

제6회 천하제일 코딩대회

Official Solutions

출제 및 검수

✓ 김준겸 ryute	구글 코리아	고려대학교 컴퓨터학과
✓ 김준서 junseo		한양대학교 컴퓨터소프트웨어학부
✓ 나정휘 jhnah917		숭실대학교 컴퓨터학부
✓ 박세훈 prarie		성균관대학교 소프트웨어학과
✓ 박찬솔 chansol		숭실대학교 컴퓨터학부
✓ 심준 wesley2003		국민대학교 소프트웨어학부
✓ 오주원 kyo20111		숭실대학교 소프트웨어학부
✓ 이성현 hibye1217		한양대학교 컴퓨터소프트웨어학부
✓ 정현서 jhweszt2		서울대학교 컴퓨터공학부

현장 스태프

- ✓ 김동건 선린인터넷고등학교 소프트웨어과
- ✓ 김서윤 선린인터넷고등학교 소프트웨어과
- ✓ 김세희 선린인터넷고등학교 소프트웨어과
- ✓ 이제윤 선린인터넷고등학교 소프트웨어과
- ✓ 심민성 third1234 선린인터넷고등학교 소프트웨어과
- ✓ 조찬우 myyh1234 선린인터넷고등학교 소프트웨어과
- ✓ 이름을 공개하지 않는 여러 사람들

Sponsors



예선 문제	의도한 난이도	출제자
QA 심준의 병역판정검사	Easy	심준
QB 11월 11일	Easy	이성현
QC 순열 정렬	Medium	박세훈
QD 영어 시험	Medium	나정휘
QE 가장 긴 등차 부분 수열	Hard	박세훈

본선 문제	의도한 난이도	출제자
A Gravity Hackenbush	Medium	이성현
B K-균형 잡힌 수	Challenging	오주원
C Merge the Tree and Sequence	Hard	이성현
D 바지 구매	Easy	나정휘
E 반전 수와 쿼리	Medium	김준서
F 시간딱딱총	Hard	김준서
G 인공 신경망	Hard	나정휘
H 최대 최소공배수	Medium	박세훈
I 최장 최장 증가 부분 수열	Hard	나정휘
J 행성 정렬	Medium	김준서

QA. 심준의 병역판정검사

arithmetic, implementation

출제진 의도 – **Easy**

- ✓ 처음 푼 사람: **정태건**, 4분
- ✓ 처음 푼 사람(Open Contest): **riroan**, 3분
- ✓ 출제자: 심준

QA. 심준의 병역판정검사

- ✓ 주어진 표를 꼼꼼하게 조건문으로 옮기면 해결할 수 있습니다.
- ✓ C, C++, Java에서 정수 나눗셈의 결과는 정수라는 것에 주의해야 합니다.

QB. 11월 11일

implementation

출제진 의도 – **Easy**

- ✓ 처음 푼 사람: **장태환**, 11분
- ✓ 처음 푼 사람(Open Contest): **riroan**, 7분
- ✓ 출제자: 이성현

QB. 11월 11일

- ✓ m 월 m 일의 m 일 전은 $m - 1$ 월 말입니다.
- ✓ 윤년이 주어지는 경우와 해가 바뀌는 경우에 주의해야 합니다.

QC. 순열 정렬

greedy

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: **장태환**, 17분
- ✓ 처음 푼 사람(Open Contest): **riroan**, 11분
- ✓ 출제자: 박세훈

QC. 순열 정렬

- ✓ 맨 앞에 있는 수를 최대한 작게 만드는 것이 좋습니다.
- ✓ 두 번째 수는 맨 앞에 있는 수보다 크거나 같은 수 중 가장 작은 수로 만드는 것이 좋습니다.
- ✓ 주어진 수열을 앞에서부터 차례대로 보면서
- ✓ 바로 앞에 있는 수보다 크거나 같은 수 중 가장 작은 값이 되도록 연산을 수행하면 됩니다.
- ✓ 만약 i 와 $N - i + 1$ 모두 앞에 있는 수보다 크면 정렬할 수 없습니다.

