

2025 부산소프트웨어마이스터고등학교 알고리즘 경진대회 풀이 - 1학년

Official Solution

by

Team-INSERT, 김동욱

문제	의도한 난이도
A 우리의 친구 알이	Easy
B 선생님을 도와드리자	Easy
C 체스는 어려워	Easy
D 목이 너무 아파요	Medium
E 친구들과 족구를 해보자	Medium
F 학교 급식 먹기	Medium

A. 우리의 친구 알이

implementation

출제진 의도 – **Easy**

교내 대회

- ✓ 테스트케이스 제출 439번, 제출 147번, 정답 62명 (정답률 42.176%)
- ✓ 처음 푼 사람: **권길현**, 22초, Python

A. 우리의 친구 알이

- ✓ 입력은 없으며, 출력만 요구됩니다.
- ✓ 따라서, 출력 예시에 주어진 모양 · 공백 · 줄바꿈을 정확히 따라야 합니다.
- ✓ print 또는 cout을 사용하여 한 줄씩 출력하면 됩니다.
- ✓ 공백과 특수문자의 위치가 중요하므로, 복사하여 그대로 사용하는 것이 좋습니다.

B. 선생님을 도와드리자

implementation

출제진 의도 – **Easy**

교내 대회

- ✓ 테스트케이스 제출 1781번, 제출 402번, 정답 16명 (정답률 3.980%)
- ✓ 처음 푼 사람: **김현우**, 3분 47초, Python

B. 선생님을 도와드리자

- ✓ 먼저 학생 정보를 입력받아 리스트나 구조체에 저장합니다.
- ✓ 마지막 줄의 입력을 통해 정렬 기준 과목(subject) 과 정렬 방식(order) 을 결정합니다.
- ✓ 이후, 안정 정렬(C언어(stable-sort) 또는 Python(sorted))을 사용해 해당 과목 점수를 기준으로 정렬합니다.
- ✓ asc인 경우 오름차순, desc인 경우 내림차순으로 정렬하면 됩니다.

C. 체스는 어려워

implementation

출제진 의도 – **Easy**

교내 대회

- ✓ 테스트케이스 제출 1664번, 제출 466번, 정답 37명 (정답률 7.939%)
- ✓ 처음 푼 사람: **김현우**, 5분 32초, Python

C. 체스는 어려워

- ✓ 먼저 각 말에 대한 점수를 미리 정의해둡니다.
- ✓ 체스판은 총 8줄이므로, 각 줄의 문자를 하나씩 확인하면서 말이 존재할 경우 점수를 계산합니다.
- ✓ 대문자(흑색)는 black, 소문자(백색)는 white 점수에 더합니다.
- ✓ 마지막에 두 점수를 비교해 "White", "Black", "Draw" 중 하나를 출력합니다.

D. 목이 너무 아파요

implementation

출제진 의도 – **Medium**

교내 대회

- ✓ 테스트케이스 제출 450번, 제출 117번, 정답 4명 (정답률 3.418%)
- ✓ 처음 푼 사람: **석준영**, 31분 53초, C

D. 목이 너무 아파요

- ✓ 각 칸의 감소 가능 통증은 $\min(\text{해당 칸의 통증 수치}, C)$
- ✓ 이를 이용해 2차원 누적합 테이블을 구성
- ✓ 가능한 모든 파스 위치에 대해 누적합을 계산하여 최댓값 갱신
- ✓ 최적 위치와 그때 줄어든 통증의 합 출력

D. 목이 너무 아파요

- ✓ $X * Y$ 크기를 $O(1)$ 로 구하는 방법
- ✓ $\text{prefix}[i][j] = \text{reduced}[i-1][j-1] + \text{prefix}[i-1][j] + \text{prefix}[i][j-1] - \text{prefix}[i-1][j-1]$
- ✓ $\text{sum} = \text{prefix}[r2][c2] - \text{prefix}[r1-1][c2] - \text{prefix}[r2][c1-1] + \text{prefix}[r1-1][c1-1]$

