2024 SMUPC

Official Solutions

by

SMUPC 출제진

2024년 5월 18일

문제		의도한 난이도	출제자
A	SMUPC NAME	**Easy**	0yeonjae2
B	열심히 일하는 중	**Easy**	0yeonjae2
C	비밀번호 만들기	**Medium**	songfox00
D	내진 설계	**Medium**	songfox00
E	승리하라	**Hard**	2093ab
F	눈송이 탕후루 만들기	**Hard**	2093ab
G	수열 만들기	**Hard**	2093ab



A. SMUPC NAME

`implementation, string`

출제진 의도 – **Easy**

- ✓ 제출 73번, 정답 24명 (정답률 32.877%)
- ✓ first solve: **einczkew** , 8분
- ✓ 출제자: `0yeonjae2`

A. SMUPC NAME

- 문자열과 숫자를 이용해 문자열을 만들어 출력하는 문제입니다.
- 지문에 주어진 사항을 그대로 구현하면 됩니다.



B. 열심히 일하는 중

`data_structure`, `priority_queue`

출제진 의도 – **Easy**

- ✓ 제출 152번, 정답 9명 (정답률 5.921%)
- ✓ first solve: **geenieyoon** , 58분
- ✓ 출제자: `@yeonjae2`

B. 열심히 일하는 중



- N 개의 일을 모두 다 끝내는 데 걸리는 날과 일자별로의 만족감을 출력하는 문제입니다.
- 중요도가 높은 일부터 해야 하고, 일을 한 뒤에 일의 중요도는 M 만큼 감소하기 때문에 중요도 값은 매일 바뀌게 됩니다.
- 중요도 값들이 매번 내림차순 정렬된 채 존재할 수 있도록 우선순위 큐를 사용해 중요도를 관리합니다.
- 우선순위 큐를 이용해 매번 시뮬레이션하거나, 중요도 값별로의 등장 횟수를 저장하는 배열을 만들고 이를 관리하는 방법을 통해서도 풀 수 있습니다.

C. 비밀번호 만들기

dp

출제진 의도 – **Medium**

- 제출 14번, 정답 2명 (정답률 14.286%)
- first solve: **syj2003** , 77분
- 출제자: songfox00

C. 비밀번호 만들기

- 2차원 배열을 활용한 기본적인 DP 문제입니다.
- $\mathrm{dp}[i][j]$에서 i는 자릿수를, j는 i번째 자리에 들어가는 알파벳을 의미합니다.
- 길이가 1인 경우 모든 알파벳으로 시작 가능하므로 $\mathrm{dp}[1][j] = 1$로 초기화 해줍니다.
- 2부터 M 자리까지 현재 자리에 j번 알파벳, 이전 자리에 k번 알파벳이 있을 때, $|j - k| \geq N$ 이면 $\mathrm{dp}[i][j] = (\mathrm{dp}[i][j] + \mathrm{dp}[i-1][k]) \% 1\,000\,000\,007$로 갱신해줍니다.
- $1 \leq j \leq 26$까지 $\mathrm{dp}[M][j]$ 더한 값을 $1\,000\,000\,007$로 나눈 값을 출력해주면 됩니다.

D. 내진 설계

`bfs`

출제진 의도 – **Medium**

- ✓ 제출 11번, 정답 1명 (정답률 9.091%)
- ✓ first solve: **einczkew** , 32분
- ✓ 출제자: `songfox00`

D. 내진 설계

- DFS/BFS로 풀 수 있는 문제입니다.
- '@'이 있는 곳의 좌표를 기준으로 시작합니다.
- 본진은 상하좌우로 2칸까지 이동하는데 이 때, $N \times M$ 지역을 벗어나거나 '|'를 만나면 이동을 멈춥니다.
- 본진이 건물을 만나면 해당 좌표의 방문 횟수를 1 증가시킵니다. 해당 좌표의 값이 '*'이고 방문 횟수가 1이면 건물이 무너졌기 때문에 무너진 건물 개수 +1, queue에 좌표를 추가해 줍니다.