QD. 영어 시험

ad_hoc

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: **장태환**, 19분
- ✓ 처음 푼 사람(Open Contest): **bnb2011**, 15분
- ✓ 출제자: 나정희

QD. 영어 시험

- ✓ 입력으로 주어진 문자열을 N 번 출력하면 됩니다.
- ✓ 이 문자열이 왜 조건을 만족하는 **가장 짧은** 문자열인지 고민해 보세요.

QE. 가장 긴 등차 부분 수열

dp

출제진 의도 – **Hard**

- ✓ 처음 푼 사람: **장태환**, 27분
- ✓ 처음 푼 사람(Open Contest): **jthis**, 16분
- ✓ 출제자: 박세훈

QE. 가장 긴 등차 부분 수열

- ✓ 설명의 편의를 위해 수열의 길이를 N , 수의 범위를 $X = 100$ 이라고 합시다.
- ✓ $O(NX \log X)$, $O(NX)$, $O(X^3)$ 등 여러 가지 풀이가 존재합니다.
- ✓ $O(NX \log X)$ 풀이와 $O(NX)$ 풀이를 차례로 설명하겠습니다.

QE. 가장 긴 등차 부분 수열

- ✓ $P(i, c)$ 를 i 번째 수보다 뒤에 있는 가장 앞에 있는 c 의 위치라고 정의합시다.
- ✓ P 배열은 i 가 큰 것부터 구하면 $O(NX)$ 시간에 전처리할 수 있습니다.
- ✓ 공차가 d 인 등차 부분 수열을 구하는 가장 간단한 방법은
수열의 모든 원소 A_i 에 대해, A_i 부터 시작해서 $A_i + d, A_i + 2d, \dots$ 를 찾는 것입니다.
- ✓ P 배열을 이용하면 **바로 다음 원소**를 찾는 작업을 $O(1)$ 시간에 수행할 수 있습니다.
- ✓ 그러므로 시작 지점 A_i 와 공차 d 가 주어졌을 때
- ✓ 등차 부분 수열의 길이에 비례하는 시간에 길이를 구할 수 있습니다.

QE. 가장 긴 등차 부분 수열

- ✓ 수의 범위는 최대 X 이기 때문에 공차가 d 인 등차 수열의 최대 길이는 X/d 입니다.
- ✓ $X/1 + X/2 + X/3 + \dots + X/X \approx O(X \log X)$ 이므로
- ✓ $O(X \log X)$ 시간에 어떤 지점에서 시작하는 가장 긴 등차 부분 수열을 구할 수 있습니다.
- ✓ N 개의 시작점에 대해 동일한 작업을 수행하면 $O(NX \log X)$ 에 문제를 해결할 수 있습니다.
- ✓ 공차가 $d = 0$ 인 경우는 따로 처리해야 합니다.

QE. 가장 긴 등차 부분 수열

- ✓ 공차가 d 이고 마지막 원소가 x 인 등차 부분 수열의 최대 길이를 $D(d, x)$ 라고 정의합시다.
- ✓ 주어진 수열을 앞에서부터 차례대로 보면, 점화식은 $D(d, A_i) \leftarrow D(d, A_i - d) + 1$ 입니다.
- ✓ 각 원소마다 $O(X)$ 개의 상태만 변경하므로 $O(NX)$ 시간에 정답을 구할 수 있습니다.

A. Gravity Hackenbush

game_theory

출제진 의도 – Medium

- ✓ 처음 푼 사람: 장준하엉덩이는빨개, 37분
- ✓ 처음 푼 사람(Open Contest): **dtc03012**, 5분
- ✓ 출제자: 이성현

A. Gravity Hackenbush

- ✓ 두 플레이어가 번갈아가며 그래프의 간선을 하나씩 지우는 게임입니다.
- ✓ 1번 플레이어는 빨간색과 초록색, 2번 플레이어는 파란색과 초록색 간선만 지울 수 있습니다.
- ✓ 지울 수 있는 간선이 없는 플레이어가 패배하게 됩니다.

