

# 2025 SCON

## Official Solutions

## 주최

- ✓ 송실대학교 IT대학

## 주관

- ✓ 컴퓨터학부 문제해결 소모임 SCCC

## 운영

- ✓ 조문성 jms020820
- ✓ 진민성 m4ushold
- ✓ 나정휘 jhnah917
- ✓ 김서윤 moonlit

송실대학교 컴퓨터학부

송실대학교 소프트웨어학부

Quora

송실대학교 컴퓨터학부

## 출제

- ✓ 김서윤 moonlit
- ✓ 김한결 rkaskan506
- ✓ 박근형 w8385
- ✓ 이창민 chleee
- ✓ 조문성 jms020820
- ✓ 진민성 m4ushold
- ✓ 최도현 dohyonne

송실대학교 컴퓨터학부

송실대학교 컴퓨터학과

무소속

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 소프트웨어학부

송실대학교 컴퓨터학부

## 검수

- ✓ 강혜아 cologne
- ✓ 김준겸 ryute
- ✓ 김준서 junseo
- ✓ 나정휘 jhnah917
- ✓ 오주원 kyo20111
- ✓ 정현서 jhwest2

NEXON KOREA

고려대학교 컴퓨터학과

채널코퍼레이션

Quora

삼성전자

서울대학교 컴퓨터공학부

## 포스터/문제지 표지/명찰 디자인

- ✓ 송은서 eunseo03

송실대학교 컴퓨터학부

## 대회 현장 스태프

- ✓ 강민희 mingnei
- ✓ 김민성 minseongid
- ✓ 김성주 rla777
- ✓ 이서진 seojin329
- ✓ 이시온 siooooooni
- ✓ 진세림 qwerty1120
- ✓ 최명일 fdsa313

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

송실대학교 컴퓨터학부

## Sponsors



문제	의도한 난이도	문제 아이디어	문제 작업
<b>A</b> 알파벳 블록	<b>Easy</b>	최도현	최도현
<b>B</b> 합의 최소	<b>Easy</b>	진민성	진민성
<b>C</b> 특별상 눈치게임	<b>Easy</b>	최도현	박근형
<b>D</b> N거리 건너기	<b>Medium</b>	김한결	김한결
<b>E</b> even하게 익은 SCON	<b>Medium</b>	조문성	조문성
<b>F</b> $A = B \oplus C$	<b>Medium</b>	김서윤	김서윤
<b>G</b> 불꽃놀이의 아름다움 2	<b>Hard</b>	진민성	박근형
<b>H</b> 서로소 조합	<b>Hard</b>	최도현	이창민
<b>I</b> 대결	<b>Hard</b>	조문성	조문성
<b>J</b> 맛있는 스콘 만들기	<b>Challenging</b>	최도현	최도현

# A. 알파벳 블록

math

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **JJJ**, 2분
- ✓ 문제 아이디어: 최도현
- ✓ 문제 작업: 최도현



## A. 알파벳 블록

- ✓ 상자 1개에는 S가 2개, C가 4개, O가 1개, N이 1개 필요합니다.
- ✓ 서로 교환 가능한 블록끼리는 하나의 블록으로 통일해서 생각하면 쉽습니다.
- ✓ N과 O를 각각 S와 C로 교환하면, S 3개와 C 6개를 이용해 상자를 구성할 수 있습니다.
- ✓ 따라서 정답은  $\min(\lfloor \frac{b_S + b_N}{3} \rfloor, \lfloor \frac{b_C + 2b_O}{6} \rfloor)$  이 됩니다.

## B. 합의 최소

math

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **말안되네**, 5분
- ✓ 문제 아이디어: 진민성
- ✓ 문제 작업: 진민성

## B. 합의 최소

- ✓  $i$  번째 수로 가능한 가장 작은 값은  $\min(A_i, A_{i+1}, \dots, A_N)$  입니다.
- ✓ 따라서 수열  $B_i = \min(A_i, A_{i+1}, \dots, A_N)$  를 빠르게 구하면 문제를 해결할 수 있습니다.
- ✓  $i$  를  $N - 1$  부터 1 까지 순회하면서  $B_i = \min(B_{i+1}, A_i)$  를 계산하면 됩니다.

