질문용 <https://app.sli.do/event/wUjWoPefTFpV5YPq1Fu2TZ/live/questions>

온라인 플랫폼 repliit

Icpc 대회?

백준, 프로그래머스 (삼성, 카카오)

구현, dfs/bfs 를 활용한 탐색, 그리디 가 많이 나옴

Dfs/bfs 위주 탐색문제 공부, 구현 난이도 갈수록 높아짐, 기출문제 많이 돌리기

1. 문법공부
2. 코드업에서 200문제 풀기
3. 알고리즘 이론과 핵심 문제 풀기
4. 백준 온라인 저지에서 유형별 문제 20개이상
5. 백준 온라인 저지에서 삼성 sw 역량테스트 풀기
6. 프로그래머스 카카오 문제 풀기
7. 알고리즘 복습 하고 골드1~5 문제 마스터

난이도 거의 골드2~5

카카오가 가장 어려움

문제 풀이를 위한 의식의 흐름

1. 문제의 조건 : n 범위, 시간제한
2. 조건을 통해 시간 복잡도 유추
3. 복잡도에 만족하는 알고리즘 아이디어(그리디) 생각
4. 그래프 형태를 보고 완전 탐색 생각
5. 경우의 수가 너무 많으면 다이나믹 프로그래밍으로 접근

복잡도 – 시간복잡도, 공간복잡도

Nlogn이 가장 많이 나옴

수행시간은 데이터의 개수에 비례할 것임을 예측할 수 있음. (데이터의 합을 할 때)

코드 내부적으로 다른 함수를 호출한다면 그 함수의 복잡도까지 고려

연산횟수가 5억을 넘어가면 c언어에서 1~3초 소요, 코테 시간제한은 대략 1~5초 정도

문제를 풀때는 가장 먼저 시간제한을 봐야한다.

1초일 때

N 범위가

500일 때 n3제곱

2000일 때 n제곱

100000경우 nlogn

10000000일 때 n

Log는 생각보다 매우 빠른 알고리즘 (n의 승이기 때문에)

문제해결과정

1. 지문일기 및 컴퓨터적 사고
2. 요구사항 복잡도 분석
3. 기본 입출력 문제

데이터를 가지고 정렬하는 문제

자바 입출력 할 때 FAST READER 템플릿 사용

나동빈씨 깃허브 (참고하면 좋음) : <https://github.com/ndb796/BOJ_Java>



“나 \는 인식이 안되므로 앞에 \을 붙인다

코테시 언어는 c나 파이썬 자바 많이 사용 보통 처음에 c가 빨리풀기 좋은데 카카오 같은 경우 파이썬이 매우 유용

실제 테스트에서도 실시간으로 피드백을 받을 수 있는가? 횟수 제한이 있는가? X, 제한이 있을 수도 있고, 없을 수도 있음

회사 검색 꿀팁 : 재택근무 가능한 it기업회사 깃허브 (구글에 검색)

딥러닝 엔지니어링(모델 사용) : 대학원 x , 딥러닝 리서치(모델 만들기) : 대학원

대학원 없이 할 수 있는 후자 직군 ? 엔지니어링 쪽에서는 대학원 없이 뽑지만, 여러 모델을 가지고 사용한 경험이나 네이버 ai 부트캠프와 같이 하시고 넘어갈 수 있음. 혹은 케글 입상도 많이 쳐줌.

깃허브에 교육용이나 로드맵같이 해서 올리면 star 많이 받는 것도 괜찮음.

(했던 것들도 올려도 ㄱㅊ)

Sw 마에스트로,, 42서울, saffy 다 해도 괜찮음, bob (보안쪽)

남들 하는거 말고 나만의 뭔가 하나 만들기

포트폴리오 노션 괜찮음, 깃이랑 노션 둘다 사용해도 좋음

코테 준비 기간 : 길어도 3개월정도 잡음. 강의 끝나고 50시간 정도 (하루에 어느정도씩)

코테 풀 때 한 시간 이상 고민하지 마라. (안 풀리는 거) – 솔루션 보고 다시 공부

백준 2588번

import java.util.\*;

import java.io.\*;

public class Main {

public static void main(String[] args) {

FastReader fr = new FastReader();

int a = fr.nextInt(); //3자리수 입력

int b = fr.nextInt();

int c1 = a\* (b%10); //1의 자리는 10 나눠서 나머지

int c2 = a \* ((b%100)/10); //십의 자리 100나눠서 10으로 나눈 몫

int c3 = a \* (b/100);

int addd = a\*b;

System.out.println(c1);

System.out.println(c2);

System.out.println(c3);

System.out.println(addd);

}

public static class FastReader {

BufferedReader br;

