

SCCC 2학기 개인 내전 풀이

Official Solutions

by

m4ushold, chleee, dkprk11

문제	난이도
A 기사의 방패	Easy
B 제비 뽑기	Easy
C 숫자 놀이	Easy
D 영화관	Easy
E 공정한 수열	Medium
F 측정기 놀이	Medium
G 생일 축하 퍼즐	Medium
H 다과제	Medium
I AREA 998244353	Medium
J 최대합	Hard
K 자료구조 및 실습	Hard
L 서버 롤백	Challenging

A. 기사의 방패

math

난이도 – **Easy**

A. 기사의 방패

- ✓ 두 방패를 이어 붙인 둘레의 길이를 최소화해야 합니다.
- ✓ 두 삼각형에서 하나의 변의 길이를 각각 a, b 라고 가정해봅시다.
- ✓ 그러면 둘레의 길이는 모든 삼각형의 변의 길이 $-2 * \min(a, b)$ 가 됩니다. 따라서 각 삼각형에서 가장 큰 변의 길이를 가져와서 최솟값의 두배를 빼주면 됩니다.

B. 제비 뽑기

bruteforce

난이도 – **Easy**

B. 제비 뽑기

- ✓ $n \leq 100$ 이며 최대 점수도 100점이므로 모든 숫자를 불러보며 점수가 더 높은 인원의 수를 구하는 방법을 시도할 수 있습니다.
- ✓ 창민이가 101을 부를 수 있다는 점에 유의합니다.

C. 숫자놀이

math, simulation

난이도 – **Easy**

C. 숫자 놀이

- ✓ 현재 단어를 말하고 있는 아이의 인덱스를 p 라고 합시다.
- ✓ $p = 0$ 에서 출발하여 다음 k 번째 단어를 부르는 아이의 인덱스를 $p := (p + k) \bmod n$ 으로 구할 수 있습니다.
- ✓ 두 번 이상 지목될 시 게임이 종료되므로 게임이 진행되는 횟수는 최대 n 번입니다.
- ✓ 즉, 최대 n 번만 시뮬레이션 하면 되므로 시간 복잡도는 $O(n^2)$ 입니다.

D. 영화관

math, gcd

난이도 – **Medium**

D. 영화관

- ✓ i, j 를 위치라고 하고 행순서, 열순서대로 해당하는 숫자를 표현하면 $r = (i - 1) * m + j, c = (j - 1) * n + i$ 입니다.
- ✓ 따라서 $r = c$ 이면 $(i - 1)m + j = (j - 1)n + i$ 이고 정리하면 $(m-1)(i-1) = (n-1)(j-1)$ 입니다.
- ✓ 이 방정식은 i 와 j 가 $1 \leq i \leq n$ 과 $1 \leq j \leq m$ 범위를 만족하는 해의 개수를 찾는 문제입니다.
- ✓ 이 방정식의 해의 개수는 $\gcd(m - 1, n - 1)$ 의 배수를 찾는 것과 같습니다.
- ✓ 각각은 lcm 과 어떤 정수 s 의 곱으로 표현이 가능하고 $(i - 1)(m - 1) = (j - 1)(n - 1) = \text{lcm}(n - 1, m - 1) * s$.

D. 영화관

✓ 수식을 정리해보면

$$i - 1 = \frac{lcm(n - 1, m - 1) * s}{m - 1}, \quad j - 1 = \frac{lcm(n - 1, m - 1) * s}{n - 1}$$

$$i - 1 = \frac{(n - 1) * s}{gcd(n - 1, m - 1)}, \quad j - 1 = \frac{(m - 1) * s}{gcd(n - 1, m - 1)}$$

$$i - 1 = 0 \leq \frac{(n - 1) * s}{gcd(n - 1, m - 1)} \leq n - 1, \quad j - 1 = 0 \leq \frac{(m - 1) * s}{gcd(n - 1, m - 1)} \leq m - 1$$

$$0 \leq s \leq gcd(n - 1, m - 1)$$

- ✓ 따라서 $i - 1$ 과 $j - 1$ 가 가질 수 있는 값의 개수는 $gcd(m - 1, n - 1) + 1$ 입니다. (0부터 시작하므로)
- ✓ gcd는 유클리드 호제법을 사용하면 $O(\log \max(n, m))$ 에 구할 수 있습니다

