

2024 NPC 모의대회 #0 - Div1+2

Official Solutions

by

소프트웨어 19 박병규

- ✓ solved.ac 티어, 알고리즘 분류를 꺼주세요. (설정 → 보기 → 보지 않기로 변경)
- ✓ 문제 지문은 모두 영어이며, 문제 번호는 난이도 순입니다.
- ✓ 대회 중 모든 검색은 허용됩니다. (티어, 알고리즘 분류는 보지 말아주세요!)
- ✓ 20:20-22:20 대회
- ✓ 22:30-23:00 문제 풀이

문제		난이도	알고리즘
A	Streets Ahead	Silver5	set/map
B	Code Guessing	Silver5	bruteforce, case_work
C	Corrupted Gradebook	Gold4-5	dynamic_programming
D	Shuffles	Gold4	math, adhoc
E	ABC String	Gold1	greedy
F	Hopscotch	Platinum3	math, combinatorics, dnc

A. Streets Ahead

set/map

난이도 – **Silver5**

- ✓ 제출 ?? 번, 정답 ?? 명 (정답률 ??.??%)
- ✓ 처음 푼 사람: ??, ?? 분

- ✓ Naive한 방법은 $\mathcal{O}(QN|S|)$ 으로 시간초과가 발생합니다.
- ✓ BBST 혹은 Hash 기반 key-value 자료구조를 사용할 수 있습니다.
- ✓ C++에서는 map, python에서는 dictionary를 사용하면 $\mathcal{O}(Q|S|\log N)$ 에 해결할 수 있습니다.

B. Code Guessing

bruteforce, case_work

난이도 – **Silver5**

- ✓ 제출 ?? 번, 정답 ?? 명 (정답률 ??.??%)
- ✓ 처음 푼 사람: ??, ?? 분

- ✓ bruteforce: ${}_9C_2$ 개의 Bob 숫자 경우의 수를 모두 탐색하며 정답을 찾을 수 있습니다.
- ✓ case work: Alice의 카드 숫자가 (i, j) 일 때, 다음과 같은 경우 정답이 존재합니다.
- ✓ ABBA이고 $j - i = 3$ 일 때, ABAB이고 $i = 6, j = 8$ 일 때, BABA이고 $i = 2, j = 4$ 일 때,
- ✓ AABB이고 $j = 7$ 일 때, BBAA이고 $i = 3$ 일 때, BAAB이고 $i = 2, j = 8$ 일 때,
- ✓ 이 경우를 모두 검사해 답을 구하면 됩니다.

C. Corrupted Gradebook

dynamic_programming

난이도 - Gold4-5

- ✓ 제출 ?? 번, 정답 ?? 명 (정답률 ??.??%)
- ✓ 처음 푼 사람: ??, ?? 분

- ✓ string의 길이 $|S|$ 는 최대 $3G$ 임이 보장됩니다.
- ✓ 이 때 파싱할 수 있는 모든 경우의 수를 탐색한다면, 시간복잡도는 $\mathcal{O}(CG_{|S|}C_G)$ 로, 시간초과가 발생합니다.
- ✓ 최댓값만 저장하면 되므로, i 번째 문자까지 j 개의 수로 파싱했을 때 최대 점수 합만을 저장하면 어떨까요?
- ✓ $dp(0, 0) = 0, ans = dp(|S|, G)$ 로 두고 dp 를 돌릴 수 있습니다.

- ✓ 점화식은 $dp(i, j) = \max_{k=1}^3 (dp(i - k, j - 1) + getInt(i - k + 1, i))$ 입니다.
- ✓ $getInt(i - k + 1, i)$ 가 0 이 아니면서 0 으로 시작하는지, 100 을 초과하는지도 체크합니다.
- ✓ 최종적으로 구해야 할 평균은 double로 바꿔 round 연산을 수행하거나, 나머지 연산으로 구합니다.
- ✓ 시간복잡도는 $\mathcal{O}(CG^2)$ 입니다.

D. Shuffles

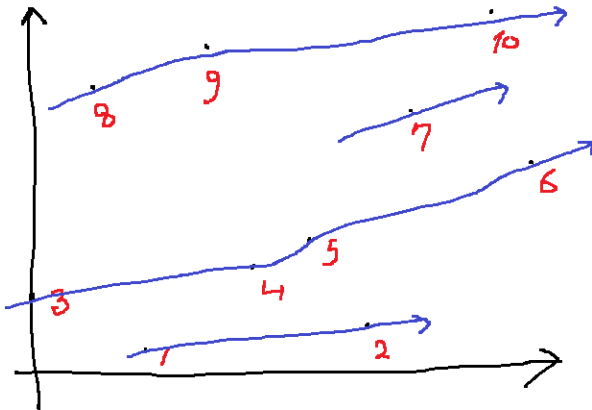
math, adhoc

난이도 - Gold4

- ✓ 제출 ?? 번, 정답 ?? 명 (정답률 ??.??%)
- ✓ 처음 푼 사람: ??, ?? 분

D. Shuffles

✓ 한 가지 관찰을 해봅시다. 예제 입력2 배열의 x 축을 인덱스, y 축을 값으로 한 그래프입니다.



