        sh 'cp $SECRETS_FILE .'
    }
    withCredentials([file(credentialsId: 'firebase_key', variable: 'SECRETS_FILE')
    ]) {
        // SECRETS_FILE 변수를 사용하여 파일을 프로젝트 디렉토리로 복사
        sh 'chmod -R 755 .'
        sh 'mkdir -p firebase'
        sh 'cp $SECRETS_FILE ./firebase'
    }
}

dir('frontend') {
    withCredentials([file(credentialsId: 'frontend_secret', variable: 'SECRETS_FILE')
    ]) {
        // SECRETS_FILE 변수를 사용하여 파일을 프로젝트 디렉토리로 복사
        sh 'chmod -R 755 .'
        sh 'cp $SECRETS_FILE .'
    }
}

stage('Build') {
    dir('back') {
        sh 'chmod +x gradlew'
        sh './gradlew clean build'
    }

    dir('frontend') {
        sh 'npm i'
        sh 'npm run build'
    }
}

stage('Deploy') {
    sh 'docker-compose down'
    sh 'docker rmi -f recode_back_image:0.0'
    sh 'docker rmi -f recode_front_image:0.0'
    sh 'docker-compose up -d'
}

// 성공한 경우의 후처리
mattermostSend(color: 'good',
                message: "빌드 성공: ${env.JOB_NAME} #${env.BUILD_NUMBER} by ${env.gitlabUserName ?: '제킨스'}\n(<${env.BUILD_URL}|Details>)",
                endpoint: 'https://meeting.ssafy.com/hooks/rdbj4sqy63ygpejut6bzmy9so
w',
                channel: 'd210-jenkins')
} catch (Exception e) {
    // 실패한 경우의 후처리
    mattermostSend(color: 'danger',
                    message: "빌드 실패: ${env.JOB_NAME} #${env.BUILD_NUMBER} by ${env.gitlabUserName ?: '제킨스'}\n(<${env.BUILD_URL}|Details>)",
                    endpoint: 'https://meeting.ssafy.com/hooks/rdbj4sqy63ygpejut6bzmy9so
w',

```

```

        channel: 'd210-jenkins')
    throw e // 예외를 다시 던져서 빌드 실패 상태를 유지
}
}

```

Nginx

위치(서버) : `/home/ubuntu/nginx/default.conf`

```

upstream app {
    server recode-back:8080; # WAS 컨테이너의 이름
}

server {
    listen 80;
    listen 443 ssl;
    server_name k10d210.p.ssafy.io;

    root /usr/share/nginx/html; # 정적 파일이 위치한 디렉토리 지정

    location / {
        try_files $uri $uri/ /index.html; # 정적 파일 제공
    }

    location /api {
        proxy_pass http://app; # API 요청을 WAS로 전달
    }

    # k10d210.p.ssafy.io을 위한 SSL 설정
    ssl_certificate /etc/letsencrypt/live/k10d210.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/k10d210.p.ssafy.io/privkey.pem;
}

server {
    listen 80;
    listen 443 ssl;
    server_name www.recode-d210.com;

