

2023 SCON

Official Solutions

주최

- ✓ 송실대학교 IT대학

주관

- ✓ 컴퓨터학부 문제해결 소모임 SCCC

운영

- ✓ 박찬솔 chansol
- ✓ 나정휘 jhnah917

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

출제

- ✓ 나정휘 jhnah917
- ✓ 박찬솔 chansol
- ✓ 오주원 kyo20111
- ✓ 이성서 edenooo

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 소프트웨어학부

송실대학교 컴퓨터학부

검수

- ✓ 김준겸 ryute
- ✓ 김준서 junseo
- ✓ 白崎杏子 cologne

고려대학교 컴퓨터학과

한양대학교 컴퓨터소프트웨어학부

開拓團訓練所

대회 현장 스태프

- ✓ 김건중 9ada 송실대학교 컴퓨터학부
- ✓ 나세빈 you08045 송실대학교 컴퓨터학부
- ✓ 남혜민 nam14785 송실대학교 컴퓨터학부
- ✓ 박상철 eric2057 송실대학교 컴퓨터학부
- ✓ 박서은 pts0414 송실대학교 컴퓨터학부
- ✓ 신종원 shinjw4675 송실대학교 전자정보공학부
- ✓ 이계상 송실대학교 소프트웨어학부
- ✓ 진민성 m4ushold 송실대학교 소프트웨어학부
- ✓ 최원석 rudeore0928 송실대학교 컴퓨터학부

명찰/상장 디자인

- ✓ 권욱제 wookje

송실대학교 컴퓨터학부

문제지 및 풀이 슬라이드 제작

- ✓ 나정휘 jhnah917

송실대학교 컴퓨터학부

문제 일러스트

- ✓ 나정휘 jhnah917

송실대학교 컴퓨터학부

Sponsors

HYUNDAI
MOBIS

STARTLINK

문제	의도한 난이도	출제자
A 정보섬의 대중교통	Easy	박찬솔
B 팀명 정하기	Easy	나정휘
C 등차수열의 합	Easy	오주원
D 선택 정렬의 이동 거리	Medium	나정휘
E prlong longf	Medium	이성서
F 안전한 건설 계획	Medium	이성서
G Traveling SCCC President	Hard	나정휘
H SCCC 신입 부원 모집하기	Hard	박찬솔
I 산책과 퀴리	Challenging	이성서
J 아이템	Challenging	오주원

A. 정보섬의 대중교통

math, implementation

출제진 의도 - **Easy**

- ✓ 처음 푼 팀: **반드시 가야지요**, 2분
- ✓ 처음 푼 팀(Open Contest): **asdf1705**, 1분
- ✓ 출제자: 박찬솔

A. 정보섬의 대중교통

- ✓ $N \leq B$ 이므로 항상 지하철을 놓치지 않고 이용할 수 있습니다.
- ✓ 따라서 N 의 값은 정답에 영향을 미치지 않습니다.
- ✓ $A < B$ 이면 Bus, $A > B$ 이면 Subway, $A = B$ 이면 Anything을 출력하면 됩니다.

B. 팀명 정하기

math, implementation, sorting

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **반드시 가야지요**, 8분
- ✓ 처음 푼 팀(Open Contest): **asdf1705**, 4분
- ✓ 출제자: 나정휘

B. 팀명 정하기

- ✓ 첫 번째 팀명은 입학 연도를 오름차순으로 정렬하고 100으로 나눈 나머지를 출력하면 됩니다.
- ✓ 두 번째 팀명은 팀원을 문제 수 내림차순으로 정렬한 다음 성씨의 첫 글자를 출력하면 됩니다.
- ✓ 팀원이 항상 3명만 주어지기 때문에 조건문만 이용해도 문제를 해결할 수 있습니다.

