

2024 SCON

Official Solutions

주최

- ✓ 송실대학교 IT대학

주관

- ✓ 컴퓨터학부 문제해결 소모임 SCCC

후원

- ✓ 현대모비스
- ✓ 퓨리오사AI
- ✓ Jane Street
- ✓ 스타트링크

운영&출제

- | | | | |
|---|-----|------------|---------------|
| ✓ | 김성주 | rla777 | 숭실대학교 컴퓨터학부 |
| ✓ | 나정휘 | jhnah917 | 숭실대학교 컴퓨터학부 |
| ✓ | 박찬솔 | chansol | 숭실대학교 컴퓨터학부 |
| ✓ | 유상원 | sangwon090 | 숭실대학교 컴퓨터학부 |
| ✓ | 오주원 | kyo20111 | 숭실대학교 소프트웨어학부 |
| ✓ | 진세림 | qwerty1120 | 숭실대학교 컴퓨터학부 |

검수

- ✓ 김영현 kipa00
- ✓ 윤교준 yclock
- ✓ 白崎杏子 cologne

서울대학교 컴퓨터공학부

서울대학교 컴퓨터공학부

開拓団訓練所

테스트

- ✓ 우민규 mingyu331
- ✓ 이지언 ez_code
- ✓ 진민성 m4ushold

서울과학고등학교

연세대학교 아시아학과

숭실대학교 소프트웨어학부

포스터, 명찰, 상장 디자인

✓ 박수현 shiftpsh

서강대학교 컴퓨터공학과

문제지, 풀이 슬라이드 제작

✓ 김영현 kipa00

서울대학교 컴퓨터공학부

✓ 나정휘 jhnah917

숭실대학교 컴퓨터학부

✓ 박수현 shiftpsh

서강대학교 컴퓨터공학과

대회 현장 스태프

- ✓ 길수민 2093ab 숙명여자대학교 소프트웨어학부
- ✓ 김영현 kipa00 서울대학교 컴퓨터공학부
- ✓ 박경욱 sk091204091204 연세대학교 인공지능학과
- ✓ 우민규 mingyu331 서울과학고등학교
- ✓ 이지언 ez_code 연세대학교 아시아학과
- ✓ 정은채 celina324 이화여자대학교 컴퓨터공학과
- ✓ 조다니엘 arduinocc04 서강대학교 컴퓨터공학과
- ✓ 진민성 m4ushold 숭실대학교 소프트웨어학부
- ✓ 천민재 open_mind 홍익대학교 캠퍼스자율전공(서울)
- ✓ 운영&출제진 모두

Sponsors



Sponsors



문제	의도한 난이도	출제자
A 과민성 대장 증후군	Easy	유상원
B 팀명 정하기 2	Easy	나정휘
C 온데간데없을뿐더러	Easy	나정휘
D 미로 탈출	Medium	오주원
E 수식 고치기	Medium	박찬솔
F 피보나치 기념품	Medium	나정휘
G 시간표 만들기	Hard	박찬솔
H 아이템 2	Hard	오주원
I 불꽃놀이의 아름다움	Hard	박찬솔
J Traveling SCCC President 2	Hard	오주원

A. 과민성 대장 증후군

implementation, simulation

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **세스콘**, 1분
- ✓ 처음 푼 팀 (Open Contest): **riroan**, 0분
- ✓ 출제자: 유상원

A. 과민성 대장 증후군

- ✓ i 번째 날까지 누적된 스트레스 양을 S_i 라고 하면, $S_i = \max(S_{i-1} + A_i, 0)$ 입니다.
- ✓ $1 \leq i \leq N$ 이면서 $S_i \geq M$ 을 만족하는 i 의 개수를 구하면 됩니다.