E. 공정한 수열

math, bruteforce

난이도 – **Medium**

E. 공정한 수열

- ✓ 짝수 인덱스를 *even*, 홀수 인덱스를 *odd* 라고 합시다.
- ✓ 주어진 문자열에서 *even* 의 1의 개수와 *odd* 의 1의 개수를 먼저 구합니다.
- ✓ 어떤 인덱스 i 에서 문자를 지운다는 것은, 지금까지 봤던 *even* 의 1의 개수와 *odd* 의 1의 개수를 유지 한 뒤, 이 뒤에 존재하는 *even* 과 *odd* 의 1의 개수를 바꾼다고 생각할 수 있습니다.
- ✓ 즉, i 까지 구한 *even* 과 *odd* 의 1의 개수를 알면 i 에서 문자를 지웠을 때, 문자열 전체 *even* 의 1의 개수와 *odd* 의 1의 개수를 $O(1)$ 에 구할 수 있습니다.
- ✓ 시간 복잡도는 $O(n)$ 입니다.

F. 측정기 놀이

implementation, brute force

난이도 – **Medium**

F. 측정기 놀이

- ✓ 가능한 거리의 후보 수는 $n(n + 1)/2$ 입니다.
- ✓ 가능한 모든 순서쌍을 보면서 map과 set등을 통해 모든 정점이 특정 거리를 가지는지 확인하면 됩니다.
- ✓ 시간 복잡도는 $O(n^2 \log n)$ 입니다.

G. 생일 축하 퍼즐

simulation, constructive

난이도 – **Medium**

G. 생일 축하 퍼즐

- ✓ 문제의 추가 조건에 주목합시다.
- ✓ n 번 회전하기 전까지 같은 무늬가 등장하지 않기에 최대 $n - 1$ 번까지 회전시켰을 때, 적어도 하나의 열을 unique하게 만들 수 있습니다.
- ✓ 즉 i 번째 층을 최대 $n - 1$ 번까지 회전시키면서 등장한 문자열이 i 개 이상이면 $i + 1$ 번째 층으로 넘어가는 식으로 시뮬레이션 하면 모든 퍼즐을 풀 수 있습니다.
- ✓ 시간복잡도는 $O(n^3)$ 입니다.
- ✓ 여담으로 지문에 등장한 dkprk11의 생일은 9월 11일이었습니다.

H. 다과제

`topological_sort`, `greedy`

난이도 – **Medium**

H. 다과제

- ✓ 현재 PC에서 최대한 같은 PC에서 문제를 푸는 그리디가 성립합니다.
- ✓ 노트북에서 시작하는 경우와 데스크탑에서 시작하는 경우를 나눠 위상 정렬을 수행하면 됩니다.

I. AREA 998244353

sort, binary_search

난이도 - **Medium**

I. AREA 998244353

- ✓ 먼저 방공호를 오름차순으로 정렬합니다.
- ✓ 어떤 마을의 위치를 k 라고 하면 k 에서 가장 가까운 방공호는 k 에서 보다 크거나 같은 수 중 가장 먼저 등장하거나, k 보다 작거나 같은 수 중 가장 나중에 등장하는 위치에 존재합니다.
- ✓ 이는 이분 탐색으로 $O(\log m)$ 에 찾을 수 있습니다.
- ✓ 전체 시간 복잡도는 $O((n + m) \log m)$ 입니다.

J. 최대 합

dp, prefix_sum

난이도 - **Hard**

J. 최대 합

- ✓ 주어진 $m \times n$ 행렬에서 경계선에 있는 숫자들의 합이 최대가 되는 직사각형을 찾는 문제입니다.
- ✓ $R[i][j]$: i 행 1열부터 j 열까지의 합, $C[i][j]$: j 열 1행부터 i 행까지의 합을 미리 계산하여, 경계선 합을 $O(1)$ 시간에 구할 수 있도록 합니다.
- ✓ $D[s][e][j]$: s 행과 e 행을 포함하는 직사각형에서, j 열을 오른쪽 경계로 가질 때, 왼쪽 경계선(1열부터 j 열 중 하나)까지의 경계 합의 최댓값을 저장합니다.
- ✓ 이때 경계 합은 s 행의 일부, e 행의 일부, 그리고 s 행과 e 행 사이의 한 열을 포함하는 부분 직사각형의 합입니다.
- ✓ $D[s][e][j] = (\text{행 } s \text{의 } j \text{열까지 합}) + (\text{행 } e \text{의 } j \text{열까지 합}) - (\text{열 } j \text{의 } s \text{행부터 } e \text{행까지 합})$
- ✓ 이를 전처리된 누적 합 배열을 이용하여 $O(1)$ 에 계산할 수 있습니다.
- ✓ $D[s][e][j] = -R[s][j] - R[e][j] + (C[e][j] - C[s-1][j])$