A. Gravity Hackenbush

- ✓ 1번 플레이어는 빨간색 간선을 최대한 천천히 소모해야 합니다.
- ✓ 마찬가지로 2번 플레이어는 파란색 간선을 최대한 천천히 소모해야 합니다.
- ✓ 그러므로 두 플레이어 모두 초록색 간선을 먼저 지우는 것이 최적입니다.
- ✓ 초록색 간선을 먼저 지운 뒤, 빨간색과 파란색 간선을 지우면 됩니다.
- ✓ 지울 수 있는 간선이 없는 플레이어가 패자가 됩니다.

B. K-균형 잡힌 수

greedy

출제진 의도 – **Challenging**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 223분
- ✓ 처음 푼 사람(Open Contest): **hyperbolic**, 89분
- ✓ 출제자: 오주원

B. K-균형 잡힌 수

- ✓ 만약 주어진 수 X 가 K -균형 잡힌 수라면 X 를 출력하면 됩니다.
- ✓ 그렇지 않은 경우, X 보다 작은 수를 차례대로 확인해야 합니다.
- ✓ 정답의 길이가 X 보다 작다면 항상 $999 \dots 99$ 가 정답입니다.
- ✓ 그러므로 길이가 X 와 동일하면서 X 보다 작은 수 중에 정답이 있는지 확인하면 됩니다.

B. K-균형 잡힌 수

- ✓ X 보다 작은 수 Y 는 어떻게 생겼을까요?
- ✓ $X[i]$ 와 $Y[i]$ 가 처음으로 다른 지점 i 가 존재하고, $X[i] > Y[i]$ 를 만족합니다.
- ✓ i 이후에는 어떠한 숫자들이 오더라도 항상 $Y < X$ 입니다.
- ✓ 수의 첫 몇 자리가 고정된 상태에서 K -균형 잡힌 수를 만들 수 있는지 확인할 수 있다면
- ✓ X 의 뒷자리부터 1씩 감소시켜 가면서 확인하는 것을 통해 정답을 구할 수 있습니다.

B. K-균형 잡힌 수

- ✓ 만약 현재 수에 t 개의 숫자를 추가해서 K -균형 잡힌 수를 만들 수 있다면
- ✓ $K \geq 1$ 이기 때문에 $t + 1$ 개의 숫자를 추가해도 K -균형 잡힌 수를 만들 수 있습니다.
- ✓ Y 의 첫 n 자리가 주어졌을 때 K -균형 잡힌 수로 만들기 위해 필요한 최소 개수를 구합시다.
- ✓ 숫자 i 를 사용한 횟수를 C_i , 가장 많이 사용한 숫자의 사용 횟수를 mx 라고 합시다.
- ✓ 지금까지 사용한 숫자들의 사용 횟수를 모두 $mx - K$ 이상으로 만들어야 합니다.
- ✓ 그러므로 필요한 숫자의 최소 개수는 $\max(0, mx - K - C_i)$ 를 모두 더한 것입니다.
- ✓ 필요한 최소 개수가 (X 의 길이) – n 보다 작거나 같다면 K -균형 잡힌 수를 만들 수 있습니다.

B. K-균형 잡힌 수

- ✓ 문제의 정답을 구하는 것은 간단합니다.
- ✓ 먼저 X 의 뒷자리부터 차례대로 보면서 $X[i] > Y[i]$ 인 첫 지점 i 를 구합니다.
- ✓ 뒷부분을 최대화하는 것은 앞자리부터 차례대로 보면서 최대한 큰 숫자를 넣으면 됩니다.