B. 팀명 정하기 2

string, implementation

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **떨거지들**, 6분
- ✓ 처음 푼 팀(Open Contest): **riroan**, 2분
- ✓ 출제자: 나정휘

B. 팀명 정하기 2

- ✓ 팀명 S 에 등장하는 알파벳 소문자, 대문자, 숫자의 개수를 각각 L, U, D 라고 합시다.
- ✓ 아래 세 가지 조건을 모두 만족하는 팀명을 찾아서 출력하면 됩니다.
 - 대문자가 소문자보다 더 많이 등장하면 안 된다. ($U \leq L$)
 - 팀명은 10글자 이하로 지어야 한다. ($|S| \leq 10$)
 - 숫자가 아닌 글자가 하나 이상 포함되어 있어야 한다. ($D < |S|$)

C. 온데간데없을뿐더러

math, arithmetic

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **세스콘**, 4분
- ✓ 처음 푼 팀 (Open Contest): **nflight11**, 1분
- ✓ 출제자: 나정휘

C. 온데간데없을뿐더러

- ✓ N 개의 양의 정수 A_1, A_2, \dots, A_N 이 주어지면, 이들을 모두 이어 붙인 수를 구해야 합니다.
- ✓ $L_i = \lfloor \log_{10} A_i \rfloor + 1$ 이라고 정의합시다.
- ✓ A_1, A_2, \dots, A_i 를 이어붙인 결과 S_i 는 $S_i = S_{i-1} \times 10^{L_i} + A_i$ 를 이용해 계산할 수 있습니다.
- ✓ L_i 는 다양한 방법으로 구할 수 있습니다.
 - $\log_{10}()$ 함수 사용
 - 문자열로 변환한 뒤 길이 계산
 - $1 \leq A_i \leq 99$ 인 점을 이용해 $(A[i] < 10 ? 1 : 2)$ 로 계산

D. 미로 탈출

case_work

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: ↑ ↑ 곧 따라잡힐 예정, 31분
- ✓ 처음 푼 팀(Open Contest): **golazcc83**, 13분
- ✓ 출제자: 오주원

D. 미로 탈출

- ✓ $S < E$ 일 때의 풀이만 작성합니다.
- ✓ $S > E$ 이면 두 값을 뒤집은 다음 같은 방법으로 해결할 수 있습니다.

D. 미로 탈출

- ✓ S 와 E 를 기준으로 5개의 구역으로 나눌 수 있습니다.
 1. $A \rightarrow [1, S - 1]$
 2. $S \rightarrow [S, S]$
 3. $B \rightarrow [S + 1, E - 1]$
 4. $E \rightarrow [E, E]$
 5. $C \rightarrow [E + 1, N]$
- ✓ S 와 E 의 값에 따라서 A, B, C 는 존재하지 않을 수 있습니다.
- ✓ A, B, C 의 존재 여부에 따른 $2^3 = 8$ 가지 상황을 모두 생각해 봅시다.
- ✓ 8가지 상황 모두 중복 방문 없이 $N - 1$ 번의 이동/순간 이동으로 탈출하는 방법이 존재합니다.
- ✓ 따라서 중복 방문을 하지 않으면서 순간 이동의 횟수를 최소화하는 방법을 찾으면 됩니다.

D. 미로 탈출

- ✓ A, B, C 가 존재하지 않는 경우
 - $N = 2$ 일 때만 가능하며, 비용을 소모하지 않고 탈출할 수 있습니다.
- ✓ A, C 가 존재하지 않는 경우
 - $S = 1, E = N$ 일 때만 가능하며, 비용을 소모하지 않고 탈출할 수 있습니다.
- ✓ B 가 존재하지 않는 경우 / B, C 가 존재하지 않는 경우
 - E 를 마지막에 방문하기 위해 A 로 먼저 이동하면, 순간 이동 없이 E 로 돌아갈 수 없습니다.
 - S 에서 1로 이동한 뒤, N 으로 순간 이동하고 E 로 이동하면 비용을 최소화할 수 있습니다.
 - 1의 비용으로 탈출할 수 있습니다.

D. 미로 탈출

✓ A, B 가 존재하지 않는 경우

- $S = 1, E = 2, N > 2$ 인 상황입니다.
- 순간 이동 없이 S 에서 E 를 거치지 않고 C 로 이동할 수 없습니다.
- 따라서 N 으로 순간 이동한 다음 E 로 이동하면 비용을 최소화할 수 있습니다.
- 1의 비용으로 탈출할 수 있습니다.

✓ A 가 존재하지 않는 경우

- 마찬가지로, 순간 이동 없이 S 에서 E 를 거치지 않고 C 로 이동할 수 없습니다.
- 1에서 $E - 1$ 로 이동한 뒤, N 으로 순간 이동한 다음 E 까지 이동하면 탈출할 수 있습니다.
- 1의 비용으로 탈출할 수 있습니다.