## C. 특별상 눈치게임

bruteforcing

출제진 의도 – **Easy**

- ✓ 처음 푼 팀: **TeamRAM**, 26분
- ✓ 문제 아이디어: 최도현
- ✓ 문제 작업: 박근형

### C. 특별상 눈치게임

- ✓ 정수  $i$  를 고른 팀의 개수를  $cnt[i]$  라고 합시다.
- ✓ 어떤 조합  $(x, y, z)$  를 고르는 것은  $cnt[x]$  와  $cnt[y]$  와  $cnt[z]$  에 각각 1 씩 더한 것과 같습니다.
- ✓  $cnt$  배열을 뒤에서부터 보면서, 처음으로  $cnt[i] = 1$  인  $i$  를 고른 팀이 특별상을 받게 됩니다.
- ✓ 이 과정에서 필요한 연산의 횟수는 최대 100번입니다.
- ✓ 정수 3개를 고르는 방법의 수는  $C = \frac{100 \times 99 \times 98}{6} \approx 1.6 \times 10^5$  입니다.
- ✓ 따라서 가능한  $C$  가지 방법을 모두 순회하면서, 특별상을 받을 수 있는지 확인하면 됩니다.
- ✓ 총 연산 횟수는  $100C \approx 1.6 \times 10^7$  정도이므로, 제한 시간 안에 정답을 구할 수 있습니다.

## D. N거리 건너기

implementation

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **1학년1학기ps생초보3인큐**, 57분
- ✓ 문제 아이디어: 김한결
- ✓ 문제 작업: 김한결

## D. N거리 건너기

- ✓ 시계 방향과 반시계 방향으로 이동할 때의 소요 시간을 각각 구한 뒤 비교하면 됩니다.
- ✓ 문제에서 주어진 수열  $A$ 에서  $x$  번 횡단보도가 등장하는 위치를  $P_x$ 라고 합니다.
- ✓ 즉,  $P_x$ 는  $x$  번 횡단보도의 초록불이 켜지는 순서를 의미합니다.

## D. N거리 건너기

- ✓ 바로 이전에 건너 횡단보도의 번호를  $i$ , 다음에 건너야 하는 횡단보도의 번호를  $j$  라고 합시다.
- ✓  $P_i < P_j$  라면  $P_j - P_i$  초 뒤에 다음 횡단보도를 건널 수 있습니다.
- ✓  $P_i > P_j$  라면  $A_{P_i+1}, A_{P_i+2}, \dots, A_N, A_1, \dots, A_{P_j}$  번 신호등을 차례로 기다려야 하므로  $N - (P_i - P_j)$  초 뒤에 건널 수 있습니다.
- ✓ 대기 시간을 매번 상수 시간에 계산할 수 있으므로 전체 시간 복잡도는  $O(N)$  입니다.



## E. even하게 익은 SCON

dp, mathematics

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **EME**, 21분
- ✓ 문제 아이디어: 조문성
- ✓ 문제 작업: 조문성

## E. even하게 익은 SCON - 동적 계획법

- ✓ 문자를 한 글자씩 추가하는 방식으로 문자열을 구성한다고 생각해 봅시다.
- ✓  $S$ 와  $C$ 의 실제 개수는 중요하지 않고, 홀짝 여부만 중요하다는 것을 알 수 있습니다.
- ✓ 그러므로 다음과 같은 점화식을 생각해 볼 수 있습니다.
  - $D(i, 0) :=$  길이가  $i$ 이고,  $S$ 와  $C$ 의 개수가 짝수인 문자열의 수
  - $D(i, 1) :=$  길이가  $i$ 이고,  $S$ 와  $C$ 의 개수가 홀수인 문자열의 수