E. 친구들과 족구를 해보자

implementation

출제진 의도 – **Medium**

교내 대회

- ✓ 테스트케이스 제출 565번, 제출 189번, 정답 7명 (정답률 3.703%)
- ✓ 처음 푼 사람: **박태률**, 34분 57초, Python

E. 친구들과 족구를 해보자



- ✓ 이 문제는 격자 위에서 여러 명의 사람이 족구공까지 도달하려고 할 때, 가장 먼저 도착할 사람을 찾는 시뮬레이션 문제이다. 일반적으로 이러한 문제는 여러 개의 시작점에서 목표까지의 최단 경로를 찾아야 하므로 BFS 알고리즘을 고려할 수 있다. 하지만 이 문제의 특성상, BFS보다 훨씬 간단한 방식으로 해결할 수 있다.

E. 친구들과 족구를 해보자

- ✓ 문제에서 사람은 오직 .(빈 칸)으로만 이동할 수 있으며, 운동장에는 어떠한 장애물도 존재하지 않는다. 또한 이동은 상하좌우로만 가능하므로, 사람이 공까지 이동하는 최소 이동 횟수는 단순히 맨해튼 거리로 계산할 수 있다. 즉, 어떤 사람이 (x_1, y_1) 위치에 있고 공이 (x_2, y_2) 에 있다면, 그 사람이 공까지 도달하는 데 걸리는 시간은 $|x_1 - x_2| + |y_1 - y_2|$ 가 된다. 이 수식은 BFS를 수행하지 않고도, 최단 거리를 직접 계산할 수 있게 해준다.

E. 친구들과 족구를 해보자



- ✓ 따라서 문제를 푸는 방법은 다음과 같이 간단히 정리된다. 먼저 운동장 전체를 입력으로 받아 사람과 공의 위치를 파악한다. 이후 각 사람에 대해 공까지의 맨해튼 거리를 계산하여, 이들 중 가장 짧은 거리를 가진 사람을 찾는다. 가장 짧은 거리를 가진 사람의 번호(입력 순서 기준으로 1번부터)를 출력하면 된다. 거리가 같은 사람이 여러 명인 경우는 없다.

F. 학교 급식 먹기

implementation

출제진 의도 – **Medium**

교내 대회

- ✓ 테스트케이스 제출 600번, 제출 59번, 정답 4명 (정답률 6.779%)
- ✓ 처음 푼 사람: **여철준**, 22분 36초, Python

F. 학교 급식 먹기

- ✓ $DP[i]$: i 일까지 고려했을 때 규칙을 지키며 얻을 수 있는 최대 만족도
- ✓ $score[i]$: i 일에 급식을 먹을 시 얻을 수 있는 만족도
- ✓ DP 점화식: $DP[i] = \max(DP[i - 1], DP[i - 2] + score[i])$

- ✓ 가능한 경우
- ✓ 오늘 먹지 않으면 $\Rightarrow DP[i - 1]$ (전날까지의 최대 만족도 유지)
- ✓ 오늘 먹으면 $\Rightarrow DP[i - 2] + score[i]$ (전날을 건너뛰고, 오늘의 만족도 추가)

F. 학교 급식 먹기

DP 초기값 설정

- ✓ $DP[0] = \max(0, score[0])$
- ✓ $score[0]$ 이 음수인 경우 먹는 게 손해일 수 있음
- ✓ $DP[1] = \max(DP[0], score[1])$
- ✓ 둘째 날에 먹을지, 첫째 날 먹었을지 중 선택

F. 학교 급식 먹기

서브태스크 분석

- ✓ 1. $2 \leq N \leq 20$, 점수 양수
완전탐색 or DP 점화식 사용
- ✓ 2. $2 \leq N \leq 100$, 점수 양수
DP 점화식 사용
- ✓ 3. 추가 제약 조건 없음.
DP 점화식 사용 + $DP[0]$ 초기화 사용 (DP 초기값 설정 페이지 참고)

F. 학교 급식 먹기

문제 분석

- ✓ 날짜의 개수 $N, 2 \leq N \leq 100$
- ✓ DP 점화식 사용 시간 복잡도 = $O(N)$
- ✓ 완전탐색 시간 복잡도 = $O(2^N)$
- ✓ 공간 복잡도 = $O(N)$