C. 등차수열의 합

math

출제진 의도 - **Easy**

- ✓ 처음 푼 팀: 스콘빨리먹기대회우승팀, 5분
- ✓ 처음 푼 팀(Open Contest): **sedev57**, 2분
- ✓ 출제자: 오주원

C. 등차수열의 합

- ✓ 두 등차수열의 합은 등차수열이어야 합니다.
- ✓ 따라서 A 가 등차수열일 때만 $A_i = B_i + C_i$ 인 두 등차수열 B, C 가 존재합니다.
- ✓ B, C 를 만드는 것은 여러 방법이 있지만, $B_i = A_i, C_i = 0$ 으로 만드는 것이 가장 간단합니다.

D. 선택 정렬의 이동 거리

implementation, sorting

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **스콘빨리먹기대회우승팀**, 14분
- ✓ 처음 푼 팀(Open Contest): **sedev57**, 10분
- ✓ 출제자: 나정휘

D. 선택 정렬의 이동 거리

- ✓ 입력으로 주어지는 배열은 1 부터 N 까지의 수가 한 번씩 등장하는 배열입니다.
- ✓ 따라서 선택 정렬의 i 번째 단계에서는 i 가 A_i 로 이동합니다.
- ✓ 각 수의 위치를 저장하는 배열을 함께 관리하면 $O(N)$ 시간에 문제를 해결할 수 있습니다.

E. prlong longf

string, bruteforcing, dp

출제진 의도 - **Medium**

- ✓ 처음 푼 팀: **스콘빨리먹기대회우승팀**, 18분
- ✓ 처음 푼 팀(Open Contest): **xiaowuc1**, 13분
- ✓ 출제자: 이성서

E. prlong longf

- ✓ 두 가지 풀이가 존재합니다.
 1. 완전 탐색을 이용한 $O(2^{N/4})$ 풀이
 2. 동적 계획법을 이용한 $O(N)$ 풀이
- ✓ 차례대로 설명합니다.

E. prlong longf - 완전 탐색

- ✓ 풀이 1.
- ✓ 문자열에서 longlong 중 일부를 골라서 int로 변환하는 방법의 수를 구하는 문제입니다.
- ✓ $N \leq 80$ 이므로 long은 최대 $80/4 = 20$ 번 등장합니다.
- ✓ 정답이 $2^{20} = 1\,048\,576$ 이하로 충분히 작으므로, 모든 방법을 시도하는 완전 탐색으로 정답을 구할 수 있습니다.

E. prlong longf - 동적 계획법

- ✓ 풀이 2.
- ✓ `longlongdoublelonglong`처럼 분리되어 있는 `longlong...`은 서로 독립입니다.
- ✓ 따라서 연속한 `longlong...`마다 따로 경우의 수를 계산한 뒤 모두 곱하면 됩니다.
- ✓ $D[n]$ 을 `long`이 n 개 연속으로 붙어 있는 문자열을 복원하는 경우의 수라고 정의하면,
- ✓ $D[0] = D[1] = 1$,
- ✓ $D[n] = D[n - 1] + D[n - 2]$ 이라는 점화식이 성립합니다.
- ✓ 따라서 동적 계획법을 이용해 $O(N)$ 시간에 정답을 구할 수 있습니다.

F. 안전한 건설 계획

graphs, greedy

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **스콘빨리먹기대회우승팀**, 24분
- ✓ 처음 푼 팀(Open Contest): **xiaowuc1**, 18분
- ✓ 출제자: 이성서

F. 안전한 건설 계획

- ✓ 두 가지 풀이가 존재합니다.
 1. 그리디 기법을 이용한 $O(N^5)$ 풀이
 2. 그래프 탐색을 이용한 $O(N + M)$ 풀이
- ✓ 차례대로 설명합니다.

F. 안전한 건설 계획 - 그리디 기법

- ✓ 풀이 1.
- ✓ 삼각형을 적당한 순서로 추가해서 완전 그래프를 만드는 문제입니다.
- ✓ 비용이 적게 드는 연산부터 진행하는 그리디 전략을 이용해 해결할 수 있습니다.
- ✓ 완전 그래프가 될 때까지 아래 과정을 반복하면 됩니다.
 1. 비용이 0인 보강 작업을 할 수 있으면 아무거나 하나 찾아서 진행
 2. 그렇지 않은 경우, 비용이 1인 보강 작업을 아무거나 하나 찾아서 진행
- ✓ 매번 수행할 보강 작업을 $O(N^3)$ 시간에 찾을 수 있습니다.
- ✓ 보강 작업은 최대 $O(N^2)$ 번 진행하므로 전체 시간 복잡도는 $O(N^5)$ 입니다.
- ✓ $N \leq 40$ 으로 입력 크기가 작기 때문에 시간 제한 안에 문제를 해결할 수 있습니다.