C. Merge the Tree and Sequence

graphs, greedy

출제진 의도 – **Hard**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 163분
- ✓ 처음 푼 사람(Open Contest): **jthis**, 65분
- ✓ 출제자: 이성현

C. Merge the Tree and Sequence

- ✓ 한 구역의 점수은 $(A_1 + A_2 + \cdots + A_k)(B_1 + B_2 + \cdots + B_k)$ 꼴로 계산됩니다.
- ✓ 이는 $(A_1 + \cdots + A_k)B_1 + (A_1 + \cdots + A_k)B_2 + \cdots + (A_1 + \cdots + A_k)B_k$ 입니다.
- ✓ S_x 를 x 번째 구역에 속한 간선들의 끝점에 적힌 수의 합이라고 정의합시다.
- ✓ A'_i 를 i 번 정점에 달려있는 간선이 속한 구역들의 S_x 의 합이라고 정의합시다.
- ✓ 전체 점수는 $A'_1B_1 + A'_2B_2 + \cdots + A'_N B_N$ 꼴로 표현할 수 있습니다.

C. Merge the Tree and Sequence

- ✓ 점수의 최댓값은 A'_i 와 B_i 를 모두 오름차순 정렬해서 순서대로 대응시켜서 구할 수 있습니다.
- ✓ 최솟값은 A'_i 를 오름차순, B_i 를 내림차순 정렬해서 순서대로 대응시켜서 구할 수 있습니다.
- ✓ A'_i 를 구하는 것은 DFS, Union-Find 등을 이용해서 할 수 있습니다.

D. 바지 구매

math

출제진 의도 – **Easy**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 5분
- ✓ 처음 푼 사람(Open Contest): **kidw0124**, 34분
- ✓ 출제자: 나정휘

D. 바지 구매

- ✓ 바지는 $f(x) = u_i$ 를 만족하는 x 지점에서 시루의 하체에 걸리게 됩니다.
- ✓ 그러므로 $f(v_i) = u_i$ 인지 확인하면 됩니다.

E. 반전 수와 쿼리

ad_hoc

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: **<https://youtu.be/F09HNpeiQZ4>**, 84분
- ✓ 처음 푼 사람(Open Contest): **heeda0528**, 10분
- ✓ 출제자: 김준서

E. 반전 수와 쿼리

- ✓ 주어진 쿼리보다 조금 더 쉬운 문제를 풀어보겠습니다.
- ✓ 인접한 두 수 P_i 와 P_{i+1} 을 교환하면 반전 수는 어떻게 바뀔까요?
- ✓ $P_i < P_{i+1}$ 이면 반전 수가 1 증가하고, $P_i > P_{i+1}$ 이면 1 감소합니다.
- ✓ 즉, 인접한 두 수를 교환하면 항상 반전 수를 2로 나눈 나머지가 변합니다.

E. 반전 수와 쿼리

- ✓ 이제 두 수 P_l 과 P_r 을 교환하는 쿼리를 풀어봅시다.
- ✓ P_l 과 P_r 을 교환하는 작업은 다음과 같이 생각할 수 있습니다.
 1. 인접한 원소끼리 교환해서 P_r 을 P_l 바로 왼쪽으로 이동 ($r - l$ 번 교환)
 2. 인접한 원소끼리 교환해서 P_l 을 P_{r-1} 바로 오른쪽으로 이동 ($r - l - 1$ 번 교환)
- ✓ 두 수를 교환하는 것은 인접한 원소를 $2(r - l) - 1$ 번 교환하는 것이라 생각할 수 있습니다.
- ✓ 홀수 번 교환하는 것이므로 항상 반전 수를 2로 나눈 나머지가 변합니다.

