

ICPC 준비의 첫걸음

숭실대학교 컴퓨터학부 나정휘

jeoungh.nah@gmail.com

<https://justicehui.github.io/etc/2023/05/30/icpc-preparation/>

목차

- ICPC 대회 소개
- 필요한 능력과 훈련 방법
- 최근 기출 문제 분석
- 학교 전공과목과의 연계
- 대회 전략과 팀워크

ICPC 대회 소개

프로그래밍 경시대회

- 주어진 시간 동안 주어진 문제를 최대한 많이 푸는 형식의 대회
 - 개인 대회는 4~5시간 3~4문제
 - 3인 1팀 대회는 5시간 12문제
 - 실시간으로 모든 참가자의 점수를 알 수 있는 대회

UCPC 2022 본선 (Frozen)

11	1 st	초비상!! 코딩 말했어요, 시간초과 땀어요, Ofast가 해줘야 해요		A	B	C	D	E	F	G	H	I	J	K	L	M	1390 min.
	2 nd	우리가 우승할 수 있을 리 없잖아, 무리무리! (U+203B, 무리가, 아니었다?)		A	B	C	D	E	F	G	H	I	J	K	L	M	1565 min.
10	3 rd	BabyPenguin retro3014, gs18115, moonrabbit2		A	B	C	D	E	F	G	H	I	J	K	L	M	1277 min.
	4 th	실러캔스, 암모나이트, 삼엽충		A	B	C	D	E	F	G	H	I	J	K	L	M	1372 min.
9	5 th	UCPC의 최신 동향 나폴리폴리, 고슬고슬비빈, 탕맛기픈		A	B	C	D	E	F	G	H	I	J	K	L	M	985 min.
	6 th	DM 확인 부탁드립니다! 🙏 안녕하세요~, 코드 관련 질문 드려도 될까요?, 오늘 중으로 시간 되시나요?		A	B	C	D	E	F	G	H	I	J	K	L	M	1593 min.
8	7 th	aHR0cDovL2Vyci5vLXlua3lv 7j207Y647KeA64qU7jiB6rWt, 7jeQ715c7Iuc7j6R65CY7Ja0, 64Ka8Iuc656N64uz64pk7ZWY		A	B	C	D	E	F	G	H	I	J	K	L	M	1074 min.
	8 th	Rinjinbu_SSHS EaglesV7, greg3566, ravor8878		A	B	C	D	E	F	G	H	I	J	K	L	M	1108 min.
	9 th	예일대 예선, 일등, 딱대		A	B	C	D	E	F	G	H	I	J	K	L	M	1199 min.
7	10 th	NULL None, nullptr, NIL		A	B	C	D	E	F	G	H	I	J	K	L	M	802 min.
	11 th	kopagi kohandy, pasa3232, gina		A	B	C	D	E	F	G	H	I	J	K	L	M	813 min.
	12 th	송실사이버의대를다니고나의성공시대시작됐다 모, 새, 기		A	B	C	D	E	F	G	H	I	J	K	L	M	844 min.
	13 th	새내기과 퀘리 maruii, azber, etyu		A	B	C	D	E	F	G	H	I	J	K	L	M	861 min.
	14 th	OReO Orange, Red, Orange		A	B	C	D	E	F	G	H	I	J	K	L	M	1016 min.
	15 th	I hate PS ★ Amel, namil208, knon0501		A	B	C	D	E	F	G	H	I	J	K	L	M	1021 min.
	16 th	bitKOIn gs15120, juney, onjo0127		A	B	C	D	E	F	G	H	I	J	K	L	M	1414 min.

프로그래밍 경시대회



어떤 문제를 풀고 있길래 저렇게 몰입하고 있는 걸까?

Problem Solving?

- Problem Solving
 - 수학/퍼즐 문제를 효율적으로 해결하는 방법을 찾는 분야
 - 입력을 받아서 올바른 출력을 내는 프로그램을 만들어야 함

Problem Solving?

- Problem Solving
 - 수학/퍼즐 문제를 효율적으로 해결하는 방법을 찾는 분야
 - 입력을 받아서 올바른 출력을 내는 프로그램을 만들어야 함

문제

2×N 크기의 벽을 2×1, 1×2, 1×1 크기의 타일로 채우는 경우의 수를 구해보자.

입력

첫째 줄에 $N(1 \leq N \leq 1,000,000)$ 이 주어진다.

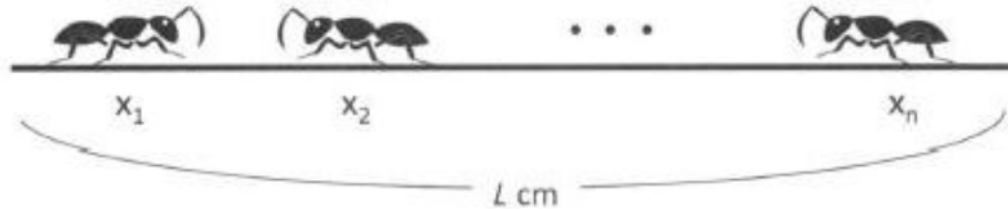
출력

첫째 줄에 경우의 수를 1,000,000,007로 나눈 나머지를 출력한다.

Problem Solving?

- Problem Solving
 - 수학/퍼즐 문제를 효율적으로 해결하는 방법을 찾는 분야
 - 입력을 받아서 올바른 출력을 내는 프로그램을 만들어야 함

길이가 L cm인 수평 막대 위에 n 마리의 개미가 있다. 개미는 왼쪽 또는 오른쪽으로 1cm/s 의 속도로 걷고 있다. 개미는 막대의 끝에 도달하면 밑으로 떨어진다. 또한 두 개미가 마주치면 서로 다른 방향으로 돌아가야 한다. 개미들의 위치와 초기 방향이 주어졌을 때, 모든 개미가 막대 밑으로 떨어지는데 걸리는 시간을 구해보자.



ICPC

- International Collegiate Programming Contest
 - 대회의 구성
 - 국제 결승의 역할을 하는 World Finals
 - 지역 예선의 역할을 하는 Regional Contest
 - 서울 지역 대회
 - 한국 대학교 팀들은 주로 Asia Seoul Regional Contest에 참가
 - 서울 지역 대회는 한국 대학생 프로그래밍 경시대회 본선을 겸함
 - 대회 형식
 - 3인 1팀, 각 팀은 컴퓨터 1대만 사용 가능
 - 25페이지 이내의 문서를 만들어서 가져갈 수 있음

ICPC

- ICPC Asia Seoul Regional Contest
 - 인터넷 예선 (10월)
 - 3시간 10~12문제
 - 각자 학교에 모여서 지도 교수 감독하에 예선 진행
 - 1/2 규칙: 각 학교의 예선 등록 팀 수의 절반 이하만 본선 진출 가능
 - <http://icpckorea.org/archives/2472>
 - 본선 (11월)
 - 5시간 12문제
 - 모든 참가 팀이 한 장소에 모여서 대회 참가
 - 대상 1팀, 금상 1팀, 은상 3팀, 동상 4팀, 장려상 5팀
 - 상위 2~3개 대학은 월드 파이널 진출 가능

재미있어 보이나요?

- PS를 하는 세 가지 유형의 사람
 - Type A. 별로 하기 싫은 사람
 - Type B. 취미로 하는 사람
 - Type C. 여기에 뼈를 묻기로 다짐한 사람
- 그래서 이걸 하면 뭐가 좋을까요?

PS로 얻을 수 있는 것

- Type A. 별로 하기 싫은 사람
 - 개발자로 취업하려면 코딩 테스트 통과해야 해서 하기 싫어도 해야 함
 - 꼭 참고 3~6개월만 열심히 하면 통과할 수 있으니 빨리 통과하고 탈출하자.