    root /usr/share/nginx/html; # 정적 파일이 위치한 디렉토리 지정

    location / {
        try_files $uri $uri/ /index.html; # 정적 파일 제공
    }

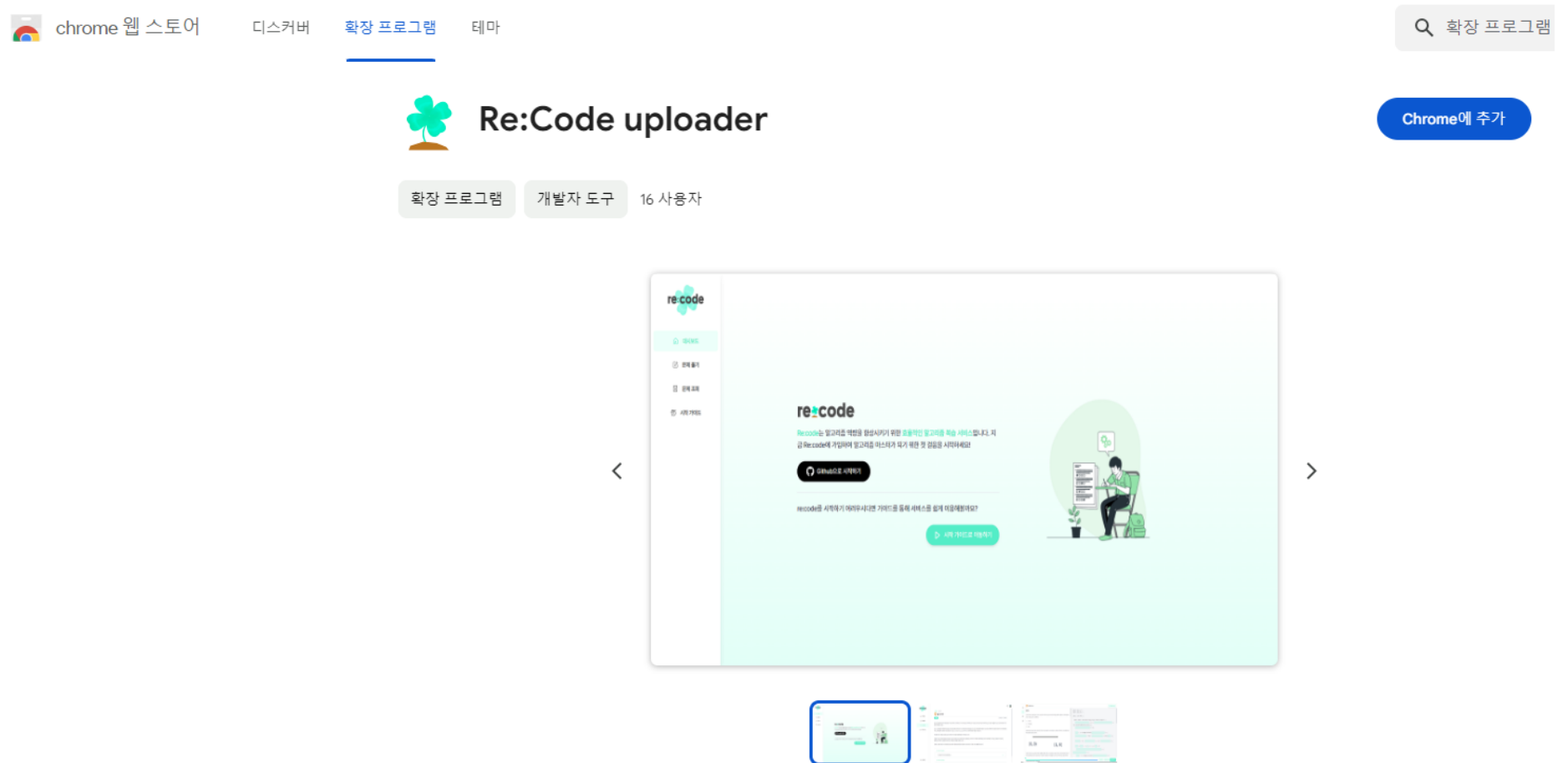
    location /api {
        proxy_pass http://app; # API 요청을 WAS로 전달
    }