E. 반전 수와 쿼리

- ✓ 마지막으로 구간 P_l, P_{l+1}, \dots, P_r 을 뒤집는 쿼리를 풀어봅시다.
- ✓ 구간을 뒤집는 작업은 다음과 같이 생각할 수 있습니다.
 1. P_l 과 P_r 교환
 2. P_{l+1} 과 P_{r-1} 교환
 3. ...
- ✓ 두 수를 교환하는 작업을 $\lfloor \frac{r-l+1}{2} \rfloor$ 번 수행합니다.
- ✓ 따라서 $\lfloor \frac{r-l+1}{2} \rfloor$ 이 홀수면 반전 수를 2로 나눈 나머지가 변합니다.
- ✓ 두 쿼리를 모두 $O(1)$ 에 처리할 수 있으므로 전체 문제를 $O(Q)$ 에 해결할 수 있습니다.

F. 시간딱딱충

math

출제진 의도 – **Hard**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 23분
- ✓ 처음 푼 사람(Open Contest): **parkky**, 110분
- ✓ 출제자: 김준서

F. 시간딱딱충

- ✓ 출발 시간이 늦어질수록 도착 시간도 단조 증가합니다.
- ✓ T 이후에 도착하는 가장 빠른 출발 시간 S 를 구한 다음
- ✓ S 시간에 출발했을 때 정확히 T 시간에 도착하는지 확인하면 됩니다.
- ✓ 임의의 출발 시간에 대한 도착 시간을 구할 수 있다면 이분 탐색을 이용해 S 를 구할 수 있습니다.
- ✓ 도착 시간을 $O(N)$ 에 구하는 방법을 알아봅시다.

F. 시간딱딱충

- ✓ 횡단보도를 건너기 시작하는 시간을 구하는 것을 제외하면 쉽게 해결할 수 있습니다.
- ✓ 횡단보도를 건너기 시작하는 시간을 $O(1)$ 에 구하는 방법을 알아봅시다.
- ✓ 아래 식을 만족하는 음이 아닌 정수 X 가 존재해야 t 시간에 i 번째 횡단보도를 건널 수 있습니다.

$$C_i + A_i X \leq t < t + D_i \leq C_i + A_i X + B_i$$

- ✓ 만약 t 시간에 횡단보도를 건널 수 있다면, $X = \lfloor (t - C_i) / A_i \rfloor$ 에서 위 식이 성립합니다.
- ✓ 위 식이 성립하지 않는 경우에는 횡단보도를 한 주기 만큼 기다려야 하므로 $X = \lfloor (t - C_i) / A_i \rfloor + 1$ 입니다.
- ✓ 단, $t < C_i$ 인 경우에는 횡단보도가 시작할 때까지 기다려야 합니다. ($t \leftarrow C_i$)

F. 시간딱딱총

- ✓ 횡단보도가 N 개 있으므로 임의의 출발 시간에 대한 도착 시간을 $O(N)$ 에 구할 수 있습니다.
- ✓ 이분 탐색의 탐색 범위는 $O(T)$ 이므로 전체 시간 복잡도는 $O(N \log T)$ 입니다.

G. 인공신경망

implementation, math

출제진 의도 – **Hard**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 67분
- ✓ 처음 푼 사람(Open Contest): **kidw0124**, 61분
- ✓ 출제자: 나정휘

G. 인공 신경망

- ✓ 매번 인공 신경망을 통째로 계산하면 $O(NMQ)$ 로 시간 초과를 받게 됩니다.
- ✓ 연산 횟수를 줄여야 합니다.
- ✓ 잘 생각해 보면 신경망 전체를 $O(NM)$ 시간에 인공 신경 하나로 합칠 수 있습니다.
- ✓ 신경망 전체를 하나의 인공 신경으로 합치면 매번 $O(NQ)$ 시간에 출력값을 계산할 수 있습니다.
- ✓ 전체 시간 복잡도는 $O(N(M + Q))$ 입니다.
- ✓ 여담으로, 이런 신경망은 선형 함수밖에 만들지 못하기 때문에
실제 신경망에서는 **활성화 함수**라는 것을 사용해 비선형 함수로 만듭니다.