## E. even하게 익은 SCON - 동적 계획법

- ✓ 점화식은 문자열의 맨 뒤에 추가하는 문자에 따라 두 가지 경우로 나뉘서 계산할 수 있습니다.
  - $S$  또는  $C$ 를 추가하는 경우:  $D(i, j) \leftarrow 2 \times D(i - 1, 1 - j)$
  - $S$ 와  $C$ 가 아닌 다른 문자를 추가하는 경우:  $D(i, j) \leftarrow 24 \times D(i - 1, j)$
- ✓ 따라서 아래 식을 이용해 답을  $O(N)$  시간에 문제의 정답인  $D(N, 0)$ 을 구할 수 있습니다.
  - $D(i, 0) = 24 \times D(i - 1, 0) + 2 \times D(i - 1, 1)$
  - $D(i, 1) = 24 \times D(i - 1, 1) + 2 \times D(i - 1, 0)$

#### E. even하게 익은 SCON - 조합론

- ✓ 별해로 이항 계수와 모듈러 곱셈 역원으로 정답을 구할 수 있습니다.
- ✓  $S, C$ 의 개수를  $2i$ 개로 고정시켜 봅시다.
- ✓ 이때 가능한 문자열의 개수는  $\binom{n}{2i} \cdot 2^{2i} \cdot 24^{n-2i}$  개 입니다.
- ✓ 따라서 정답은  $\sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{2i} \cdot 2^{2i} \cdot 24^{n-2i}$  입니다.
- ✓  $2^{2i}$  와  $24^{n-2i}$  는  $O(n)$  시간에 모두 계산할 수 있습니다.
- ✓ 모듈러 곱셈 역원을 이용하면  $O(n + \log P)$  시간의 전처리를 통해 이항 계수를 상수 시간에 구할 수 있습니다.
- ✓ 따라서 전체 시간 복잡도는  $O(n + \log P)$  입니다. (단,  $P = 10^9 + 7$ )

### E. even하게 익은 SCON - 일반항

- ✓  $A, B, \dots, Z$ 를 변수로 생각합시다.
- ✓  $(A + B + \dots + Z)^n$ 을 전개한 각 항은 길이  $n$ 인 서로 다른 문자열입니다.
  - 예)  $(A + B + \dots + Z)^2 = AA + AB + \dots + AZ + BA + \dots + ZA + ZB + \dots + ZZ$
- ✓  $S = C = -1$ , 나머지 변수에 1을 대입하면 각 문자열의 값은  $(-1)^{(S, C \text{가 등장한 횟수})}$ 입니다.
  - 예)  $SCON = (-1) \times (-1) \times 1 \times 1 = (-1)^2$ ,  $ASDF = 1 \times (-1) \times 1 \times 1 = (-1)^1$
  - 문자열의 값은  $S, C$ 가 홀수번 등장하면 값은  $-1$ , 짝수번 등장하면  $1$ 이 됩니다.
  - $22^n = (A + B + \dots + Z)^n =$  길이  $n$ 인 서로 다른 모든 문자열 값의 합
$$= (S, C \text{가 짝수번 등장하는 문자열 수}) - (S, C \text{가 홀수번 등장하는 문자열 수})$$
$$26^n = (S, C \text{가 짝수번 등장하는 문자열 수}) + (S, C \text{가 홀수번 등장하는 문자열 수})$$
- ✓ 두 식을 연립하면  $(S, C \text{가 짝수번 등장하는 문자열 수}) = \frac{26^n + 22^n}{2}$ 입니다.

$$F. A = B \oplus C$$

math, dp

출제진 의도 – **Medium**

- ✓ 처음 푼 팀: **와장창특공대**, 105분
- ✓ 문제 아이디어: 김서윤
- ✓ 문제 작업: 김서윤

## F. $A = B \oplus C$

- ✓  $A_{3k-2}, A_{3k-1}, A_{3k}$ 의 값을 결정하는 방법은 4가지입니다.
  1.  $0 \oplus 0 = 0$
  2.  $0 \oplus 1 = 1$
  3.  $1 \oplus 0 = 1$
  4.  $1 \oplus 1 = 0$
- ✓ 1을 2개, 0을 1개 사용하는 방법이 3가지, 0만 3개 사용하는 방법이 1가지입니다.