PS로 얻을 수 있는 것

- Type B. 취미로 하는 사람
 - 돈 안 드는 좋은 취미
 - 여러 전공 과목을 날로 먹을 수 있음

PS로 얻을 수 있는 것

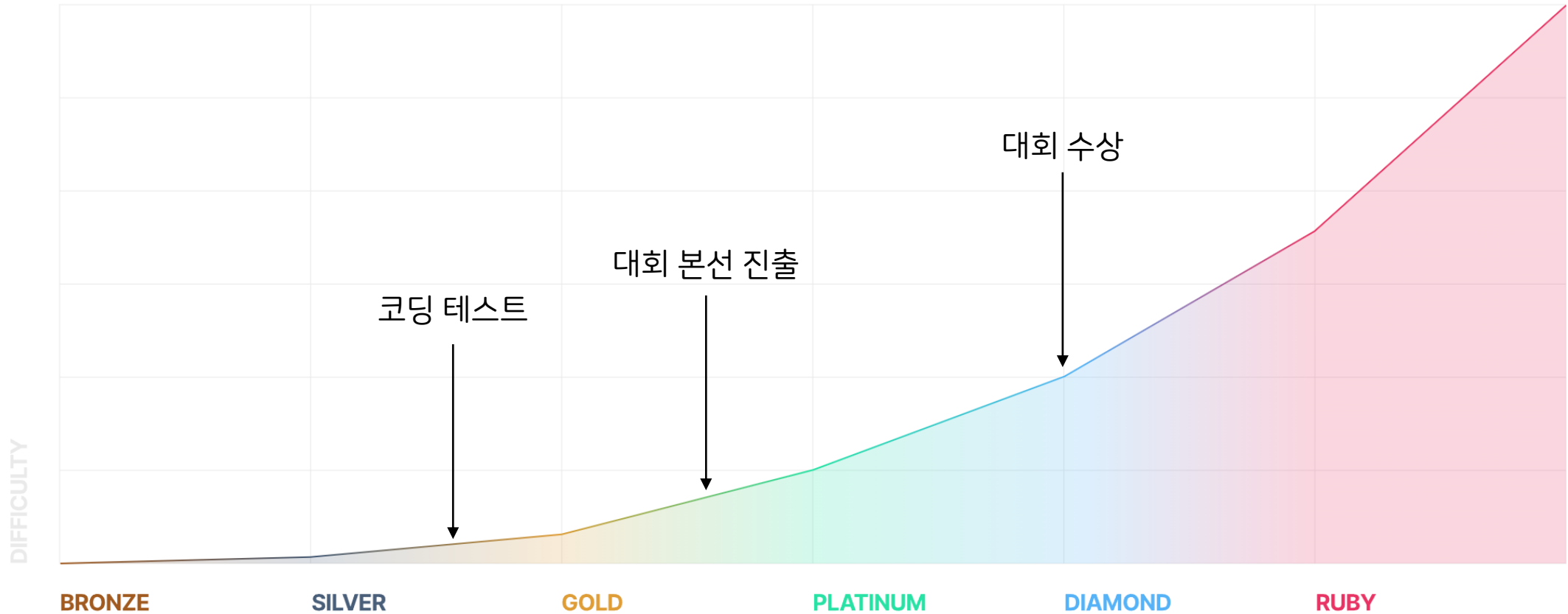
- Type C. 여기에 뼈를 묻기로 다짐한 사람
 - 돈을 벌 수 있음 (과외, 강의, 문제 출제, 대회 상금 등)
 - 평범한 대학생이 얻기 힘든 스펙 (수상 실적, 대기업 임직원 교육 등)
 - 좋은 인맥

하지만 현실은...



하지만 현실은...

- 다음 단계로 넘어갈 때마다 공부해야 하는 양이 기하급수적으로 증가함



하지만 현실은...

- 다음 단계로 넘어갈 때마다 공부해야 하는 양이 기하급수적으로 증가함
- 똑똑한 사람들이 너무 많음
 - 아무리 열심히 공부해도 이길 수 없을 것 같은 사람이 있음
 - 하지만 이런 괴물은 1년에 1명 정도밖에 안 나오고 대회에서 상 받는 사람은 약 50명
 - 우리처럼 평범한 사람도 수상을 노려볼 수 있다!
- 탈출 타이밍 재는 것도 실력
 - 길고 짧은 건 대봐야 알 수 있으니
 - 일단 시작하고 생각해도 괜찮음

필요한 능력과 훈련 방법

필요한 능력

- 여러분이 갖춰야 하는 능력
 - 배경지식
 - 문제 해결 능력
 - 구현 능력
- <https://baactree.tistory.com/52>

필요한 능력

- 배경지식
 - 프로그래밍 문법, 자료구조/알고리즘, 수학적 지식 등 문제를 해결하는 데 필요한 지식
 - 수학에서 근의 공식을 생각해 보자.
 - 똑똑한 사람들은 혼자서 발견할 수도 있지만, 우리는 그런 사람이 아니니까 공부해야 함
- 부족하면 생기는 일
 - 어떻게 풀지 몰라서 해설을 열었는데 외계어가 적혀 있음
- 훈련하는 방법
 - 블로그, 책, 강의, ...
 - 공부하기 가장 쉽고, 가르치기도 가장 쉬움

필요한 능력

- 문제 해결 능력
 - 주어진 문제를 내가 알고 있는 배경지식을 사용할 수 있는 형태로 바꾸는 능력
 - 미분/적분 공식만 암기하고 수능 30번을 풀 수 있을까?
 - 부족하면 생기는 일
 - “이걸 왜 생각 못 했지?”
 - “아 이거 아는 건데...” - 사실 모르는 내용
- 훈련하는 방법
 - ~~좋은 머리를 갖고 다시 태어난다~~
 - 문제를 많이 푼다, 사고 과정을 복기한다, 교수들의 생각을 따라간다, ...

필요한 능력

- 구현 능력
 - 내가 생각한 풀이 과정을 코드로 옮기는 능력
- 부족하면 생기는 일
 - “풀이는 알겠는데 어떻게 구현하지?”
 - “아 이거 다 풀었는데...” - 스코어보드에는 못 푼 걸로 기록되어 있음
- 훈련하는 방법
 - 문제를 많이 푼다, 다른 사람의 코드를 많이 보면서 좋은 구현 방법을 공부한다, ...
 - 고인물 한 명 정해서 코딩 스타일을 따라 하는 것도 좋음

문제를 푸는 방법

- 문제를 푸는 방법
 - 문제를 매우 많이 풀어서 DB에 넣어두고, 문제를 풀 때마다 DB에 쿼리를 날림
 - 똑똑하지 않은 사람들도 충분한 훈련을 통해 잘 사용할 수 있음
 - 문제를 많이 풀고 복기하는 것이 중요함
 - 관찰 - 추측 - 증명 - 관찰 - 추측 - 증명 - ...
 - 머리가 좋을수록 풀이까지의 경로가 짧아지는 경향이 있음
 - 다른 사람이 작성한 풀이를 읽을 때 "이 풀이에 도달하는 사고 과정" 위주로 읽는 것이 좋음
 - 가능한 시간 복잡도와 알고리즘을 모두 나열하고 하나씩 대입해 보면서 생각(...)
 - 풀이를 찾는 Decision Tree를 만들어서 사용(??)
- 여러분에게 잘 맞는 방법을 찾아 보세요