H. 최대 최소공배수

math, number_theory

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 45분
- ✓ 처음 푼 사람(Open Contest): **tlsdydaud1**, 4분
- ✓ 출제자: 박세훈

H. 최대 최소공배수

- ✓ 최소 공배수가 커지기 위한 조건은 무엇이 있을까요?
 1. 되도록 큰 수를 선택해야 합니다.
 2. 선택한 수들은 서로소가 되어야 합니다.
- ✓ 이 조건을 만족시키면서 3개의 수를 선택하는 방법을 생각해 봅시다.

H. 최대 최소공배수

- ✓ 되도록 큰 수를 선택해야 하므로 일단 $N, N - 1, N - 2$ 를 선택해 봅시다.
- ✓ 연속한 세 수를 선택했기 때문에 $3, 4, 5, \dots$ 의 배수는 각각 최대 1개 존재합니다.
- ✓ 그러므로 짝수가 2개 이상 있는지만 확인하면 됩니다.
- ✓ 만약 N 이 홀수라면 $N - 1$ 만 짝수이므로 정답은 $N(N - 1)(N - 2)$ 입니다.
- ✓ 만약 N 이 짝수라면 N 과 $N - 2$ 가 모두 짝수이므로 둘 중 하나를 버려야 합니다.
- ✓ N 을 버리면 $(N - 1)(N - 2)(N - 3)$ 이고, $N - 2$ 를 버리면 $N(N - 1)(N - 3)$ 입니다.
- ✓ 만약 N 과 $N - 3$ 이 모두 3의 배수라면 정답은 $(N - 1)(N - 2)(N - 3)$ 입니다.
- ✓ 그렇지 않은 경우 정답은 $N(N - 1)(N - 3)$ 입니다.

I. 최장 최장 증가 부분 수열

dp

출제진 의도 – **Hard**

- ✓ 처음 푼 사람: <https://youtu.be/F09HNpeiQZ4>, 34분
- ✓ 처음 푼 사람(Open Contest): **tlsdydaud1**, 30분
- ✓ 출제자: 나정휘

I. 최장 최장 증가 부분 수열

- ✓ 최단 경로로 이동해야 하므로 오른쪽과 아래로만 이동할 수 있습니다.
- ✓ $D(i, j)$ 를 (i, j) 를 마지막 원소로 하는 최장 증가 부분 수열의 길이라고 정의합시다.
- ✓ 왼쪽 위부터 차례대로 $D(i, j) = \max_{r \leq i, c \leq j} D(r, c) + 1$ 를 계산하면 됩니다.
- ✓ 전체 시간 복잡도는 $O(N^4)$ 입니다.
- ✓ 여담으로, 이 문제는 $O(N^2 \log^2 N)$ 에 해결할 수 있습니다.

J. 행성 정렬

math, number_theory

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: **무지성 코딩**, 11분
- ✓ 처음 푼 사람(Open Contest): **pjshwa**, 8분
- ✓ 출제자: 김준서

J. 행성 정렬

- ✓ T_1, T_2, \dots, T_{N-2} 의 최소 공배수를 구하면 됩니다.
- ✓ 유clidean 호제법을 이용하면 두 수 a, b 의 최소 공배수를 $O(\log \max(a, b))$ 에 구할 수 있습니다.
- ✓ 따라서 전체 시간 복잡도는 $O(N \log 10^9)$ 입니다.
- ✓ 직접 소인수 분해를 해서 최소 공배수를 구하는 방법으로도 문제를 해결할 수 있습니다.
- ✓ 루트 시간에 소인수 분해를 하면 전체 시간 복잡도는 $O(N\sqrt{10^5})$ 입니다.
- ✓ 루트 시간에 소인수 분해를 하는 방법은 제3회 천하제일 코딩대회 본선 G번을 참고해 주세요.

- ✓ 예선/본선 문제의 모범 코드는

<https://github.com/justiceHui/Sunrin-Contest/tree/main/Sunrin-ICPC-2022>
에서 확인할 수 있습니다.

- ✓ 감사합니다.