    # www.recode-d210.com을 위한 SSL 설정
    ssl_certificate /etc/letsencrypt/live/www.recode-d210.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/www.recode-d210.com/privkey.pem;
}

```

서비스 이용

1. 레코드 확장 프로그램 설치



2. 레코드 서비스와 확장 프로그램 연동

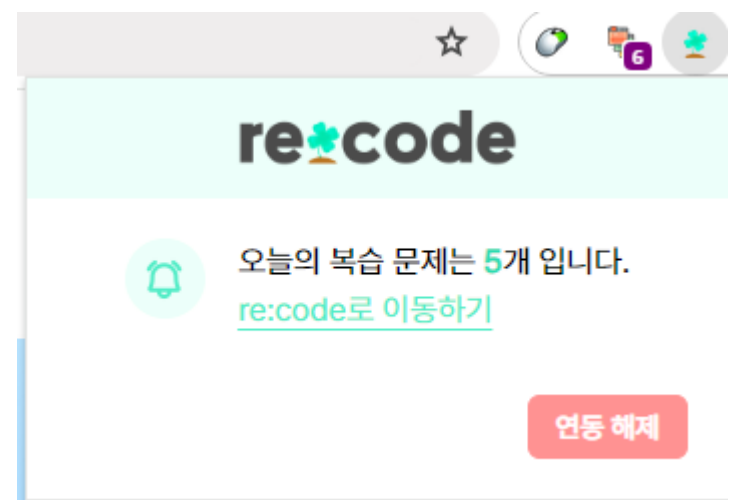
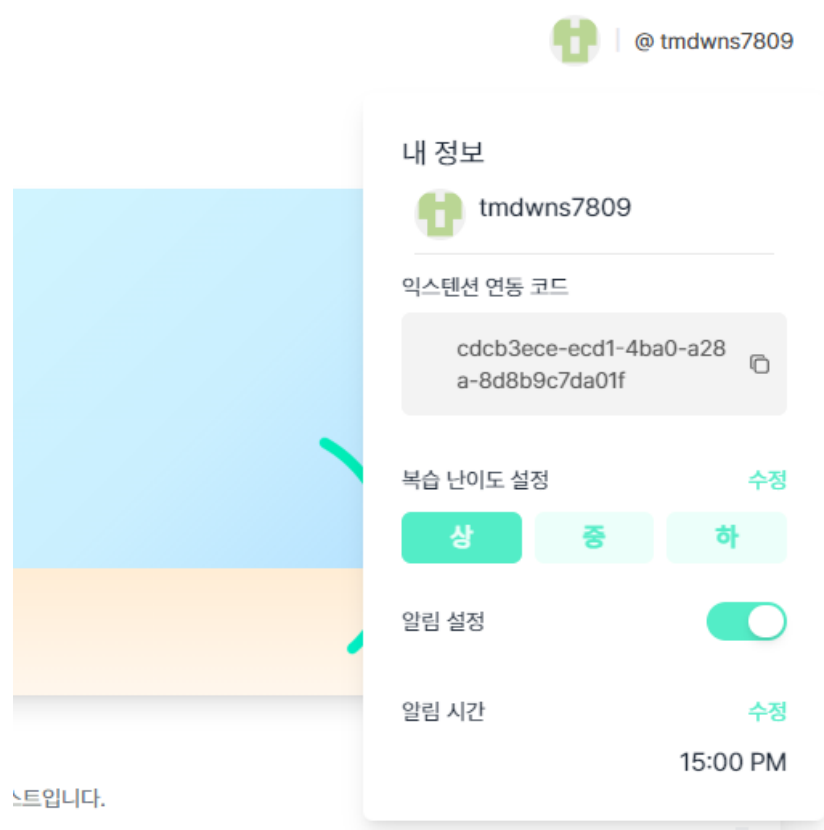
1. 확장 프로그램 클릭 후 서비스 페이지로 이동

2. 깃헙 로그인 하기



3. 오른쪽 상단 내 정보 클릭후 연동코드 복사하기

4. 확장 프로그램에서 연동 코드 입력하여 연동하기



3. 복습하고 싶은 코드 등록하기

1. 백준 사이트에서 맞힌 코드 옆에 생성된 RE:CODE 버튼 클릭

BAE<JOON>
ONLINE JUDGE

tmdwns7809 | 설정 | 로그아웃 |

문제 | 문제집 | 대회 3 | 채점 현황 | 랭킹 | 게시판 | 그룹 | 더 보기

[NEW] 코딩 테스트 준비 온라인 강의 할인 중 / 코딩 테스트 및 인터뷰 온라인 강의

문제

tmdwns7809

모든 언어

맞았습니다!!

검색

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간	
78368793	tmdwns7809	5 18428	맞았습니다!!	RE:CODE	11416 KB	200 ms	node.js / 수정	1471 B	1일 전
67883447	tmdwns7809	4 17069	맞았습니다!!	RE:CODE	14476 KB	140 ms	Java 11 / 수정	1266 B	7달 전
67769596	tmdwns7809	3 30054	맞았습니다!!	RE:CODE	46092 KB	600 ms	Java 11 / 수정	1952 B	7달 전
67579067	tmdwns7809	2 2515	맞았습니다!!	RE:CODE	142332 KB	1428 ms	Java 11 / 수정	1391 B	7달 전
67251090	tmdwns7809	4 3055	맞았습니다!!	RE:CODE	14304 KB	124 ms	Java 11 / 수정	1348 B	7달 전
67215717	tmdwns7809	4 2239	맞았습니다!!	RE:CODE	17780 KB	404 ms	Java 11 / 수정	1505 B	7달 전