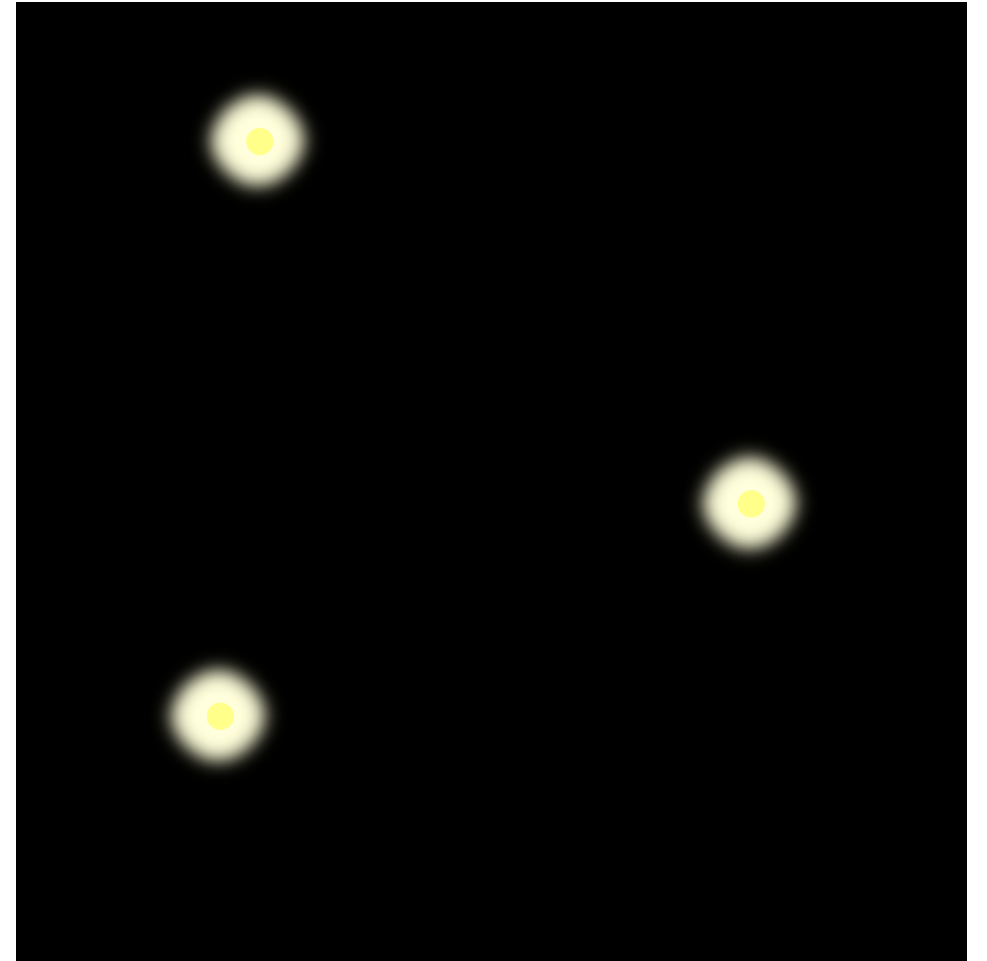
문제를 푸는 과정

- PS는 미로찾기
 - 앞이 보이지 않는 어두운 미로
 - “배경지식”은 미로에 있는 촛불
 - “문제 해결 능력”은 촛불의 밝기



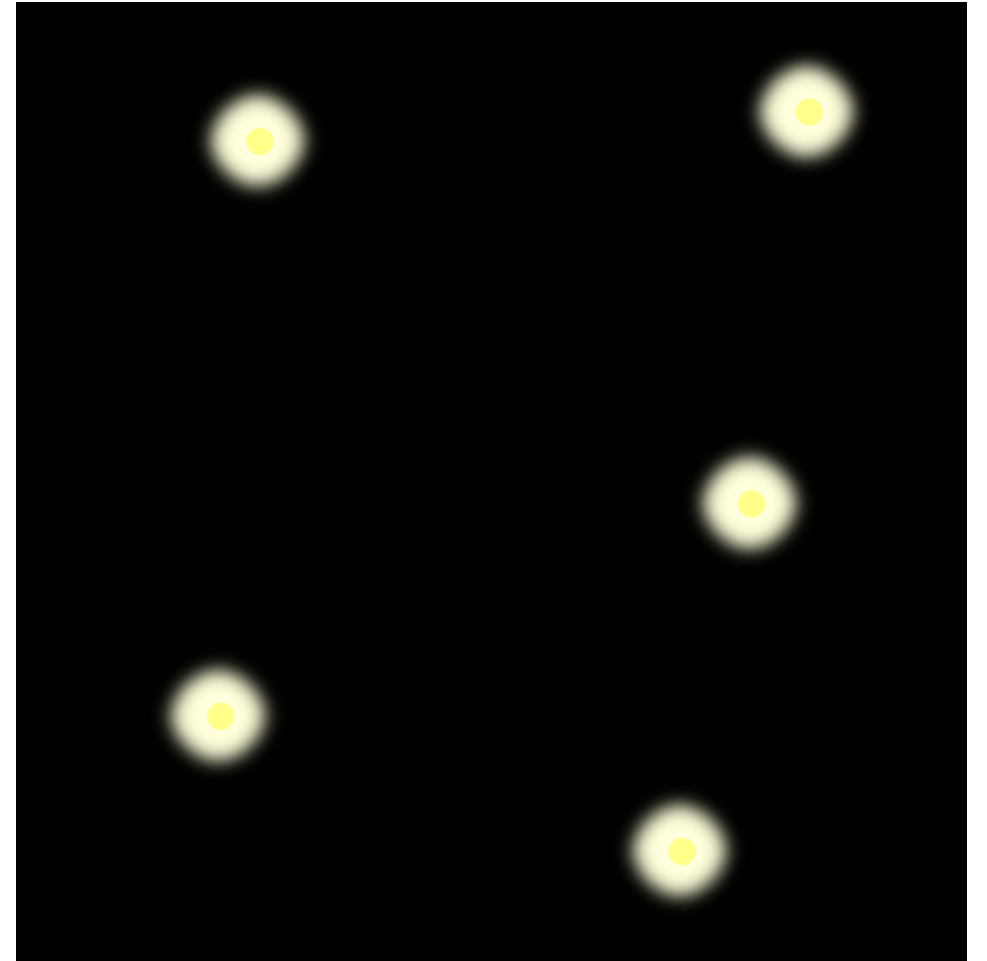
문제를 푸는 과정

- PS는 미로찾기
 - 앞이 보이지 않는 어두운 미로
 - “배경지식”은 미로에 있는 촛불
 - “문제 해결 능력”은 촛불의 밝기



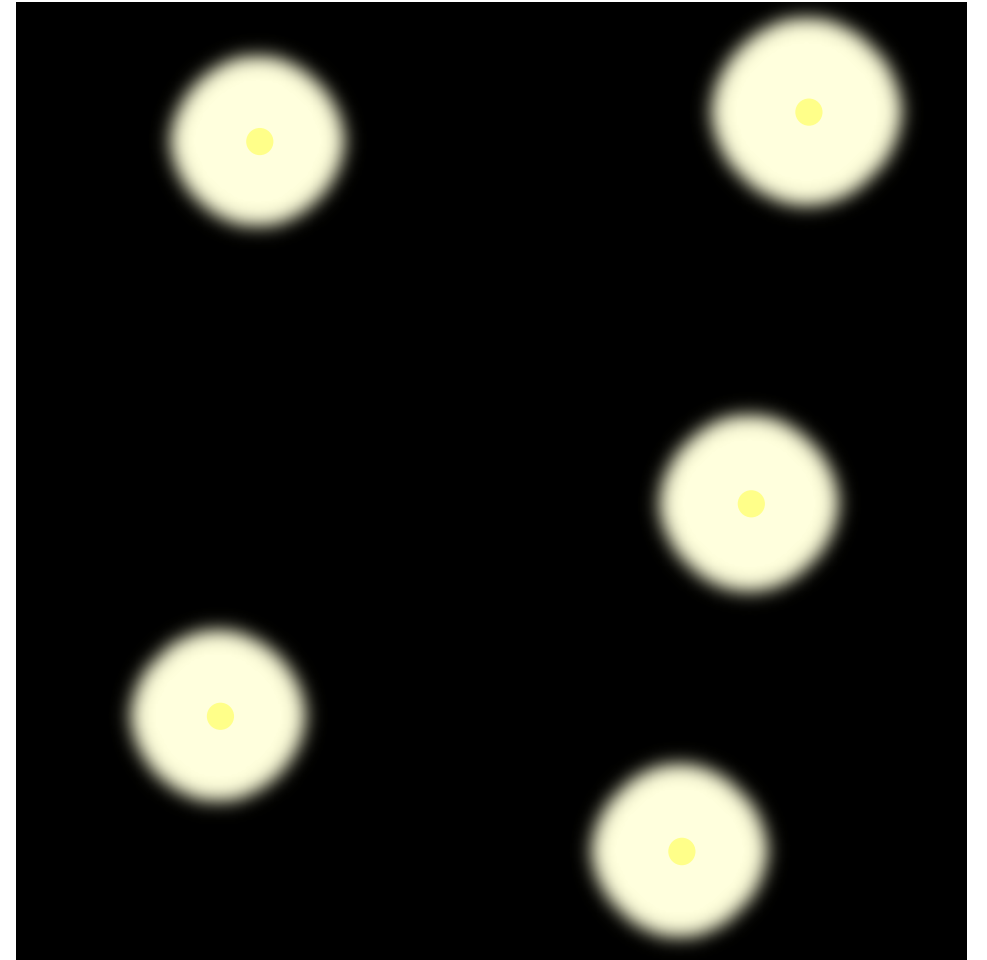
문제를 푸는 과정

- PS는 미로찾기
 - 앞이 보이지 않는 어두운 미로
 - “배경지식”은 미로에 있는 촛불
 - “문제 해결 능력”은 촛불의 밝기
- 배경지식을 공부하면?



문제를 푸는 과정

- PS는 미로찾기
 - 앞이 보이지 않는 어두운 미로
 - “배경지식”은 미로에 있는 촛불
 - “문제 해결 능력”은 촛불의 밝기
- 배경지식을 공부하면?
- 문제 해결 능력을 기르면?



문제를 푸는 방법

- 문제를 푸는 방법

- 문제를 매우 많이 풀어서 DB에 넣어두고, 문제를 풀 때마다 DB에 쿼리를 날림
 - “문제의 유형”도 사전지식으로 취급해서 수많은 곳에 촛불을 설치
- 관찰 - 추측 - 증명 - 관찰 - 추측 - 증명 - ...
 - 합리적인 근거를 토대로 인접한 광원으로 이동
- 가능한 시간 복잡도와 알고리즘을 모두 나열하고 하나씩 대입해 보면서 생각(...)
- 풀이를 찾는 Decision Tree를 만들어서 사용(??)
 - 어떤 광원에서 탐색을 시작할지 결정
- 세 가지 방법 모두 사용할 수 있어야 함

고수가 되는 방법

- 잘해지는 과정

- 내가 잘하는 것 같음 → 재미있어짐 → 재미있어서 더 열심히 공부함 → 잘해짐 → 잘해서 더 재미있음 → 더 열심히 공부함 → 더 잘해짐 → ...

- 이 사이클 안에 들어가야 실력이 꾸준히 올라감

- 처음부터 재미있으면 운이 좋은 거니까 열심히 공부하면 되고
 - 그게 아니라면 일단 잘해지는 것밖에 방법이 없으니 열심히 공부하자.

- 사이클 안에 들어간 다음에는?

- 스트레스받지 않고 PS가 재미있게 느껴지는 범위 안에서만 공부하는 것을 추천함
 - 전 재미없으면 즉시 키보드에서 손 떼고 침대로 갑니다.

추천 자료

- 개념 설명

- kks227 - <https://blog.naver.com/kks227>
- IOI KOREA - <https://www.youtube.com/@ioikorea5159>
- JusticeHui - <https://justicehui.github.io/tutorial>
- 숭실대학교 SCCC 강의 자료 - <https://github.com/justicehui/SSU-SCCC-Study>

- 문제집

- solved.ac 클래스 문제집 - <https://solved.ac/class>
- 백준 온라인 저지 단계별로 풀어보기 - <https://acmicpc.net/step>
- USA Computing Olympiad - <https://boj.kr/usaco>

최근 기출 문제 분석

최근 기출 문제 분석

- 예선

- 2020: S4 G4 G4 G2 G? P5 P4 P1 D5 D5 D? D3