## F. $A = B \oplus C$

- ✓ 따라서 1이  $x$  개, 0이  $y$  개 있을 때 만들 수 있는 수열의 개수  $f(x, y)$  는 다음과 같이 계산할 수 있습니다.

$$f(x, y) = \begin{cases} 0 & \text{if } x < 0 \text{ or } y < 0 \\ 1 & \text{if } x = y = 0 \\ 3f(x-2, y-1) + f(x, y-3) & \text{otherwise} \end{cases}$$

- ✓ 따라서 동적 계획법을 이용하면  $O(XY)$  시간에 문제를 해결할 수 있습니다.



## G. 불꽃놀이의 아름다움 2

bipartite\_graph, dfs

출제진 의도 – **Hard**

- ✓ 처음 푼 팀: **와장창특공대**, 57분
- ✓ 문제 아이디어: 진민성
- ✓ 문제 작업: 박근형

## G. 불꽃놀이의 아름다움 2

- ✓ 문제에서 주어진 구조를 정점과 간선이  $N$  개인 연결 그래프로 생각합시다.
- ✓ 이러한 그래프는 트리에 간선 1개를 추가해서 만들 수 있습니다.
- ✓ 따라서 이 그래프는 **단순 사이클** 하나와, 그 주변에 몇 개의 **트리**가 붙어 있는 형태입니다.
- ✓ **단순 사이클**은 길이가 짝수일 경우 2개, 홀수일 경우 3개의 색으로 칠할 수 있습니다.
- ✓ **트리**는 깊이를 2로 나눈 나머지를 이용하면 항상 2개의 색으로 칠할 수 있습니다.
- ✓ 따라서 문제의 정답은 사이클의 길이를 2로 나눈 나머지를 이용해 결정할 수 있습니다.
- ✓ DFS 등을 이용해 사이클의 길이를 구하면  $O(N)$  시간에 문제를 해결할 수 있습니다.

## G. 불꽃놀이의 아름다움 2

- ✓ 여담으로, 어떤 그래프를 2개의 색으로 칠할 수 있는 것과 그래프에 홀수 길이 사이클이 없는 것은 동치입니다.
- ✓ 따라서 정답이 항상 2 또는 3이라는 것을 관찰했다면, 그래프가 이분 그래프인지 판별하는 것으로 더 편하게 구현할 수도 있습니다.

## H. 서로소 조합

math, number\_theory

출제진 의도 – **Hard**

- ✓ 처음 푼 팀: **1학년1학기ps**생초보**3인**큐, 158분
- ✓ 문제 아이디어: 최도현
- ✓ 문제 작업: 이창민

## H. 서로소 조합

- ✓ 첫 번째로 주어진 조합을  $C_1$ , 두 번째로 주어진 조합을  $C_2$  라고 합시다.
- ✓  $C_1$  과  $C_2$  를 직접 계산하여 최대 공약수가 1인지 확인하기에는 수가 너무 커질 수 있습니다.
- ✓  $n!$  은  $n$  이하의 소수의 곱으로 표현할 수 있다는 사실을 이용합시다.
- ✓  $0!$  부터  $5000!$  까지를 소인수분해했을 때 특정 소수가 몇 번 곱해졌는지를 구할 수 있다면  $C_1$  과  $C_2$  의 소인수와 그 소인수가 몇 번 곱해졌는지를 뺄셈을 통해 구할 수 있습니다.
- ✓ 어떤 소수가  $C_1$  의 소인수이면서  $C_2$  의 소인수라면 두 조합은 서로소가 아닙니다.

## H. 서로소 조합

- ✓ 5000 이하의 소수의 개수를  $P$  라고 합시다.
- ✓  $n!$  을 소인수분해 할 때는  $(n - 1)!$  을 소인수분해 한 결과와  $n$  을 소인수분해 한 결과를 더해서 구할 수 있습니다.
- ✓  $0!$  부터  $5000!$  까지 팩토리얼을 소인수분해하는데 걸리는 시간은  $O(5000P + S)$  이며, 여기서  $S$  는 0 부터 5000 까지를 소인수분해하는데 걸리는 시간입니다.
- ✓ 쿼리마다  $O(P)$  의 시간이 걸리므로,  $Q$  개의 쿼리를 처리하는데 걸리는 시간은  $O(PQ)$  입니다.
- ✓ 따라서, 전체 시간 복잡도는  $O(5000P + S + PQ)$  입니다. 구현을 간단하게 하기 위해  $P$  대신 5000 을 사용해도 괜찮습니다.