D. 미로 탈출

- ✓ A, B, C 가 모두 존재하는 경우
 - B 구역에 진입하기 위해서는 반드시 S 또는 E 를 거쳐야 합니다.
 - 따라서 처음에 B 로 이동해야 중복 방문을 방지할 수 있습니다.
 - 이후 A, C 를 방문해야 하는데, 구역을 이동할 때는 순간 이동을 사용해야만 합니다.
 - 따라서 최소 2번의 순간 이동이 필요하고, 항상 2번의 순간 이동으로 모두 방문할 수 있습니다.
 1. S 에서 $E - 1$ 로 이동해서 B 모두 방문
 2. 1로 순간 이동한 다음 $S - 1$ 로 이동해서 A 모두 방문
 3. N 으로 순간 이동한 다음 E 로 이동해서 C 모두 방문 후 탈출
- ✓ C 가 존재하지 않는 경우
 - 같은 방법으로 2번의 순간 이동으로 중복 방문 없이 탈출할 수 있습니다.

D. 미로 탈출

- ✓ 8가지 상황을 모두 정리하면, 다음과 같은 결론을 얻을 수 있습니다.
1. S 와 E 가 양 끝에 있으면 정답은 0
 2. S 가 끝에 있거나 S 와 E 가 인접한 위치에 있으면 정답은 1
 3. (1), (2)에 해당하지 않으면 정답은 2

E. 수식 고치기

ad_hoc, dp

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **떨거지들**, 37분
- ✓ 처음 푼 팀(Open Contest): **angelo10**, 18분
- ✓ 출제자: 박찬솔

E. 수식 고치기

- ✓ A_{N-1} 과 A_N 만 바꾸더라도 항상 T와 F를 만들 수 있습니다.
 - $A_{N-1} = \&, A_N = F \rightarrow$ 수식의 결과: F
 - $A_{N-1} = |, A_N = T \rightarrow$ 수식의 결과: T
- ✓ 따라서 답은 2 이하이고, 정답이 0 또는 1이 될 수 있는지 확인하면 됩니다.
- ✓ 어떤 A_i 하나를 바꿨을 때 원하는 결과가 나온다면 답은 1입니다.
- ✓ $N + 1$ 개의 식을 각각 $O(N)$ 시간에 평가할 수 있으므로 전체 시간 복잡도는 $O(N^2)$ 입니다.

E. 수식 고치기

- ✓ **별해:** $D(i, j) = A_1, \dots, A_i$ 의 결과를 $j \in \{T, F\}$ 로 만들기 위해 필요한 최소 변경 횟수
- ✓ 시간 복잡도: $O(N)$

F. 피보나치 기념품

math, ad_hoc, constructive, case_work

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **떨거지들**, 32분
- ✓ 처음 푼 팀(Open Contest): **nflight11**, 16분
- ✓ 출제자: 나정휘

F. 피보나치 기념품

- ✓ 피보나치 수열은 다음과 같이 정의되는 수열입니다.
 - $F_1 = F_2 = 1$
 - $F_n = F_{n-1} + F_{n-2}$ (단, $n \geq 3$)
- ✓ $N \equiv 1 \pmod{3}$ 이면 F_1 를 제외한 나머지, 그렇지 않으면 모든 기념품을 나눠줄 수 있습니다.

F. 피보나치 기념품

- ✓ **관찰:** 길이가 3의 배수인 F 의 연속한 부분 수열은 항상 합이 같게 분배할 수 있습니다.
 - $b - a + 1$ 이 3의 배수일 때 F 의 연속한 부분 수열 F_a, F_{a+1}, \dots, F_b 를 생각해 봅시다.
 - b 는 $a + 3k + 2$ 꼴이고, 피보나치 수열의 정의에 의해 $F_{a+3k} + F_{a+3k+1} = F_{a+3k+2}$ 입니다.
 - $F_{a+2}, F_{a+5}, \dots, F_b$ 와 나머지로 나누면 합이 같게 분배할 수 있습니다.
- ✓ 따라서 $N \equiv 0 \pmod{3}$ 이면 항상 N 개의 기념품을 나눠줄 수 있습니다.