- 5문제 4팀, 총 2팀 진출

- 2021: S5 S1 G2 G1 P4 P4 P2 P2 D5 D4 D4 D4

- 6문제 1팀, 5문제 2팀, 4문제 2팀, 총 4팀 진출

- 2022: S4 S3 S2 G3 G2 P5 D5 D5 D4 D2 R?

- 7문제 1팀, 6문제 3팀, 5문제 1팀, 총 4팀 진출

- 6문제 해결한 팀 중 한 팀은 휴학생 팀이라 본선 X

- 성균관대에서 본선 진출을 하려면 골드 이하의 문제를 모두(약 5문제) 해결해야 함

최근 기출 문제 분석

- 예선 하위 5문제

- 기초적인 개념을 잘 활용하는 문제
- 어려운 배경지식을 공부하는 것보다는 알고 있는 내용을 잘 활용하는 것이 중요

2020	S4	G4	G4	G2	G?
	그리디	최단 경로	스패닝 트리	DP	DFS
2021	S5	S1	G2	G1	P4
	정렬	완전 탐색	DP	기하	그리디
2022	S4	S3	S2	G3	G2
	그리디	그리디	그리디	DP	DP

최근 기출 문제 분석

- 난이도가 비슷한 대회
 - BAPC (<https://boj.kr/bapc>, 벨기에/네덜란드/룩셈부르크)
 - GCPC (<https://boj.kr/gcpc>, 독일)
 - NCPC (<https://boj.kr/ncpc>, 노르웨이/덴마크/스웨덴/아이슬란드/필란드)
 - Southeast USA Regional Division 1 (<https://boj.kr/ser>)
- 조금 더 쉬운 대회
 - BAPC 예선
 - Southeast USA Regional Division 2

최근 기출 문제 분석

- 본선

- 2018: B1 G2 G1 P4 P4 P4 P2 P2 P1 D5 R4 R4
- 2019: S4 G4 G3 G2 P4 D5 D5 D5 D4 D3 D2 D1
- 2020: B1 G3 G2 G2 P3 P2 P1 D5 D4 D3 D2 D1
- 2021: S4 G5 P3 P3 P3 P2 P2 D3 D3 D1 D1 R5
- 2022: S2 G4 G3 G3 P5 P3 P2 D5 D4 D3 D2 R1

- 수상을 하기 위해서는 플래티넘 이하의 모든 문제를 해결해야 함
- 월드 파이널을 노린다면 다이아 하위권 문제를 모두 해결해야 함

최근 기출 문제 분석

- 수상을 노린다면...
 - 플래티넘 ~ 다이아 하위권 문제에 배경지식이 있으면 쉽게 풀 수 있는 문제가 꽤 자주 나옴
 - 2019 예선: 컨벡스헐 트릭(D5)
 - 2019 본선: 센트로이드 분할(P2), 세그먼트 트리 변형(D5)
 - 2020 본선: 도미네이터 트리(D2), FFT(P1), 가우스 소거법(P3), 분할 정복 최적화(D4)
 - 2021 본선: KMP 알고리즘 변형(D3)
 - 2022 예선: 평방 분할(D5)
- 비슷한 컨셉의 문제가 여러 번 출제되기도 함
 - 2021 예선 L(P2) ≡ 2022 본선 C(D4)
 - 2019 본선 H(D5) = 2014 KOI 중등부 4번
 - 2022 본선 B(D2) = 2016 1차 TST 기출 ≡ 2019년 2차 TST 기출
 - 2022 본선 A(D5) = BOJ 16883. 대각 게임
 - 2021 본선 K(D3) = 2011 CEOI 기출 ≡ 2020 1차 TST 기출 = 2021 SCPC 2차 예선 기출

수상을 노린다면...