67135691	tmdwns7809	4 14502	맞았습니다!!	RE:CODE	276572 KB	588 ms	Java 11 / 수정	2137 B	7달 전
65827683	tmdwns7809	3 11048	맞았습니다!!	RE:CODE	79496 KB	536 ms	Java 11 / 수정	961 B	8달 전
65786211	tmdwns7809	1 1149	맞았습니다!!	RE:CODE	14532 KB	144 ms	Java 11 / 수정	938 B	8달 전
65568597	tmdwns7809	4 17281	맞았습니다!!	RE:CODE	63204 KB	568 ms	Java 11 / 수정	1911 B	8달 전
65566037	tmdwns7809	4 17281	맞았습니다!!	RE:CODE	63140 KB	588 ms	Java 11 / 수정	5921 B	8달 전
65553532	tmdwns7809	4 15961	맞았습니다!!	RE:CODE	274100 KB	680 ms	Java 11 / 수정	1272 B	8달 전
65444412	tmdwns7809	5 1759	맞았습니다!!	RE:CODE	15732 KB	140 ms	Java 11 / 수정	1165 B	8달 전

2. 등록할 코드의 제목 설정 후 등록

문제	tmdwns7809	모든 언어	맞았습니다!!	검색	
문제	결과	메모리	시간	언어	코드
5 18428	맞았습니다!!	RE:CODE 11416 KB	200 ms	node.js / 수정	147
4 17069	맞았습니다!!	RE:CODE 14476 KB	140 ms	Java 11 / 수정	120
3 30054	맞았습니다!!		600 ms	Java 11 / 수정	199
2 2515	맞았습니다!!		1428 ms	Java 11 / 수정	139
4 3055	맞았습니다!!		124 ms	Java 11 / 수정	134
4 2239	맞았습니다!!		404 ms	Java 11 / 수정	150
4 14502	맞았습니다!!		588 ms	Java 11 / 수정	213
2 11048	맞았습니다!!	RE:CODE 79496 KB	536 ms	Java 11 / 수정	96
1 1149	맞았습니다!!	RE:CODE 14532 KB	144 ms	Java 11 / 수정	938
4 17281	맞았습니다!!	RE:CODE 63204 KB	568 ms	Java 11 / 수정	19