F. 안전한 건설 계획 - 그래프 탐색

- ✓ 풀이 2.
- ✓ 정답의 상한과 하한을 찾아 봅시다.
- ✓ 만약 그래프가 연결 그래프라면, 0의 비용으로 완전 그래프를 만들 수 있습니다.
- ✓ 연결 요소가 여러 개라면, 간선이 하나 이상 있는 연결 요소 A 와 임의의 연결 요소 B 를 비용이 1인 보강 작업을 통해 연결할 수 있습니다.
- ✓ 따라서 정답은 (연결 요소의 개수) - 1 이하입니다.
- ✓ 비용이 0인 보강 작업은 그래프의 연결 요소의 개수를 줄일 수 없습니다.
- ✓ 비용이 1인 보강 작업은 그래프의 연결 요소의 개수를 최대 1만큼 줄일 수 있습니다.
- ✓ 따라서 정답은 (연결 요소의 개수) - 1 이상입니다.

F. 안전한 건설 계획 - 그래프 탐색

- ✓ 정답의 상한과 하한이 같으므로, 정답은 정확히 (연결 요소의 개수) - 1 입니다.
- ✓ DFS/BFS 등의 그래프 탐색 알고리즘을 이용하면 $O(N + M)$ 시간에 정답을 구할 수 있습니다.
- ✓ 이 풀이가 성립함을 이용해 그리디 기법 풀이의 정당성을 증명할 수 있습니다.

G. Traveling SCCC President

mst

출제진 의도 - **Hard**

- ✓ 처음 푼 팀: **스콘빨리먹기대회우승팀**, 55분
- ✓ 처음 푼 팀(Open Contest): **aeren**, 25분
- ✓ 출제자: 나정휘

G. Traveling SCCC President

- ✓ 순간 이동을 하지 않고 직접 이동해야 하는 간선을 생각해 봅시다.
- ✓ 이 간선들은 연결 그래프를 이뤄야 하므로 정답은 MST의 가중치보다 크거나 같습니다.
- ✓ 이제 가중치가 MST와 동일한 해가 존재함을 보일 것입니다.
- ✓ MST를 만든 다음 1번 정점에서 시작하는 DFS를 수행합니다.
- ✓ 방문하지 않은 정점으로 이동할 때 직접 이동하고 다시 돌아올 때 순간 이동을 하면
- ✓ 정확히 MST와 동일한 가중치로 모든 정점을 방문할 수 있습니다.
- ✓ 모든 정점을 한 번씩 방문한 이후에는 순간이동만을 사용해 시간의 소요 없이 모든 회의를 진행할 수 있습니다.
- ✓ MST는 프림 알고리즘이나 크루스칼 알고리즘 등을 이용해 $O(M \log M)$ 에 구할 수 있습니다.

H. SCCC 신입 부원 모집하기

dp_bitfield

출제진 의도 - **Hard**

- ✓ 처음 푼 팀: **스콘빨리먹기대회우승팀**, 140분
- ✓ 처음 푼 팀(Open Contest): **aeren**, 40분
- ✓ 출제자: 박찬솔

H. SCCC 신입 부원 모집하기

- ✓ 세 가지 풀이가 존재합니다.
 1. 동적 계획법을 이용한 $O(NKX^K)$ 풀이
 2. 최소 비용 최대 유량을 이용한 $O(N^3K + N^2K^2)$ 풀이
 3. 이분 그래프 최대 매칭을 이용한 $O(N^2KX)$ 풀이
- ✓ 차례대로 설명합니다.