- 배경지식 공부
 - 대회 전에 급하게 공부하는 것은 소용없음
 - 평소에 solvedac 티어 올릴 겸 공부하는 것 추천
- 팀 노트 준비
 - 각 알고리즘이 어떤 것을 입력받아서 무엇을 계산하는지 알아야 함
 - 빠르고 검증된 코드 사용 - 구글에 Library Checker 검색!
- 웰노운 테크닉 공부
 - 유명한 유형은 한 번씩 다 공부하는 것이 좋음
 - solvedac 클래스 문제집, P3..D3 범위 푼 사람 수로 정렬, ...
 - BOJ에 올라오는 대학교 교내대회 문제 추천

학교 전공과목과의 연계

학교 전공과목 연계

- 컴퓨터교육과: 자료구조, 알고리즘, 컴퓨터구조
- 컴퓨터공학과: 이산수학, 선형대수학, 문제해결기법, 오토마타
- 소프트웨어학과: 확률과 랜덤 프로세스

학교 전공과목 연계

- 컴퓨터 구조
 - 실행 시간을 줄여야 하는 경우가 있음
 - N^2 는 TLE를 받지만, $N^2/8$ 은 AC를 받는 경우
 - N^2 이 정해인 문제에서 N^3 풀이만 알고 있을 때 어쨌든 문제를 풀고 싶은 경우
- 컴퓨터 구조 과목에서 배우는 내용이 큰 도움이 됨
 - 캐시 히트
 - 분기 예측
 - 메모리 정렬, SIMD

학교 전공과목 연계

 성균관대 프로그래밍 대회 상수 커팅


 전체  이미지  뉴스  동영상  도서 : 더보기 도구





검색결과 약 9,510개 (0.27초)






 github.io
https://justicehui.github.io > 2022/02/27

2022 성균관대학교 프로그래밍 경진대회 검수와 상수 커팅 이야기


2022. 2. 27. — 2022년 성균관대학교 프로그래밍 경진대회에 검수자로 참여했습니다. 이번 대회에는 저보다 훨씬 실력이 좋은 분들이 검수자로 함께 참여했습니다. 다른 ...

 2021 ICPC Trio SIMD

 전체  뉴스  이미지  동영상  도서 : 더보기 도구

검색결과 약 8개 (0.34초)

 github.io
https://justicehui.github.io > 2021/11/15 > simd-in-ps

SIMD in PS - ICPC Seoul Regional 2021 L. Trio - JusticeHui

2021. 11. 15. — **SIMD**는 Single Instruction Multiple Data의 약자로, 하나의 명령어로 여러 데이터에 대해 동일한 연산을 동시에 수행하는 방식입니다. 예를 들어 아래 ...

학교 전공과목 연계

- 선형대수학

- 2020 서울 리저널 J번. Switches (BOJ 20178)

- 입력으로 행렬 A 가 주어질 때, 다음과 같은 연립 방정식을 해결해야 함

- $A_{1,1}x_1 + A_{2,1}x_2 + \dots + A_{n,1}x_n = y_1 \pmod{2}$

- $A_{1,2}x_1 + A_{2,2}x_2 + \dots + A_{n,2}x_n = y_2 \pmod{2}$

- ...

- $A_{1,n}x_1 + A_{2,n}x_2 + \dots + A_{n,n}x_n = y_n \pmod{2}$

- $A^T x = b$ 의 해를 구하는 문제

- 가우스 소거법을 이용해 $O(N^3)$ 에 해결하는 문제

학교 전공과목 연계

- 선형대수학
 - 2022 BOJ 연말 대회 G번. 크루스칼 알고리즘
 - 그래프의 스패닝 트리의 개수를 구하는 문제
 - 다음과 같은 행렬 L 을 만들자.
 - $L(i, i) = i$ 의 차수
 - $L(i, j) = (i$ 와 j 를 연결하는 간선의 개수) $\times -1$
 - L 에서 임의의 행과 열을 하나씩 삭제한 다음 행렬식(determinant)를 구하면 정답
 - = Kirchhoff's theorem
- 이 밖에도 Tutte Matrix, LGV Theorem 등 그래프 이론에서 유용한 정리가 있음

학교 전공과목 연계

- 오토마타
 - 정규식을 사용해 파싱하는 문제 → 오토마타 그 자체
 - 동적 계획법: 점화식에서의 상태 전이도 유한 상태 오토마타의 일종이라고 생각할 수 있음

학교 전공과목 연계

- 확률과 랜덤 프로세스
 - 주어진 수열에서 과반을 차지하는 원소가 있는지 판별하는 문제 (BOJ 2912)
 - 임의의 원소를 골라서 그 원소의 등장 횟수를 구함
 - 만약 그 원소가 과반을 차지하고 있으면 정답을 찾은 것
 - 과반을 차지하지 않는다면 다른 원소를 다시 선택하는 것을 반복
 - 30번 반복하면 문제를 해결할 수 있음
 - 왜?

학교 전공과목 연계

- 확률과 랜덤 프로세스

- 주어진 수열에서 과반을 차지하는 원소가 있는지 판별하는 문제 (BOJ 2912)
 - 임의의 원소를 골라서 그 원소의 등장 횟수를 구함
 - 만약 그 원소가 과반을 차지하고 있으면 정답을 찾은 것
 - 과반을 차지하지 않는다면 다른 원소를 다시 선택하는 것을 반복
- 과반을 차지하지 않는 원소를 고를 확률은 아무리 높아 봐야 $1/2$
- 따라서 이 프로세스를 30번 반복했는데 과반인 원소를 찾지 못할 확률은 $1/10^9$
- 채점 프로그램이 고장 날 확률보다 작으니까 맞는 풀이라고 생각해도 무방함

학교 전공과목 연계

- 확률과 랜덤 프로세스

- $N \times N$ 크기의 행렬 A, B, C 가 주어졌을 때 $A \times B = C$ 인지 판별하는 문제

- A 와 B 를 곱한 다음 C 와 비교하면 $O(N^3)$

- 랜덤한 $N \times 1$ 행렬 X 를 만든 다음 $ABX = CX$ 인지 판별

- BX 를 계산하는데 $O(N^2)$, $A(BX)$ 를 계산하는데 $O(N^2)$, CX 를 계산하는데 $O(N^2)$

- X 의 원소가 0 또는 1일 때, $AB \neq C$ 이면서 $ABX = CX$ 일 확률은 $\frac{1}{2}$ 이하

- 따라서 약 30번 정도 반복해서 한 번이라도 다르면 다르다고 판별하면 올바른 알고리즘

- Freivalds' Algorithm

대회 전략과 팀 워크

대회 전략과 팀 워크

- 대회 전략과 팀 워크의 필요성
 - ICPC는 3명이 1대의 컴퓨터를 사용해 문제를 해결하는 대회
 - 개개인의 실력도 중요하지만, 어떤 상황에서 누가 키보드를 잡는지 결정하는 것도 매우 중요함
 - $\max(A, B, C)$ 보다는 $A+B+C$ 를 만드는 것이 좋음
- 목표
 - 낭비되는 시간 감소
 - 서로의 강점을 파악
 - 서로의 약점을 보완
 - 위험 분산
- <https://youtu.be/ECIhAfYeSIk>