StringTokenizer st;

public FastReader() { br = new BufferedReader(new InputStreamReader(System.in)); }

public FastReader(String s) throws FileNotFoundException { br = new BufferedReader(new FileReader(new File(s))); }

String next() {

while (st == null || !st.hasMoreElements()) {

try { st = new StringTokenizer(br.readLine()); }

catch (IOException e) { e.printStackTrace(); }

}

return st.nextToken();

}

int nextInt() { return Integer.parseInt(next()); }

long nextLong() { return Long.parseLong(next()); }

double nextDouble() { return Double.parseDouble(next()); }

String nextLine() {

String str = "";

try { str = br.readLine(); }

catch (IOException e) { e.printStackTrace(); }

return str;

}

}

}

보통 코테에서 자릿수 나눌때는 string으로 입력받아서 한자리씩 떼어서 받음

백준 <https://www.acmicpc.net/step> 수업중에 안 한 거 다 풀기

스프링, 리액트, vue 3개중에 하나만 해도 나머지는 회사에서 배우지만, 리액트가 우세

백준 2884

FastReader fr = new FastReader(); // 문제 해결 소스 코드

int hour = fr.nextInt();

int minute = fr.nextInt();

minute -= 45; // 45분 앞당기기

if (minute < 0) { // 분이 음수가 된 경우

minute += 60;

hour -= 1;

if (hour < 0) {

hour += 24;

}

}

System.out.println(hour + " " + minute);

백엔드 할 때 프론트 공부? 백엔드만 할려면 프론트에서 api 호출만 필요, (create, research, update, delete = CRUD) 를 만들고 호출만 할 수 있을 정도면 충분하지만, 좀 작은 기업은 프론트도 좀 더 공부해야함( 풀스택 요구하는 회사) 대기업에서 백엔드는 프론트 부분 요구하지는 않음

캐클, 데이콘에서의 수상은 인공지능쪽에서 좋긴함. 6개월정도는 공부 필요, 대회만을 위한 목적을 공부해야함, 앙상블 같이

해커톤 나가면 좋음. 많이 많이

공모전 사이트 검색해서 IT쪽 많이 나가면 좋음 (사람들이 많이 관심 안 가지는 것도)

백엔드라도 해커톤 나가면 좋음

C의 입출력이 C++ 보다 빨라서 쓰는 경우 많음. C++는 STL때문에 쓰는 경우가 많음. C, C++를 섞어 쓴다. 문제풀때는 SCANF가 빠름 (속도만 보자면)

맥북 많이 씀. 그냥 회사 로컬 컴퓨터에 따라서 씀. 프론트는 자기 디바이스 사용하는 경우가 많은데 나머지는 상관 X.

컨퍼런스를 통해서 가고싶은 분야 잡기

대기업들은 아예 블라인드이기에 코테랑 면접만 봄. (AWS나 클라우드 같은 자격증은 쳐줄때가 있음.) , 기사는 필요. 그냥 코테랑 면접만 준비. 영어는 그냥 별로… 입사 후 준비. 프로젝트는 최소 2~3개 정도. 프로젝트는 흔한 거 말고 쓰일법한 거 만들어보는게 중요. 대외 활동은 많이 할 수록 좋음. (특히나 대회, 포폴)

면접준비는 CS와 포폴. CS는 대체로 복잡도나 WORST CASE에 대한 그런거, OS의 처리 같은 거도 나오는데 포폴에 대한 준비가 더 중요. 이력서나 포폴에 대한 질문 준비가 가장 중요. 거의 반반 물어봄.

자바 쓸 때 ARRAYLIST 사용 (Collections.min,max 많이 사용)

백준 10818

FastReader fr = new FastReader();

int a = fr.nextInt();

ArrayList<Integer> arrayList = new ArrayList<>();

for(int i =0;i <a; i++){

int x = fr.nextInt();

arrayList.add(x);

}

int min = Collections.min(arrayList);

int max = Collections.max(arrayList);

System.out.println(min+" "+max);

}

자바에서 각 자릿수 비교할 때 수가 작으면 문자열로 바꿔서 떼어냄

백준 2577

FastReader fr = new FastReader();

int[] cnt = new int[10];

int a = fr.nextInt();

int b = fr.nextInt();

int c = fr.nextInt();

int d = a\*b\*c;

String tmp = Integer.toString(d);

for(int i =0; i< tmp.length(); i++){

int a1 = Integer.parseInt(tmp.charAt(i)+"");

cnt[a1]+= 1;

}

for(int i = 0; i<10;i++){

System.out.println(cnt[i]);

}

소수점 아래 몇자리 = string.format(“%.몇자리f”, ~)

5단계까지 다 풀기