# I. 대결

ad\_hoc, prefix\_sum

출제진 의도 - **Hard**

- ✓ 처음 푼 팀: **와장창특공대**, 111분
- ✓ 문제 아이디어: 조문성
- ✓ 문제 작업: 조문성

## I. 대결

- ✓ 먼저  $T_i$  를 다음과 같이 정의합니다:

$$T_i = \begin{cases} B_i, & \text{if } A_i = 1 \\ -B_i, & \text{otherwise} \end{cases}$$

- ✓ 이때 [1번 팀 점수 - 0번 팀 점수]는  $T_i$  의 구간 합을 이용해 표현할 수 있습니다.
- ✓ 이를 “점수 차”라고 부르겠습니다.
- ✓  $T_i$  의  $[l, r]$  구간 합을  $S[l : r]$  이라고 정의합시다.



## I. 대결

- ✓ 이제  $K$  개의 라운드로 분할된 구간

$$[1, \dots, i_1], [i_1 + 1, \dots, i_2], \dots, [i_{k-1} + 1, \dots, i_k = N]$$

을 생각해 봅시다.

- ✓ 이때 점수 차는 다음과 같습니다:

$$S[1 : i_1] + 2 \cdot S[i_1 + 1 : i_2] + \dots + K \cdot S[i_{k-1} + 1 : i_k]$$

- ✓ 이 점수를 다시 쓰면 다음과 같이 표현할 수 있습니다:

$$S[1 : N] + S[i_1 + 1 : N] + \dots + S[i_{k-1} + 1 : N]$$

- ✓ 즉, 점수 차는  $T_i$ 의 접미사 구간 합들의 합으로 표현할 수 있습니다.

## I. 대결

- ✓ 이제 점수 차를 어떻게 하면 최대화할 수 있을까요?
- ✓  $K$  개의 라운드로 구성했을 때 얻는 점수 차는

$$S[1 : N], S[2 : N], \dots, S[N : N]$$

중  $K$  개를 골라 더한 값입니다. (단,  $S[1 : N]$  은 반드시 포함돼야 합니다.)

- ✓ 점수 차를 최대화하려면, 이들 중 양수인 값만 모두 더하면 됩니다. 이때 양수의 개수가 곧 라운드의 개수  $K$  가 됩니다.
- ✓  $T_i$  와  $T_i$  의 접미사 누적합을 구한 뒤, 양수만 골라 더하는 방식으로  $\mathcal{O}(N)$  에 문제를 해결할 수 있습니다.

# J. 맛있는 스콘 만들기

dp, data\_structures

출제진 의도 – **Challenging**

- ✓ 처음 풀 팀: **N/A**, N/A분
- ✓ 문제 아이디어: 최도현
- ✓ 문제 작업: 최도현

## J. 맛있는 스콘 만들기

- ✓  $t - 1$  초의 각 온도에서 스콘의 맛의 최댓값을 알고 있다면  $t$  초의 각 온도에서 맛의 최댓값 역시 결정할 수 있으므로 동적 계획법으로 해결할 수 있습니다.
- ✓  $dp[t][i]$  :  $t$  초에서 오븐의 온도가  $i$  일 때 스콘의 맛의 최댓값으로 두고 풀이할 수 있습니다.
- ✓ 우선  $C$  조건을 고려하지 않고 생각해 봅시다.
- ✓  $D$  조건은  $dp[t - 1]$  에서 가져올 수 있는 온도의 구간을  $[i - D, i + D]$  로 제한합니다.
- ✓  $i$  에서  $i + 1$  로 바뀌었다면 구간은  $[i - D + 1, i + D + 1]$  로 바뀌고,  $i - D$  에 있는 값은 사라지며,  $i + D + 1$  에 있는 값이 추가됩니다.
- ✓ 특정 값의 삭제와 삽입이 가능하면서 최댓값을 빠르게 구할 수 있는 자료구조를 사용하면 문제를 해결할 수 있습니다.