대회 전략과 팀 워크

- 낭비되는 시간
 - 최상의 시나리오
 - A가 코딩하는 동안 B와 C는 풀이 고민
 - A가 문제를 풀어서 B가 코딩 시작, C는 A와 풀이 토론
 - B가 문제를 풀어서 C 코딩 시작, A와 B는 각자 새로운 문제를 잡고 고민
 - ...
 - 최악의 시나리오
 - A가 코딩하다가 말려서 디버깅 시작, B와 C는 풀이를 완성했지만 A가 키보드를 차지하고 있는 상태
 - A가 디버깅을 실패하고 B에게 키보드를 넘겨줌
 - B가 코딩하는 중간중간에 A가 키보드를 뺏아서 다시 디버깅함(...)
 - 구현하지 못한 풀이가 3개 남은 채로 대회 종료
- 키보드는 하나밖에 없는 희귀한 자원이기 때문에 효율적으로 사용해야 함

대회 전략과 팀 워크

- 낭비되는 시간
 - 문제를 푸는 과정: 풀이 고민 → 구현 방법 고민 → 구현 → 디버깅
- 풀이 고민 - 키보드 X
 - 키보드가 필요하지 않음
 - 키보드를 잡고 있지 않은 다른 팀원과 토론 가능
- 구현 방법 고민 - 키보드 X
 - 어떤 기능을 함수로 분리하는 것이 좋을지 고민
 - 함수의 입력과 출력을 정의
 - 코딩이 얼마나 걸릴지 예측
- 구현 - 키보드 O
- 디버깅 - 키보드 ?
 - 뒤에서 다시 이야기함

대회 전략과 팀 워크

- 기본적인 행동 요령
 - 스코어보드 활용
 - 디버깅이 필요할 때
 - 다른 팀원이 키보드를 잡고 있을 때
 - 키보드를 잡고 있는 사람이 없을 때

대회 전략과 팀 워크

- 스코어보드 활용
 - 쉬운 문제부터 빨리 푸는 것이 좋음
 - 어떤 문제를 풀어야 할지 모르겠다면 많이 풀린 문제부터 풀어보자!
 - 한 번에 맞춘 팀이 없으면 실수하기 쉬운 문제이므로 신중하게 코딩
- 수상을 노린다면...
 - 다른 잘하는 팀이 잘 푸는 문제의 유형을 미리 숙지
 - 그 팀만 빨리 푼 문제가 있으면 때에 따라 그 문제를 건들지 않는 게 좋을 수도 있음

대회 전략과 팀 워크

- 디버깅이 필요할 때
 - 디버거를 사용하는 것은 다른 팀원이 코딩할 기회를 뺏는 것
 - 구현할 문제가 밀려 있는데 디버깅이 오래 걸릴 것 같으면 비키는 것이 좋음
 - 최대한 디버거를 사용하지 않고 디버깅해야 함 → 제출과 동시에 코드 인쇄 요청
- 눈으로 디버깅하는 연습
 - 논리가 잘못되지 않았는지
 - 코드로 잘못 옮기지 않았는지
 - 부등호 방향, i j k 제대로 썼는지
 - 사용하지 않는 변수, 함수
 - ...
- 의외로 다른 사람이 보면 잘 보일 수 있음

대회 전략과 팀 워크

- 다른 팀원이 키보드를 잡고 있을 때
 - 혼수 두기 (페어 코딩)
 - 다른 문제 의사 코드 작성
 - 코드 종이에 인쇄해서 디버깅
 - 내 구현이 5~10분 이내로 끝날 것 같으면 키보드 뺏는 것도 고려

대회 전략과 팀 워크

- 키보드를 잡고 있는 사람이 없을 때
 - 가설을 검증하고 싶으면 naïve 코드 작성해서 테스트
 - 구현할 게 많아서 미뤄둔 문제 조금씩 코딩
 - 인쇄된 코드를 디버깅하고 있다면 스트레스 테스트* 작성
 - 항상 정확한 정답을 구하는 느린 코드를 함께 작성한 다음
 - 랜덤 데이터를 집어넣으면서 반례를 찾는 프로그램
- 키보드가 비어 있다는 것을 의식해서 성급하게 구현하는 것은 좋은 선택이 아님
 - 어떻게 구현할지, 구현하는 데 시간이 얼마나 걸릴지 계산한 다음
 - 팀원들에게 예상 시간을 이야기하고 코딩 시작

대회 전략과 팀 워크

- 개인플레이 위주 - (Solver/Coder) * 3
 - 3명이 각자 문제를 풀다가 막히면 토론
 - 3명의 실력이 비슷하거나 분야가 겹치지 않을 때 주로 사용
- 매니저 중심 - (Solver/Coder) * 2 + Manager
 - 팀마다 세부 전략이 다름

2022 숭실대학교 NLP

- 팀 소개

- 2022 ICPC 서울 리저널 5등(은상, 대학 3등)
- 2022 ICPC 서울 리저널 예선 8등(대학 3등)

- 팀 구성

- jhna917 – BOJ 9000문제, solvedac 마스터, Codeforces 오렌지
- edenooo – BOJ 3000문제, solvedac 루비3, Codeforces 레드
- chansol – BOJ 1500문제, solvedac 다이아4, Codeforces 퍼플

2022 송실대학교 NLP

- 팀원의 장점

- jhnah917

- 특히 solvedac 기준 골드 1 이하의 쉬운 문제를 빠르게 해결할 수 있음
 - 구현 속도가 매우 빠름
 - 자료구조, 그래프, 기하 알고리즘을 많이 알고 있음

- edenooo

- 코드포스 레드 - 문제 풀이 능력, 코딩 속도 모두 상위권
 - 팀원 중 수학을 가장 잘함
 - 대회 참가 경험이 매우 많음

- chansol

- 풍부한 개발 경험을 바탕으로 키보드가 비었을 때 스트레스 테스트 등을 잘 구현함

2022 송실대학교 NLP

- 팀원의 단점
 - jhnah917
 - 수학을 잘 못함
 - 오래 고민해야 하는 문제를 잘 풀지 못함
 - 30분 안에 풀이를 못 찾으면 5시간이 지나도 못 찾는 경우가 많음
 - 한 번 말리면 크게 말림
 - edenooo
 - 실수를 자주 함
 - 익숙하지 않은 분야가 꽤 있음
 - chansol
 - 다른 2명에 비해 문제를 많이 안 풀었음

2022 송실대학교 NLP

- 팀 구성의 장점
 - jhnah917은 주로 백준, edenooo는 코드포스에서 활동
 - 주로 활동하는 사이트가 다르고, 각자 사이트에서 최상위권
 - 각자의 강점이 다르므로 서로의 단점을 잘 커버할 수 있음
- 팀 구성의 단점
 - 대회 시작 후 3시간이 지나면 퍼포먼스가 급격하게 떨어짐
 - 특히 jhnah917
 - 팀 연습을 통해 3시간 동안 최대한 많이 푸는 연습을 주로 했음