J. 최대 합

- ✓ 위 식은 왼쪽 경계가 1열일 때의 합이며, 실제로는 1 ~ j 열 중 최적의 위치를 찾아야 합니다.
- ✓ 각 행 쌍 (s, e)와 현재 열 j에 대해, $D[s][e][j]$ 값은 1열부터 j열까지 중 가장 큰 값을 선택하여 갱신합니다.
- ✓ $P[s][e][j]$ 배열을 사용하여 최댓값을 준 열 번호를 저장합니다.
- ✓ s, i, j를 순회하며, s행과 i행, 그리고 $P[s][i][j - 1]$ 열과 j열을 경계로 하는 직사각형의 전체 경계 합을 계산합니다.
- ✓ s행과 i행, 그리고 $P[s][i][j - 1]$ 열과 j열을 포함하는 직사각형의 경계 합은 $D[s][i][j - 1]$ 에 i행과 s행의 j열 합을 더하고 중복되는 $A[s][j]$ 와 $A[i][j]$ 를 빼는 것으로 계산됩니다.
- ✓ 이 값을 최대 합 mx와 비교하여 갱신합니다.
- ✓ mx가 갱신될 때마다 해당 직사각형의 좌표 $(s, P[s][i][j - 1], i, j)$ 를 저장합니다.

K. 자료구조 및 실습

tree_dp, combinatorics

난이도 - Hard

K. 자료구조 및 실습

- ✓ $DP[v][h][k]$ = 정점 v 를 루트로 하는 검은 정점의 높이가 h 인 레드-블랙 트리의 개수로 정의합니다. ($k = 1$ 이면 v 는 검은 정점, $k = 0$ 이면 빨간 정점)
- ✓ $DP[0][0][1] = 1$ 로 전처리합니다.
- ✓ 만약 정점 v 가 리프 노드인 경우, $DP[v][1][1] = 1$ $DP[v][0][0] = 1$ 입니다.
- ✓ 만약 정점 v 가 리프 노드가 아닌 경우(= 자식이 하나라도 존재하는 경우),
$$DP[v][h][1] = \sum DP[leftchild][h-1][*] \cdot DP[rightchild][h-1][*],$$
$$DP[v][h][0] = DP[leftchild][h-1][1] \cdot DP[rightchild][h-1][1]$$
로 구할 수 있습니다.
- ✓ $n \leq 1000$ 이므로 리프 노드부터 올라가는 형태로 가능한 모든 높이를 계산해 줄 수 있습니다.

K. 자료구조 및 실습

- ✓ 추가로, BOJ에 업로드 되어있는 문제에는 $\text{mod } 10^9 + 7$ 연산이 존재하지 않습니다.
- ✓ 따라서 python으로 해결하거나 큰 수 연산을 구현해야 합니다.
- ✓ 시간 복잡도는 $O(n^2)$ 입니다.

L. 서버 롤백

persistent segment tree

난이도 – **Hard**

L. 서버 롤백

- ✓ 주어진 시작 복구 지점 l 부터 시작하여 최소 k 개의 서로 다른 서버를 복구할 수 있게 되는 가장 작은 r 값을 찾는 문제입니다.
- ✓ $R[i]$ 를 현재보다 오른쪽에 있는 복구지점중 같은 서버를 복구하는 가장 작은 복구지점이라고 합시다.
- ✓ n 부터 1까지 순회하며 $R[i]$ 를 -1 , i 를 $+1$ 로 pst 를 업데이트 해둡니다.
- ✓ 쿼리 l, r 이 주어졌을경우 $root[i]$ 에 k 번째 숫자를 찾는 쿼리를 세그먼트 트리에 보내면 됩니다.
- ✓ 쿼리가 오프라인으로 주어지면 seg sweeping으로 가능하지만 온라인으로 주어졌기 때문에 pst 를 사용해야하고 총 $O((n + q) \log n)$ 이 걸립니다.