4 17281	맞았습니다!!	RE:CODE 63140 KB	588 ms	Java 11 / 수정	59
4 15961	맞았습니다!!	RE:CODE 274100 KB	680 ms	Java 11 / 수정	12

RE:CODE

코드 제목 입력 (최대 20자)

BFS|

닫기
업로드

4. 메인 대시보드

🏠 대시보드
📝 문제 풀기
📁 문제 조회
📖 시작 가이드
🔗 익스텐션

👤 로그아웃

tmdwns7809님 복습 연속 0일째예요!

re:code에서 함께 알고리즘 복습 해볼까요?

@ tmdwns7809

출석 체크
지난 일주일 간의 학습 여부입니다

월
화
수
목
금
토
일

주간 복습량
지난 일주일 간의 복습 횟수입니다

현재 상위 100%

월화수목금토일

오늘의 보충 문제

BOJ 1647 도시 분할 계획

오늘의 연습 문제

BOJ 11066 파일 합치기

오늘의 복습 리스트
지금까지 쌓인 복습 리스트입니다.

BOJ 17069 파이프 옮기기 2 1번째 복습

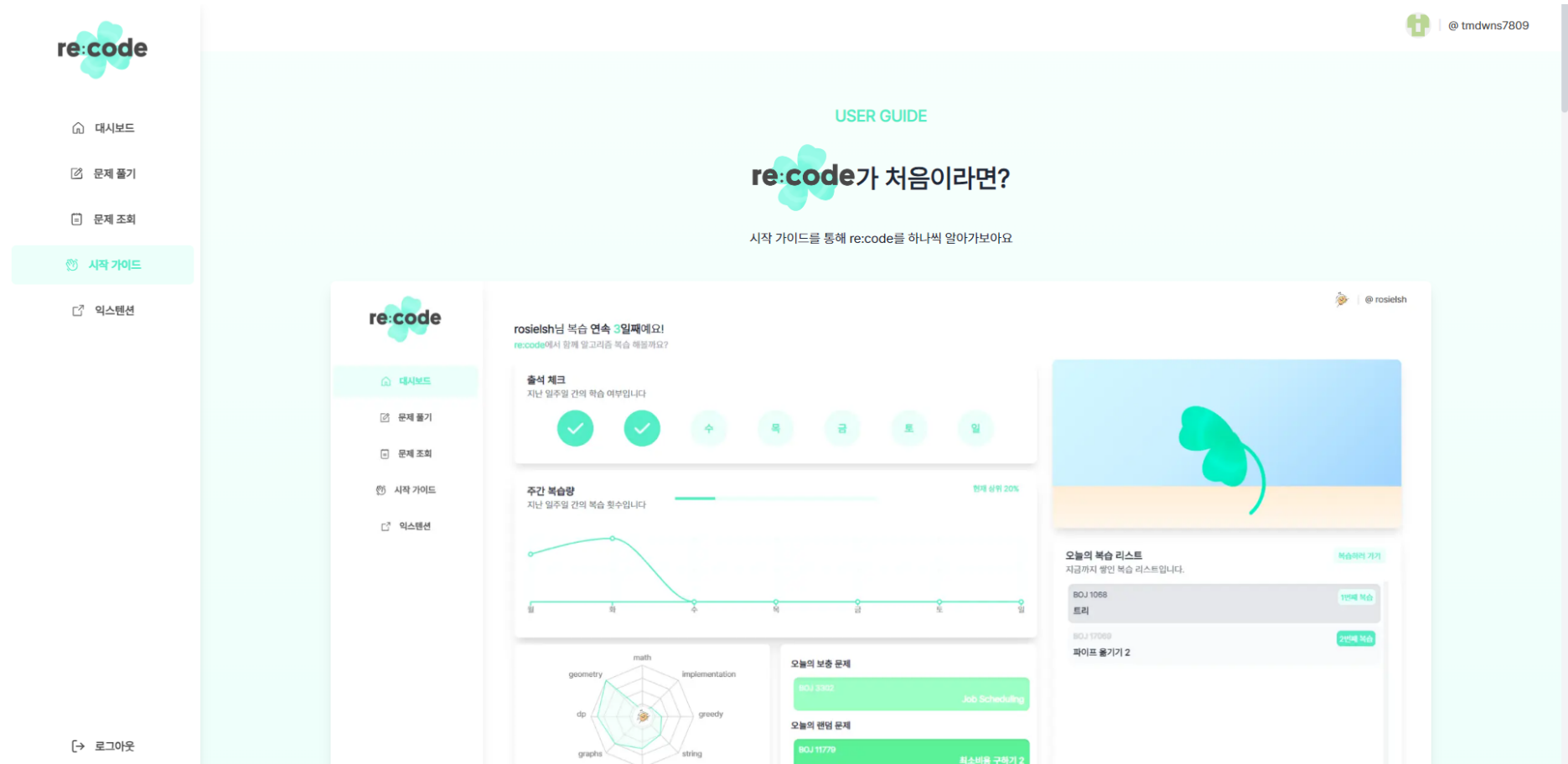
BOJ 15683 감시 1번째 복습

BOJ 2178 미로 탐색 1번째 복습

BOJ 3040 백설 공주와 일곱 난쟁이 1번째 복습

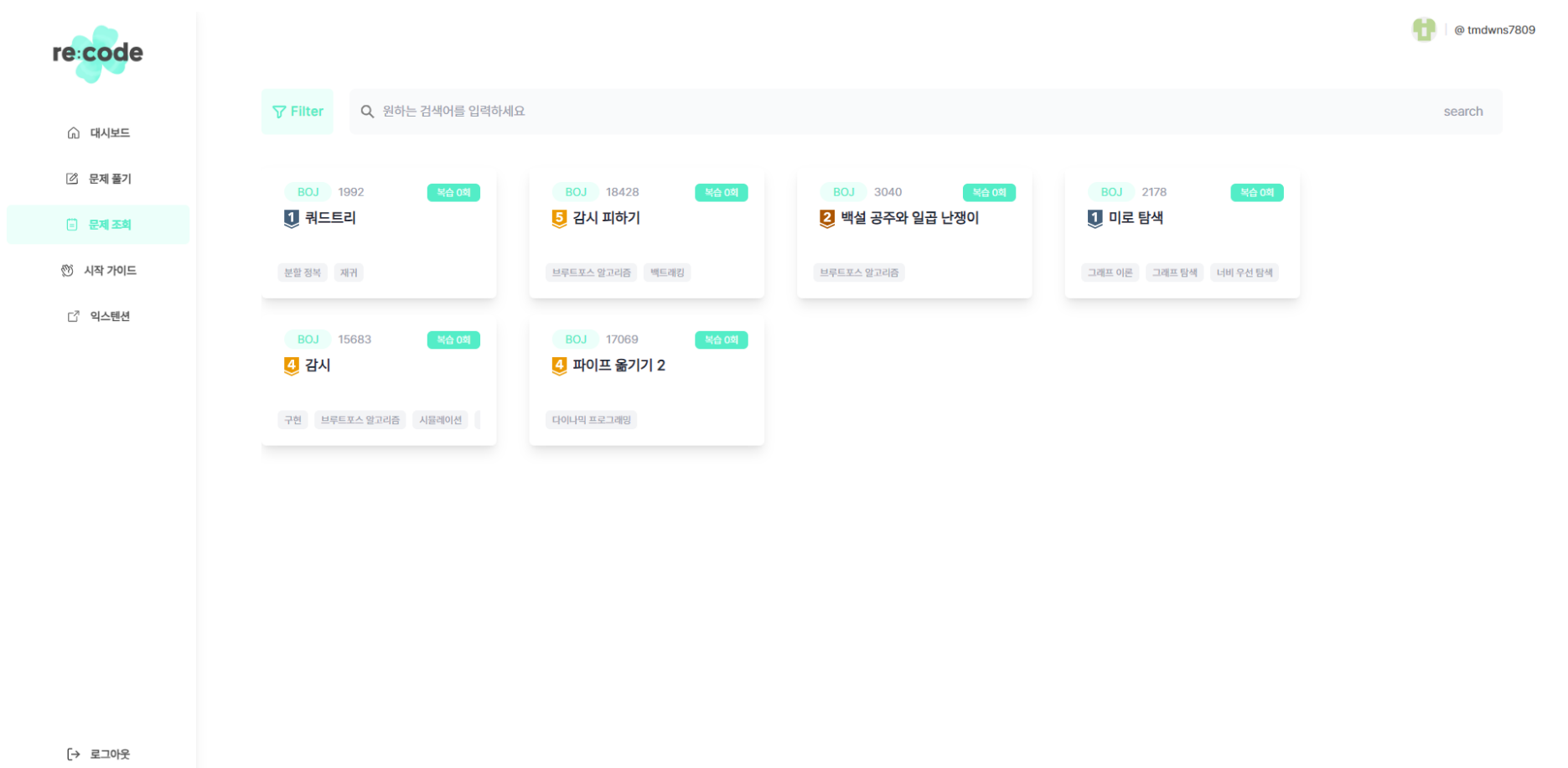
BOJ 18428 감시 피하기 1번째 복습

5. 시작 가이드



6. 문제 조회

1. 문제 전체 조회 및 검색



2. 문제 상세 조회

BOJ 17069

4

파이프 옮기기 2

복습 0회

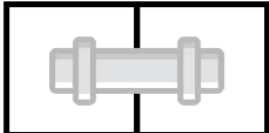
</>

다이나믹 프로그래밍

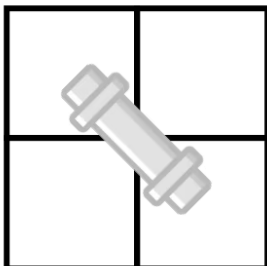
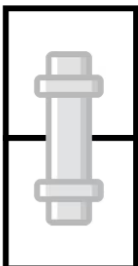
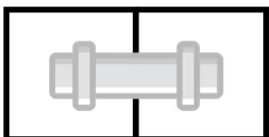
문제

유현이가 새 집으로 이사했다. 새 집의 크기는 $N \times N$ 의 격자판으로 나타낼 수 있고, 1×1 크기의 정사각형 칸으로 나누어져 있다. 각각의 칸은 (r, c) 로 나타낼 수 있다. 여기서 r 은 행의 번호, c 는 열의 번호이고, 행과 열의 번호는 1부터 시작한다. 각각의 칸은 빈 칸이거나 벽이다.

오늘은 집 수리를 위해서 파이프 하나를 옮기려고 한다. 파이프는 아래와 같은 형태이고, 2개의 연속된 칸을 차지하는 크기이다.