## J. 맛있는 스콘 만들기

- ✓ 모든  $i$ 에 대해  $[i - D, i + D]$ 에서 최댓값을 구할 때는 다음과 같은 방법을 사용할 수 있습니다.
  1. 1에서  $M$ 으로 진행하면서  $[i - D, i]$ 의 최댓값을 구하고,  $M$ 에서 1로 진행하면서  $[i, i + D]$ 의 최댓값을 구한다.
  2. 1에서  $M$ 으로 진행하면서 한 번에  $[i - D, i + D]$ 의 최댓값을 구한다.
- ✓ 해설에서는 1번 방법에서  $[i - D, i]$ 의 최댓값을 구하는 방법만 설명합니다.

## J. 맛있는 스콘 만들기

- ✓ 이제  $C$  조건을 추가해 봅시다.
- ✓  $C$  개의 자료구조를 만들고, 각각을 0 번부터  $C - 1$  번까지 번호를 붙여 구분하겠습니다.
- ✓ 1 에서  $M$  으로 진행하면서  $i$  일 때,  $i \% C$  번 자료구조에  $dp[t - 1][i]$  에 해당하는 값 또는 원소를 삽입합니다.
- ✓ 최댓값 역시  $i \% C$  번 자료구조에서 가져옵니다.
- ✓ 이후 페이지에선 자료구조 별로 어떻게 구간 내의 값들을 관리하는지 설명합니다.

## J. 맛있는 스콘 만들기 - 덱

- ✓ 덱의 back에 (온도, 맛) 쌍을 가지는 원소를 삽입합니다.
- ✓ 이때, 삽입하려는 원소보다 덱의 back에 있는 원소의 맛이 더 클 때까지 back에 있는 원소를 삭제합니다.
- ✓ 덱의 front에는 맛의 최댓값이 들어있으며, 덱에 들어있는 원소들의 맛은 단조 감소합니다.
- ✓ 만약, front에 있는 원소의 온도가  $[i - D, i]$  를 벗어났다면 front의 원소를 삭제하면 됩니다.
- ✓ 시간 복잡도는  $O(NM)$  입니다.

## J. 맛있는 스콘 만들기 - 우선순위 큐

- ✓ (온도, 맛) 쌍을 가지며 맛이 클수록 우선순위가 높은 우선순위 큐를 만들어줍니다.
- ✓ 최댓값을 가져올 때, 온도가  $[i - D, i]$  를 벗어났다면 해당 원소를 삭제하면 됩니다.
- ✓ 시간 복잡도는  $O(NM \log M)$  입니다



## J. 맛있는 스콘 만들기 - multiset, map

- ✓ 온도로 삭제 여부를 판단하는 것이 아닌 값 자체를 삭제하는 것도 고려해볼 수 있습니다.
- ✓  $[i - D, i]$ 에서 최댓값을 가져온 후,  $dp[t - 1][i - D]$ 에 해당하는 값은 더 이상 사용하지 않으므로 삭제하면 됩니다.
- ✓ multiset 또는 map을 사용하면 특정 값의 삽입/삭제가 가능하며, 최댓값도 구할 수 있습니다.
- ✓ 시간 복잡도는  $O(NM \log M)$ 입니다

## LLM 모델 테스트

문제	GPT 4.1	GPT 4.5	Claude 3.7	Claude 3.7 Rsn	Gemini 2.5 Pro	GPT o3
A	WA	AC	AC	AC	AC	AC
B	AC	AC	AC	AC	AC	AC
C	TLE	WA	AC	TLE	TLE	AC
D	AC	AC	AC	AC	AC	AC
E	AC	AC	AC	AC	AC	AC
F	WA	AC	TLE	AC	AC	AC
G	MLE	MLE	AC	AC	AC	AC
H	AC	AC	TLE	AC	AC	AC
I	WA	WA	AC	TLE	TLE	AC
J	TLE	TLE	TLE	TLE	TLE	AC
총점	4	6	7	7	7	10

- ✓ 대회 문제의 모범 코드는 <https://sccc.kr/scon/2025> 에서 확인할 수 있습니다.
- ✓ 감사합니다.