D. Shuffles

- ✓ 각 shuffle마다 파란색 화살표의 개수를 최대 2배로 만들 수 있습니다.
- ✓ 그럼 이 화살표의 개수를 어떻게 $\mathcal{O}(N)$ 만에 구할 수 있을까요?

- ✓ 각 숫자를 인덱스로, 위치를 값으로 하는 pos 배열을 만들어봅시다.
- ✓ $arr = 1, 2, 7, 3, 8, 9, 4, 5, 10, 6, pos = 0, 1, 3, 6, 7, 9, 2, 4, 5, 8$ 처럼 만들 수 있습니다.
- ✓ 이 때, $pos[i] > pos[i + 1]$ 의 개수 (cnt) 를 구하면, $cnt + 1$ 은 파란색 화살표의 개수입니다.
- ✓ $cnt + 1$ 이상인 가장 작은 2^x 의 x 가 답이 됩니다.

E. ABC String

greedy

난이도 - Gold1

- ✓ 제출 ?? 번, 정답 ?? 명 (정답률 ??.??%)
- ✓ 처음 푼 사람: ??, ?? 분

E. ABC String

- ✓ 문자열을 돌면서 시뮬레이션을 해봅니다.
- ✓ 예를 들어 $AABBCC$ 일 때, AA 에서 이미 2개가 필요함을 알 수 있습니다.
- ✓ $AABB$ 에서 AB, AB 로 나눌 수 있습니다.
- ✓ $AABBCC$ 에서 ABC, ABC 로 나뉘어져 답이 2가 됩니다.

E. ABC String

- ✓ 문자열을 돌면서 prefix마다 A, B, C 각각의 개수 합을 구해볼까요?
- ✓ 예를 들어 특정 인덱스 i 에서 $cnt(A) = 4, cnt(B) = 3, cnt(C) = 2$ 일 때, 문자열을 어떻게 나눠야 가장 적은 개수로 만들 수 있을까요?
- ✓ $ABCABCAB$, A 처럼 나누면 최소 개수가 됩니다.

E. ABC String

- ✓ $\min(cnt(A), cnt(B), cnt(C))$ 는 ABC 의 개수,
 $\max(cnt(A), cnt(B), cnt(C)) - \min(cnt(A), cnt(B), cnt(C))$ 는 ABC 를 이루지 못한
 A, B, C, AB, AC, BC 의 개수입니다.
- ✓ $\max(cnt(A), cnt(B), cnt(C)) - \min(cnt(A), cnt(B), cnt(C))$ 중 한 개는 ABC 로
이동할 수 있습니다.
- ✓ 즉 일반화하면, 모든 prefix의
 $\max(cnt(A), cnt(B), cnt(C)) - \min(cnt(A), cnt(B), cnt(C))$ 의 값을 구해 최대값을
구하면 답이 됩니다.

F. Hopscotch

math, combinatorics, dnc

난이도 - **Platinum3**

- ✓ 제출 ?? 번, 정답 ?? 명 (정답률 ??.??%)
- ✓ 처음 푼 사람: ??, ?? 분

- ✓ 문제의 경우를 수식으로 나타내볼까요?
- ✓ $x_1 + x_2 + \dots + x_m = N - mX, y_1 + y_2 + \dots + y_m = N - mY (x_i, y_i \geq 0)$ 입니다.
- ✓ 각 x, y 값 경우의 수는 독립사건이므로, 따로 구해서 곱해줄 수 있습니다.
- ✓ 이 수식을 보고 우리는 고등학교 확통을 떠올릴 수 있습니다.

F. Hopscotch

- ✓ 네, 중복조합입니다. hop 수가 m 회일 때 경우의 수는 ${}_mH_{N-mX} \times {}_mH_{N-mY}$
- ✓ $= {}_{m(1-X)+N-1}C_{N-mX} \times {}_{m(1-Y)+N-1}C_{N-mY}$ 입니다.
- ✓ 이 값을 전처리 없이 구한다면 시간초과가 뜰 것입니다.

- ✓ 이 값을 빠르게 구하기 위해, $1!$ 부터 $N!$ 및 이들의 모듈러 역원까지 미리 구해놔야 합니다.
- ✓ 모듈러 역원은 문제 지문 (페르마의 소정리) 에서처럼 $(N!)^{mod-2}$ 를 구하면 됩니다.
- ✓ 이 값은 분할정복을 활용한 거듭제곱을 통해 $\mathcal{O}(\log(mod))$ 에 구할 수 있습니다.
- ✓ 여기까지 시간복잡도는 $\mathcal{O}(N \log(mod))$ 입니다.

F. Hopscotch

- ✓ 이제 어떤 ${}_nC_r$ 이 들어와도 $\mathcal{O}(1)$ 에 구할 수 있습니다.
- ✓ $N - mX \geq 0, N - mY \geq 0$ 인 m 을 돌며 모든 값을 구해줍니다.
- ✓ 총 시간복잡도는 $\mathcal{O}(N \log(mod))$ 입니다.