파이프는 회전시킬 수 있으며, 아래와 같이 3가지 방향이 가능하다.



3. 등록된 코드 확인

가장 처음에 파이프는 $(1, 1)$ 와 $(1, 2)$ 를 차지하고 있고, 방향은 가로이다. 파이프의 한쪽 끝을 (N, N) 로 이동시키는 방법의 개수를 구해보자.

입력

첫째 줄에 집의 크기 N ($3 \leq N \leq 32$)이 주어진다. 둘째 줄부터 N 개의 줄에는 집의 상태가 주어진다. 빈 칸은 0, 벽은 1로 주어진다. $(1, 1)$ 과 $(1, 2)$ 는 항상 빈 칸이다.

출력

첫째 줄에 파이프의 한쪽 끝을 (N, N) 으로 이동시키는 방법의 수를 출력한다. 이동시킬 수 없는 경우에는 0을 출력한다.

2024-05-13 제출

빈 제목입니다. ✎

☰ 리스트에 추가하기

^

제출 정답 기록

리마인드 ON



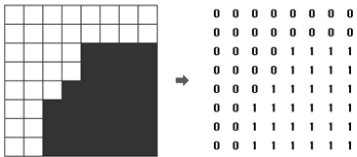
```
1 import java.io.BufferedReader;
2 import java.io.InputStreamReader;
3 import java.util.StringTokenizer;
4
5 // 17069 파이프 옮기기 2
6 public class Main {
7
8     public static void main(String[] args) throws Exception {
9
10        // 입력 객체 생성
11        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
12
13        // 크기 N 읽기
14        int N = Integer.parseInt(br.readLine());
15
16        // 맵 객체 생성
```

7. 문제 풀기

문제

흑백 영상을 압축하여 표현하는 데이터 구조로 퀴드 트리(Quad Tree)라는 방법이 있다. 흰 점을 나타내는 0과 검은 점을 나타내는 1로만 이루어진 영상(2차원 배열)에서 같은 숫자의 점들이 한 곳에 많이 몰려있으면, 퀴드 트리에서는 이를 압축하여 간단히 표현할 수 있다.

주어진 영상이 모두 0으로만 되어 있으면 압축 결과는 "0"이 되고, 모두 1로만 되어 있으면 압축 결과는 "1"이 된다. 만약 0과 1이 섞여 있으면 전체를 한 번에 나타내지를 못하고, 왼쪽 위, 오른쪽 위, 왼쪽 아래, 오른쪽 아래, 이렇게 4개의 영상으로 나누어 압축하게 되며, 이 4개의 영역을 압축한 결과를 차례대로 괄호 안에 묶어서 표현한다