F. 피보나치 기념품

- ✓ $N \equiv 2 \pmod{3}$ 이면 F_3, F_4, \dots, F_N 을 합이 동일하게 분배할 수 있습니다.
- ✓ 또한, $F_1 = F_2 = 1$ 이므로 하나씩 나눠주면 N 개의 기념품을 모두 나눠줄 수 있습니다.
- ✓ $N \equiv 1 \pmod{3}$ 이면 F_2, F_3, \dots, F_N 을 합이 동일하게 분배할 수 있습니다.
- ✓ 따라서 $F_2 + \dots + F_N$ 은 짝수인데, $F_1 = 1$ 은 홀수이므로 N 개를 모두 나눠줄 수 없습니다.
- ✓ F_1 을 제외한 $N - 1$ 개를 나눠주면 항상 나눠주는 기념품의 개수를 최대화할 수 있습니다.

G. 시간표 만들기

bruteforcing, implementation

출제진 의도 – **Hard**

- ✓ 처음 푼 팀: **떨거지들**, 77분
- ✓ 처음 푼 팀(Open Contest): **choisang0826**, 29분
- ✓ 출제자: 박찬솔

G. 시간표 만들기

- ✓ $K = \sum A_i$ 가 15 이하이므로 2^K 가지 경우를 모두 확인하는 풀이를 시도해 볼 수 있습니다.
- ✓ 각 경우마다 아래 세 가지 조건을 확인하면 됩니다.
 - 각 그룹에서 1개 이하의 과목을 선택했는가?
 - 강의 시간이 겹치는 과목이 없는가?
 - 선택한 강의의 학점 합이 22인가?

H. 아이템 2

dp

출제진 의도 – **Hard**

- ✓ 처음 푼 팀: **떨거지들**, 128분
- ✓ 처음 푼 팀(Open Contest): **lobo_prix**, 38분
- ✓ 출제자: 오주원

H. 아이템 2

- ✓ 아이템은 다음과 같은 형태로만 가져갈 수 있습니다.
 1. 길이가 K 이상인 구간
 2. 1 번 또는 N 번 아이템을 포함하는 경우 길이 K 미만인 구간도 가능
 3. 선택한 구간은 모두 서로 겹치지 않음
- ✓ A 의 앞과 뒤에 0을 K 개씩 추가해서 (2)번 조건을 무시하고 문제를 해결합니다.

H. 아이템 2

- ✓ $D(i) := A_1, A_2, \dots, A_i$ 를 적당히 가져갔을 때 가치의 최댓값이라고 정의합니다.
- ✓ A_i 를 가져가지 않으면 $D(i) \leftarrow D(i - 1)$ 입니다.
- ✓ A_i 를 가져가면 K 개 이상 가져가야 하므로 $D(i) \leftarrow \max_{j < i-K} \left\{ D(j) + \sum_{t=j+1}^i A_t \right\}$ 입니다.
- ✓ 이 점화식을 그대로 구현하면 $O(N^2)$ 또는 $O(N^3)$ 이므로 최적화가 필요합니다.

H. 아이템 2

- ✓ A 의 누적 합 배열 $S_i = \sum_{j=1}^i A_j$ 를 정의합니다.
- ✓ 점화식을 $D(i) \leftarrow \max_{j < i-K} \{D(j) + S_i - S_j\}$ 로 다시 작성할 수 있습니다.
- ✓ max 함수에서 j 와 관계없는 항을 밖으로 빼면 $D(i) \leftarrow \max_{j < i-K} \{D(j) - S_j\} + S_i$ 가 됩니다.
- ✓ 즉, $D(j) - S_j$ 의 최댓값을 효율적으로 계산할 수 있으면 문제를 빠르게 해결할 수 있습니다.

H. 아이템 2

- ✓ $M(i) := \max_{j \leq i} \{D(j) - S_j\}$ 라고 정의합시다.
- ✓ 최종 점화식은 $D(i) = \max \{D(i-1), M(i-K) + S_i\}$ 입니다.
- ✓ $D(i)$ 를 구하면 $M(i) = \max \{M(i-1), D(i) - S_i\}$ 를 이용해 M 을 갱신할 수 있습니다.
- ✓ 따라서 전체 문제를 $O(N)$ 시간에 해결할 수 있습니다.