H. SCCC 신입 부원 모집하기 - DP

- ✓ 점수가 높은 사람부터 차례대로 배정합니다.
- ✓ 각 스터디 그룹 i 의 정원이 C_i 만큼 남았을 때, 현재 인원의 상태를 $C = (C_1, C_2, \dots, C_K)$ 라는 배열로 표현할 수 있습니다.
- ✓ 아래 DP를 채운 다음, 배정 방법이 존재하는 $D[N][*]$ 를 아무거나 선택해서 출력하면 됩니다.
 - $D[i][C] = i$ 번째 사람까지 배정해서 인원의 상태가 C 가 되게 하는 최적의 배정이 존재하는가? 존재한다면 그러한 배정 방법을 아무거나 하나 저장한다.
- ✓ i 의 범위는 $[1, N]$ 이고 C 의 각 원소의 범위는 $[0, X]$ 범위이므로, C 는 $X + 1$ 진법을 이용해 0 이상 $(X + 1)^K$ 미만의 정수 값으로 표현할 수 있습니다.
- ✓ 따라서 상태의 개수는 총 $O(NX^K)$ 개입니다.
- ✓ 각 상태마다 $O(K)$ 개의 다음 상태가 있으므로 전체 시간 복잡도는 $O(NKX^K)$ 입니다.

H. SCCC 신입 부원 모집하기 - 최소 비용 최대 유량

- ✓ 문제에서 제시한 스터디 그룹 배정 알고리즘은 다음과 같은 성질을 갖고 있습니다.
 1. 배정되는 사람의 수를 최대화함
 2. 배정된 사람을 내림차순으로 정렬했을 때 사전 순 최대인 결과를 만들어 냄
- ✓ 점수가 i 번째로 높은 사람이 매칭되었을 때의 가중치를 -2^{N-i} 로 두면
- ✓ 사람을 최대한 많이 매칭시키면서 가중치를 최소화하는 문제가 됩니다.
- ✓ 따라서 이 문제는 최소 비용 최대 유량을 이용해 해결할 수 있습니다.
- ✓ 정점이 $O(N + K)$ 개, 간선이 $O(NK)$ 개, 최대 유량은 N 이므로 전체 시간 복잡도는 $O(N \times NK \times (N + K)) = O(N^3K + N^2K^2)$ 입니다.

H. SCCC 신입 부원 모집하기 - 이분 그래프 최대 매칭

- ✓ 최소 비용 최대 유량 풀이에서 사용한 성질을 그대로 이용합니다.
 1. 배정되는 사람의 수를 최대화함
 2. 배정된 사람을 내림차순으로 정렬했을 때 사전 순 최대인 결과를 만들어 냄
- ✓ DFS를 이용해 i 번째 정점을 매칭에 추가할 수 있는지 확인할 수 있습니다.
- ✓ 따라서 (배정된 사람 집합) + $\{i$ 번째 사람}의 배정이 가능한지 $O(NKX)$ 에 확인할 수 있습니다.
- ✓ 이 과정을 N 번 수행하므로 전체 시간 복잡도는 $O(N^2KX)$ 입니다.

I. 산책과 쿼리

graphs, disjoint_set

출제진 의도 – **Challenging**

- ✓ 처음 풀 팀: **스콘빨리먹기대회우승팀**, 110분
- ✓ 처음 풀 팀(Open Contest): **Lawali**, 22분
- ✓ 출제자: 이성서

1. 산책과 쿼리

- ✓ 어떤 자취방 u 의 산책 자유도가 높은지 판별하는 방법을 생각해 봅시다.
 - u 를 지나는 산책로가 있다면, 그 산책로를 왕복해서 모든 짝수를 만들 수 있음
 - u 에서 홀수 사이클로 갈 수 있다면, 아래 방법으로 충분히 큰 모든 홀수를 만들 수 있음
 1. u 에서 홀수 크기 사이클로 이동
 2. 홀수 크기 사이클을 따라서 한 바퀴 이동
 3. 다시 u 로 이동
 4. u 를 지나는 산책로 왕복
- ✓ 따라서 각 정점을 포함하는 컴포넌트에 홀수 사이클이 있는지 판별하면 됩니다.