2022 송실대학교 NLP

- 팀 전략 - 대회 초반 (-10 ~ 60분)
 - 대회 시작 전에 대회장 앞에 준비된 풍선의 개수를 보고 문제 난이도 유추

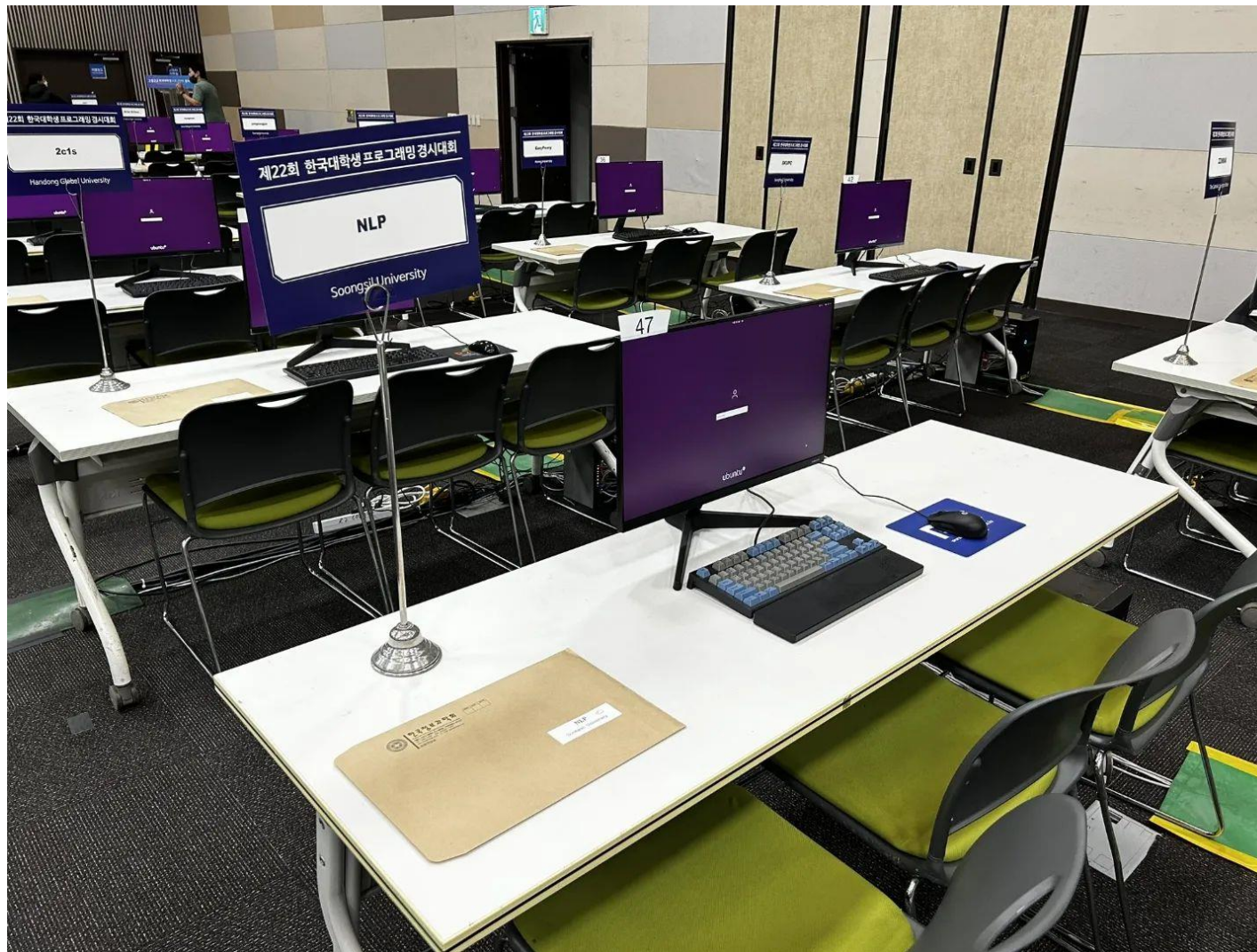
2022 숭실대학교 NLP



2022 송실대학교 NLP

- 팀 전략 - 대회 초반 (-10 ~ 60분)
 - 대회 시작 전에 대회장 앞에 준비된 풍선의 개수를 보고 문제 난이도 유추
 - edenooo가 문제지 봉인을 뜯은 다음, 쉽다고 파악된 문제를 빠르게 jhnah917에게 전달
 - chansol은 컴퓨터 로그인, IDE 세팅하고 모니터를 오른쪽으로 옮김
 - jhnah917은 쉬운 문제를 받아서 빠르게 해결

2022 송실대학교 NLP



2022 송실대학교 NLP

- 팀 전략 - 대회 초반 (-10 ~ 60분)
 - 대회 시작 전에 대회장 앞에 준비된 풍선의 개수를 보고 문제 난이도 유추
 - edenooo가 문제지 봉인을 뜯은 다음, 쉽다고 파악된 문제를 빠르게 jhnah917에게 전달
 - chansol은 컴퓨터 로그인, IDE 세팅하고 모니터를 오른쪽으로 옮김
 - jhnah917은 쉬운 문제를 받아서 빠르게 해결
 - chansol은 쉬운 문제를 찾아서 번역하고 예제를 손으로 그려서 다른 2명에게 전달
 - 스코어보드, 풍선 개수 등의 정보 활용
 - edenooo, jhnah917은 문제를 전달받은 다음 자리를 바꿔가며 코딩
 - 30분 안에 해결할 수 있는 쉬운 문제를 모두 해결하면 대회 초반은 끝.

2022 송실대학교 NLP

- 팀 전략 - 대회 초반 (-10 ~ 60분)
 - 12min. K(**G3**) **WA**
 - 15min. K(**G3**) **AC**
 - I는 쉬울 것 같은데 풀이가 안 보여서 edenooo한테 넘김
 - K는 쉬워서 바로 해결
 - 19min. I(**G3**) **AC**
 - K 코딩 끝나자마자 edenooo가 키보드 잡고 해결
 - 21min. J(**S2**) **AC**
 - 스코어보드에서 J 많이 풀렸길래 edenooo가 I 푸는 동안 읽음
 - I 코딩 끝나자마자 jhna917이 키보드 잡고 해결
 - 이후 jhna917은 L, edenooo는 E, chansol은 F 잡음
 - 51min. E(**P5**) **AC**
 - 56min. F(**G4**) **AC**

2022 송실대학교 NLP

2022 ICPC Seoul Regional

00:56

6	1 st	C14H9Cl5 Seoul National University	A	B	C	D	E	F	G	H	I	J	K	L	178 min.
5	2 nd	HappyLastDance Seoul National University	A	B	C	D	E	F	G	H	I	J	K	L	114 min.
	3 rd	BabyPenguin KAIST	A	B	C	D	E	F	G	H	I	J	K	L	139 min.
	4 th	NLP Soongsil University	A	B	C	D	E	F	G	H	I	J	K	L	181 min.
4	5 th	000102 Pohang University of Science and Technology	A	B	C	D	E	F	G	H	I	J	K	L	111 min.
	6 th	I hate PS Korea University	A	B	C	D	E	F	G	H	I	J	K	L	112 min.
	7 th	NewTrend Seoul National University	A	B	C	D	E	F	G	H	I	J	K	L	123 min.
	8 th	LaSt Chance Korea University	A	B	C	D	E	F	G	H	I	J	K	L	136 min.
	9 th	Last Coin Sogang University	A	B	C	D	E	F	G	H	I	J	K	L	141 min.
	10 th	OReO Hanyang University	A	B	C	D	E	F	G	H	I	J	K	L	146 min.

2022 송실대학교 NLP

- 팀 전략 - 대회 중반 (60 ~ 180분)
 - 긴 호흡을 갖고 고민해야 하는 문제를 해결해야 하는 단계
 - 문제를 잘 나눠 가져야 함
 - 전형적이거나 어려운 개념을 사용해야 하는 문제는 jhnah917
 - 오래 고민해야 하는 문제는 edenooo
 - chansol은 페어 코딩 또는 디버깅
- 못 풀면 문제 교환하거나 토론
- 여기까지 해서 높은 등수에 올라왔다면 거의 다 완성!