D. 내진 설계

- ✓ 여진 역시 $N \times M$ 지역을 벗어나거나 '|'를 만나면 이동을 멈춥니다.
- ✓ queue에 있는 값을 꺼내 상하좌우 1칸씩 이동하는데, 해당 좌표의 값이 '*'이고 방문 횟수가 1이거나 값이 '#'이고 방문 횟수가 2이면 무너진 건물 개수 +1, queue에 좌표를 추가해 줍니다.
- ✓ 주어진 지역을 돌면서 쓰러지지 않은 건물의 개수를 구해준 뒤 결과를 출력하면 됩니다.

E. 승리하라

`bruteforce`

출제진 의도 – **Hard**

- ✓ 제출 11번, 정답 1명 (정답률 9.091%)
- ✓ first solve: **einczkew** , 156분
- ✓ 출제자: 2093ab

- 아직 진행되지 않은 경기가 최대 20경기라는 점이 이 문제의 가장 큰 핵심입니다.
- 경기를 진행하는 경우의 수는 최대 2^{20} 까지 나올 수 있으므로 모든 경우의 수에 대해서 결과를 확인해 보면 됩니다.
- 모든 경우에 대해서 1등인지 확인하는 연산을 $\mathcal{O}(20)$ 에 진행해 봐야 하는 문제입니다.
- 비트마스킹을 활용하여 진행되지 않은 경기의 모든 경우의 수에 대하여 경기 결과를 판단할 수 있습니다.

F. 눈송이 탕후루 만들기

`binary_search`, `data_structure`, `sorting`

출제진 의도 – **Hard**

- ✓ 제출 21번, 정답 1명 (정답률 4.762%)
- ✓ first solve: **einczkew** , 220분
- ✓ 출제자: 2093ab

F. 눈송이 탕후루 만들기

- 먼저, 각 꼬치의 선분에 존재하는 과일의 수를 파악하기 위해서는 꼬치 선분과 같은 기울기를 가진 과일들을 살펴보면 된다.
- 같은 기울기를 가졌는지는 x 와 y 를 $\gcd(|x|, |y|)$ 로 약분한 값으로 같은 방향(기울기)에 있는 과일인지 판단해 볼 수 있습니다.
- C++ STL Map 자료구조를 사용하면 같은 방향을 가진 과일들의 원점부터의 거리를 vector로 저장할 수 있습니다.
- 각 키에 대하여 해당 vector에 저장된 거리의 값들은 정렬하여 놓습니다.
- 하나의 꼬치에 포함되는 과일의 수는 같은 기울기와 방향을 가진 과일 중에서 거리를 기준으로 이분탐색을 진행하여 구할 수 있습니다.
- 각도정렬을 이용한 풀이도 가능합니다.

G. 수열 만들기

`segtree`

출제진 의도 – **Hard**

- 제출 6번, 정답 0명 (정답률 0.000%)
- first solve: **000** , 00분
- 출제자: 2093ab

G. 수열 만들기

- N 범위가 $2 \leq N \leq 300\,000$ 이므로 naive하게 $\mathcal{O}(N^2)$ 에 구하는 풀이는 시간초과를 받게 됩니다.
- 세그먼트 트리 자료구조를 활용하면 이를 $\mathcal{O}(NlogN)$ 에 구할 수 있습니다.
- 세그먼트 트리에서는 구간에서 남아있는 빈자리의 개수를 관리해 줄 겁니다.
- 수 1 을 인덱스 1 에 위치시키고 이를 기준으로 자리를 생각합니다.

G. 수열 만들기

- 우선 1 부터 p_i 까지 몇 개의 빈자리가 있는지 세그먼트 트리로 $\mathcal{O}(logN)$ 에 구할 수 있습니다.
- , p_i 기준으로 시계방향으로 x_i 번째 등장하는 빈자리의 위치는 구한 값에서 x_i 을 더하는 것이랑 같습니다.
- 이때 빈자리는 총 $N-i+1$ 개 있으므로 위에서 구한 값을 $N-i+1$ 로 모듈러를 취합니다.(모듈러는 0-base 기준으로 해야 한다는 점을 유의합시다)
- 위에서 구한 값은 1 부터 시작하여 수를 놓아야 하는 위치 전까지의 빈자리의 수와 같습니다.
- 이 위치는 세그먼트 트리에서 k 번째 값을 찾는 것을 이용하면 구할 수 있습니다.
- 그 위치에 수를 저장한 후 그 자리의 빈자리가 없어진 것을 세그먼트 트리에 업데이트하면 됩니다.