위 그림에서 왼쪽의 영상은 오른쪽의 배열과 같이 숫자로 주어지며, 이 영상을 퀴드 트리 구조를 이용하여 압축하면 "`0(0011)(0111)01)1`"로 표현된다. $N \times N$ 크기의 영상이 주어질 때, 이 영상을 압축한 결과를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에는 영상의 크기를 나타내는 숫자 N 이 주어진다. N 은 언제나 2의 제곱수로 주어지며, $1 \leq N \leq 64$ 의 범위를 가진다. 두 번째 줄부터는 길이 N 의 문자열이 N 개 들어온다. 각 문자열은 0 또는 1의 숫자로 이루어져 있으며, 영상의 각 점들을 나타낸다.

출력

영상을 압축한 결과를 출력한다.



오늘 복습 문제 6문제 중 0문제 완료

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    static int N;

    static char[][] map;

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

        N = Integer.parseInt(br.readLine());

        map = new char[N][N];

        for (int i = 0; i < N; i++)
            map[i] = br.readLine().toCharArray();

        StringBuilder sb = new StringBuilder();

        (16);

        System.out.println(sb);
    }

    static void dfs(StringBuilder sb, int r, int c, int size) {
        boolean full = true;
        mainloop: for (int i = r; i < r+size; i++) {
            for (int j = c; j < c+size; j++) {
                if ( (22) ) {
                    full = false;
                    break mainloop;
                }
            }
        }
    }
}
```

건너뛰기

빈칸 초기화

복습 완료하기