I. 불꽃놀이의 이름다움

trees, graph_traversal, dp_tree

출제진 의도 – **Hard**

- ✓ 처음 풀 팀: **N/A**, N/A분
- ✓ 처음 풀 팀(Open Contest): **dkim110807**, 28분
- ✓ 출제자: 박찬솔

I. 불꽃놀이의 아름다움

- ✓ 스위치를 설치하는 N 가지 방법에 대해 매번 계산하면 $O(N^2)$ 이 되어 시간 초과를 받습니다.
- ✓ 이전에 계산한 **폭죽의 아름다움** 값을 재사용해서 중복 계산을 피하는 방법을 생각해 봅시다.
- ✓ 처음 스위치를 설치한 정점을 a 라고 합시다.
- ✓ a 를 기준으로 나뉘지는 컴포넌트 C_i 에 대해, 아래 정보를 저장합니다.
 - $weights(a, i) = C_i$ 에 속한 모든 정점 c 의 W_c 값의 합
 - 단, i 는 C_i 에 속한 정점 중 a 와 인접한 정점의 번호

I. 불꽃놀이의 이름다움

- ✓ 이제 a 와 인접한 정점 b 에 스위치를 설치하려고 합니다.
- ✓ 정점 b 를 포함하는 컴포넌트 C_b 와 그렇지 않은 컴포넌트 C_i 로 나눠서 생각해 봅시다.
- ✓ Case 1. 정점 b 를 포함하는 컴포넌트 C_b :
 - C_b 에 속한 모든 정점 c 에 대해, $D(a, c)$ 가 1씩 **감소**합니다.
 - 따라서 스위치가 a 에 있을 때의 답에서 $weights(a, b)$ 를 빼야 합니다.
- ✓ Case 2. C_b 를 제외한 나머지 컴포넌트 C_i :
 - C_i 에 속한 모든 정점 c 에 대해, $D(a, c)$ 가 1씩 **증가**합니다.
 - 따라서 스위치가 a 에 있을 때의 답에서 $weights(a, i)$ 를 더해야 합니다.
- ✓ a 에 폭죽이 새로 설치되므로, 스위치가 a 에 있을 때의 답에서 W_a 를 더해야 합니다.

I. 불꽃놀이의 이름다움

- ✓ 인접한 두 정점 a, b 에 대해, $weights(a, b)$ 의 값은 DFS 한 번으로 모두 구할 수 있습니다.
- ✓ 정점 a 에 스위치를 설치한 후, 정점 b 로 스위치를 옮길 때 답은 $O(1)$ 에 갱신할 수 있습니다.
- ✓ 따라서, DFS 두 번으로 $O(N)$ 시간에 전체 문제를 해결할 수 있습니다.

J. Traveling SCCC President 2

graph_traversal, bitmask

출제진 의도 - **Hard**

- ✓ 처음 풀 팀: **N/A**, N/A분
- ✓ 처음 풀 팀(Open Contest): **dkim110807**, 18분
- ✓ 출제자: 오주원

J. Traveling SCCC President 2

- ✓ $2^{60} - 1$ 시간을 소요하면 모든 간선을 사용할 수 있으므로 항상 N 번 정점에 도달할 수 있습니다.
- ✓ 소요 시간 X 가 주어졌을 때 N 번 정점에 도달 가능한지 판별할 수 있을까요?

J. Traveling SCCC President 2

- ✓ 전체 소요 시간이 X 이면 $w_i \wedge X = w_i$ 인 간선을 사용할 수 있습니다.
- ✓ 따라서 위 조건을 만족하는 간선만 사용하는 $1 \rightarrow N$ 경로가 존재하는지 확인하면 됩니다.
- ✓ 두 정점의 연결성 판별은 DFS, BFS 등을 이용해 $O(N + M)$ 시간에 해결할 수 있습니다.

J. Traveling SCCC President 2

- ✓ 소요 시간을 최소화해야 하므로 되도록 높은 비트를 꺼진 상태로 만들어야 합니다.
- ✓ 따라서 2^{59} 부터 2^0 까지 하나씩 k 번째 비트를 꺼보면서 문제의 정답이 될 수 있다면 유지, 정답이 될 수 없다면 다시 켜는 것으로 문제를 해결할 수 있습니다.
- ✓ 전체 시간 복잡도는 $O((N + M) \log W)$ 입니다.

- ✓ 대회 문제의 모범 코드는 <https://sccc.kr/scon/2024> 에서 확인할 수 있습니다.
- ✓ 감사합니다.