2022 송실대학교 NLP

- 팀 전략 - 대회 중반 (60 ~ 180분)
 - 59min. L(P2) WA
 - jhna917이 코딩해서 틀림
 - 디버깅하던 도중 edenooo가 D 코딩한다고 키보드 뺏어감
 - 코드 프린트해서 chansol과 함께 디버깅 시작
 - 86min. D(P3) AC
 - 108min. L(P2) RTE
 - 109min. L(P2) WA
 - edenooo가 L 문제지 압수하고 C 풀라고 던져 줌
 - edenooo/chansol은 L 새로운 풀이 찾으려 감
 - 140min. C(D4) AC
 - 틀리면 디버깅 절대 못할 것 같아서 벌벌 떨면서 제출했는데 운 좋게도 한 번에 해결
 - 157min. L(P2) AC
 - edenooo가 새로운 풀이로 처음부터 다시 짜서 해결
 - 코딩하는 동안 chansol은 데이터 제작함

2022 송실대학교 NLP

2022 ICPC Seoul Regional

402 03:00

9	1 st	C14H9Cl5 Seoul National University
	2 nd	HappyLastDance Seoul National University
	3 rd	BabyPenguin KAIST
8	4 th	NLP Soongsil University
	5 th	NewTrend Seoul National University
7	6 th	000102 Pohang University of Science and Technology
	7 th	kopagi Seoul National University
	8 th	nangman Seoul National University
	9 th	LaSt Chance Korea University
	10 th	Undergoose SungKyunKwan University

A +1	B -2	C	D	E	F	G	H	I	J	K	L	461 min.
A	B	C +1	D	E +1	F	G	H	I	J	K	L +2	581 min.
A +1	B	C	D	E	F	G	H	I	J	K	L +2	637 min.
A	B	C	D	E	F	G	H	I	J	K +1	L +3	624 min.
A	B	C +1	D	E	F	G	H	I	J	K +1	L +1	679 min.
A	B	C	D	E +1	F	G	H	I	J	K +1	L	448 min.
A	B	C	D +1	E	F	G	H	I	J +1	K +1	L -1	575 min.
A	B	C	D	E	F	G	H	I	J	K +1	L	611 min.
A -1	B	C	D +1	E +4	F +1	G	H	I	J	K	L +2	652 min.
A	B	C	D +2	E +2	F	G	H	I +1	J	K	L +1	754 min.

2022 송실대학교 NLP

- 팀 전략 - 대회 후반 (180 ~ 300분)
 - 수상을 가르는 문제 하나를 풀어내야 함
 - edenooo를 믿고 기도하는 전략
 - chansol은 손으로 테스트 케이스 제작
 - jhnah917은 페어 코딩하거나 어려운 문제를 "이전에 풀어본 문제"로 바꿀 수 있을지 고민
- 다른 팀을 향한 심리전
 - 대회 종료 1~2시간 전부터 스코어보드 갱신이 중단됨
 - 다른 팀이 어떤 문제에 제출한 횟수는 알 수 있지만, 정답 여부는 알 수 없음
 - 대회 종료 20분 전과 17분 전에 한 번씩 의미 없는 제출을 했음
 - 한 문제를 더 푼 것처럼 연기

2022 송실대학교 NLP

- 팀 전략 - 대회 후반 (180 ~ 300분)
 - 198min. A(D5) WA
 - edenooo와 jhna917이 풀이 찾고 jhna917이 코딩 시작
 - 199min. A(D5) WA
 - 233min. A(D5) WA
 - 이대로 가면 말릴 것 같아서 edenooo가 키보드 압수해서 처음부터 다시 짜기 시작
 - 254min. A(D5) AC
 - edenooo가 다시 짜서 해결
 - 280min. H(D3) WA
 - 283min. H(D3) WA
 - 블러핑

2022 송실대학교 NLP

2022 ICPC Seoul Regional

05:00

11	1 st	C14H9Cl5 Seoul National University	A +1	B +2	C	D	E	F	G -5	H +1	I	J	K	L	955 min.
10	2 nd	BabyPenguin KAIST	A +1	B -6	C	D	E	F	G	H	I	J	K	L +2	847 min.
	3 rd	HappyLastDance Seoul National University	A	B -2	C +1	D	E +1	F	G	H +1	I	J	K	L +2	860 min.
9	4 th	NewTrend Seoul National University	A +2	B -5	C +1	D	E	F	G -1	H	I	J	K +1	L +1	914 min.
	5 th	NLP Soongsil University	A +3	B	C	D	E	F	G	H -2	I	J	K +1	L +3	938 min.
	6 th	Freshquery and men KAIST	A +2	B	C +3	D	E	F	G	H	I	J	K	L +3	1256 min.
	7 th	MunSongSong Eggdrop KAIST	A +2	B -2	C	D +3	E +2	F	G	H -6	I	J	K	L +4	1452 min.
8	8 th	000102 Pohang University of Science and Technology	A +4	B	C	D	E +1	F	G	H	I	J	K +1	L	757 min.
	9 th	kopagi Seoul National University	A	B -1	C	D +1	E	F	G	H	I	J	K +1	L +1	793 min.
	10 th	I hate PS Korea University	A -2	B	C	D	E +2	F	G	H	I	J	K	L +11	1037 min.
	11 th	Undergoose SungKyunKwan University	A -5	B	C +2	D +2	E +2	F	G	H	I +1	J	K	L +1	1085 min.
	12 th	WooSeungHanDae Hanyang University	A -3	B	C	D +1	E	F	G	H	I +1	J	K	L +4	1120 min.
	13 th	DAEYOUNG1020304 Seoul National University	A -1	B -1	C	D +1	E	F	G -1	H -1	I	J	K	L +9	1185 min.

다른 팀의 전략

- 2020 고려대학교 1_Hoeaeng_2_Hawawang
 - 2021 ICPC 월드 파이널 참가
 - 2020 ICPC 서울 리저널 7등(동상, 대학 3등)
 - <https://ryute.tistory.com/62>
- 2017 서울대학교 MolaMola
 - 2018 ICPC 월드 파이널 5등(은상)
 - 2017 ICPC 서울 리저널 1등(대상, 대학 1등)
 - <https://zigui.tistory.com/19>
- 2022 포항공대 000102
 - 2022 ICPC 서울 리저널 8등(동상, 대학 4등)
 - <https://leo630.tistory.com/147>
- 2017 도쿄대학교 Cxiv-Dxiv
 - 2018 ICPC 월드 파이널 4등(금상)
 - 2017 ICPC 쓰쿠바 리저널 1등
 - <https://degwer.hatenablog.com/entries/2018/04/22>

팀 연습

- 팀 연습은 꼭 해야 함
 - 특히 (Solver/Coder) * 2 + Manager 전략은 팀원 간의 호흡이 중요함
 - 많은 연습을 통해 각자의 역할과 상황별 대처 방법을 조정해야 함
- 치명적인 실수를 하지 않도록 반복 훈련
 - 대회가 끝날 때까지 안 읽은 문제가 있는 경우
 - 한 문제를 3시간 동안 디버깅하다가 대회가 끝나는 경우
 - ...
- 대회 때는 어떤 팀원이 어떤 문제를 읽었는지 체크하는 표를 만드는 것이 좋음
 - 문제를 읽었으면 O 표시
 - 풀이를 알 것 같으면 √ 표시
 - 풀었으면 문제지를 바닥에 버림

꿀팁

- ICPC 예선
 - 한글 문제와 영어 문제가 있음
 - 한글로 된 3~4 문제가 가장 쉬운 문제이므로 한글 문제부터 빠르게 해결
- ICPC 본선 진출
 - 학교의 예선 참가 팀 수의 절반까지만 본선에 진출할 수 있음
 - 친구를 열심히 꼬셔서 예선에 등록하게 만들자.

Q&A

slido.com