I. 산책과 쿼리

- ✓ 각 정점 v 를 두 개의 정점 v_0 과 v_1 로 분할합니다.
- ✓ v_0, v_1 은 각각 짝수 시각과 홀수 시각에 v 에 위치함을 의미합니다.
- ✓ 만약 v_0 에서 v_1 로 이동할 수 있다면 v 는 산책의 자유도가 높은 정점입니다.
- ✓ 이러한 v 를 포함하는 연결 요소를 “홀수 연결 요소”라고 합니다.
- ✓ 홀수 연결 요소 안에 있는 모든 정점은 정답에 1씩 기여합니다.

I. 산책과 쿼리

- ✓ 쿼리로 두 정점의 번호 a, b 가 주어지면, a_0 과 b_1, a_1 과 b_0 을 연결합시다.
- ✓ 이후 a_0 과 a_1 이 연결되었는지 확인하고 정답을 갱신하면 됩니다.
- ✓ 이 작업은 각 집합의 크기를 관리하는 Union-Find를 이용해 빠르게 수행할 수 있습니다.
- ✓ 전체 시간 복잡도는 $O(N + Q \log N)$ 입니다.

J. 아이템

greedy, trie, dfs

출제진 의도 – **Challenging**

- ✓ 처음 풀 팀: **N/A**, N/A분
- ✓ 처음 풀 팀(Open Contest): **Lawali**, 49분
- ✓ 출제자: 오주원

J. 아이템

- ✓ 아이템을 k 개 획득하면 2^k 씩 이동할 수 있습니다.
- ✓ 즉, 2^k 로 나눈 나머지가 같은 좌표로만 이동할 수 있습니다.
- ✓ 어떤 정수를 2^k 로 나눈 나머지는 이진법으로 나타내었을 때 하위 k 비트와 동일합니다.
- ✓ 따라서 $2^{k-1}, 2^{k-2}, \dots, 2^0$ 을 나타내는 비트가 모두 같은 좌표로만 이동할 수 있습니다.

J. 아이템

- ✓ 아이템의 좌표를 하위 비트부터 차례대로 삽입해서 이진 트라이(Binary Trie)를 구축합니다.
- ✓ 아이템을 획득해서 이동 거리가 2배가 되는 것은,
- ✓ 트라이에서 해당 아이템의 좌표가 저장된 서브트리로 한 칸 이동한 다음,
- ✓ 서브트리 안에 있는 아이템만 추가로 획득할 수 있는 것과 동일합니다.

- ✓ 트라이의 각 정점에서, 그 정점을 루트로 하는 서브트리 안에 있는 아이템의 개수를 저장합니다.
- ✓ 트라이 위에서 DFS를 이용해 정답을 구할 것입니다.

J. 아이템

- ✓ DFS에서 자식 정점으로 내려가면 더이상 반대편 자식의 서브트리에 있는 아이템을 얻지 못하므로, 내려가는 쪽의 서브트리 반대편에 있는 아이템을 최대한 사용하는 것이 이득입니다.
- ✓ DFS를 하면서 각 정점에 도달할 때마다, 그 정점의 서브트리에서 선택할 예정인 아이템의 개수 t 를 관리합니다.
- ✓ 자식 정점으로 내려갈 때마다 t 를 반대편 자식의 서브트리와 최대한 많이 매칭시키는 것이 이득입니다.
- ✓ 따라서, 반대편 서브트리에 아이템이 s 개 있다면 $\min(t, s)$ 개는 반대쪽과 매칭시키고, 내려갈 서브트리에서는 (남은 $t - \min(t, s)$ 개) + (다음으로 먹을 아이템 1개)를 새로운 t 값으로 설정하고 재귀적으로 진행하면 됩니다.
- ✓ 트라이의 정점은 $O(N \log X)$ 개이므로 전체 시간 복잡도는 $O(N \log X)$ 입니다. 이때 $X = 10^{18}$ 입니다.

- ✓ 대회 문제의 모범 코드는 <https://sccc.kr/scon/2023> 에서 확인할 수 있습니다.
- ✓ 감사합니다.