제1회 보라매컵 예선 풀이

2023년 1월 8일

문제		의도한 난이도	출제자
A	특식 배부	**Beginner**	99asdfg
B	출입 기록	**Easy**	99asdfg
C	시간 외 근무 멈춰!	**Easy**	99asdfg
D	잠입	**Medium**	asdarwin03, 99asdfg
E	조교의 맹연습	**Medium**	99asdfg
F	통신소	**Hard**	asdarwin03, 99asdfg
G	전투기 출격	**Challenging**	99asdfg
H	위문공연	**Challenging**	99asdfg

A. 특식 배부

math, arithmetic
출제진 의도 – **Beginner**

- 제출 334번, 정답 275명 (정답률 85.629%)
- 처음 푼 사람: **jhnah917**, 0분
- 출제자: 99asdfg

A. 특식 배부

- 만약 주문한 치킨 마릿수 N 이 후라이드 치킨을 원하는 인원 수 A 보다 크거나 같다면 A 명에게 모두 후라이드 치킨을 나눠줄 수 있고, N 이 A 보다 작다면 최대 N 명에게 후라이드 치킨을 나눠줄 수 있습니다.
- 다른 치킨 종류도 마찬가지이므로, 각 치킨 종류에 대해 나눠줄 수 있는 최대 마릿수는 각각 $\min(N, A), \min(N, B), \min(N, C)$ 마리입니다.
- 따라서, 단순히 $\min(N, A) + \min(N, B) + \min(N, C)$ 을 출력하면 됩니다.

B. 출입기록

`implementation, simulation`

출제진 의도 – **Easy**

- 제출 329번, 정답 165명 (정답률 52.280%)
- 처음 푼 사람: **jhnah917**, 1분
- 출제자: 99asdfg

B. 출입 기록

- 들어오는 입력을 그대로 시뮬레이션하며 정답을 구해봅시다.
- 출입 카드 번호가 $200\,000$ 이하이므로, 크기 $200\,000$의 배열을 만들어 각 번호의 사람이 부대 내에 있는지 여부를 추적할 수 있습니다.
- 각 사람이 부대 안에 있으면 배열의 값을 1로, 없으면 배열의 값을 0으로 설정합시다.
- 배열의 값과 동일한 입력이 들어오는 모순적인 경우에는 정답을 1 증가시켜주고, 배열의 값과 다른 입력이 들어오면 배열의 값만 변경해줍시다.
- 모든 출입 기록이 끝난 뒤에는 부대 내에 사람이 없어야 하므로, 입력이 끝난 뒤에는 배열의 값이 1인 개수만큼 정답을 증가시켜 주어야 합니다.
- 총 시간복잡도는 $\mathcal{O}(N)$입니다.

C. 시간 외 근무 멈춰!

greedy, sorting, simulation
출제진 의도 – **Easy**

- 제출 180번, 정답 57명 (정답률 33.333%)
- 처음 푼 사람: **kyo20111**, 9분
- 출제자: 99asdfg

C. 시간 외 근무 멈춰!

- 모든 작업을 마감 기한 이전에 끝내야 하므로, 마감 기한이 더 늦는 작업을 미리 진행하느라 마감 기한이 더 빠른 작업을 진행하지 못하는 일은 발생하면 안됩니다.
- 따라서, 각 작업에 드는 시간과 관계 없이 마감 기한이 빠른 작업부터 진행하는 것이 최적입니다.
- 마감 기한을 오름차순으로 정렬한 뒤, 평시 근무와 시간 외 근무를 직접 시뮬레이션합시다.
- 구현이 마냥 간단하지만은 않습니다. 지금까지 했던 평시 근무 일수와 시간 외 근무 일수를 저장해 둔 채로 각 작업에 대해 최대한 평시 근무로 채워넣는 방식으로 구현합시다.
- 총 시간복잡도는 $\mathcal{O}(N \log N)$입니다.

D. 잠입

greedy, implementation, case_work
출제진 의도 – **Medium**

- 제출 124번, 정답 23명 (정답률 19.355%)
- 처음 푼 사람: **eunlin**, 23분
- 출제자: asdarwin03, 99asdfg

D. 잠입

- 최 상병은 자율 방범 로봇과 레이저 센서에 발각되지 않고 (N, M) 까지 도달해야 합니다.
- 자율 방범 로봇이 어떻게 움직이더라도 발각되어서는 안되므로, 이를 통해 중요한 사실을 알 수 있습니다.
  - t 초 후 자율 방범 로봇은 2차원 그리드 상에서 $1 + y + x \leq t$ 를 만족하는 모든 위치 (y, x) 에 있을 수 있습니다.
  - 즉, 최 상병이 단 한번이라도 움직이지 않거나 왼쪽, 또는 위쪽으로 이동한다면, 자율 방범 로봇에 의해 발각될 수 있는 경우가 반드시 존재합니다.
- 따라서 최 상병은 오른쪽과 아래쪽으로만 움직여서 (N, M) 에 도달해야 합니다.

D. 잠입

- 또한 레이저 센서는 항상 각 행의 경계에만 설치되어 있으므로, 오른쪽에 한해 최 상병은 모든 제약을 받지 않고 자유롭게 움직일 수 있습니다.
- 이제 최 상병이 어떻게 이동하는 것이 최적일지 생각해봅시다.
- 어떤 위치 (y, x) 에 있는 최 상병이 다음 행으로 내려가는 상황을 생각해봅시다.
- 레이저 센서를 피해 아래 방향으로 움직일 수 있는 최 상병의 열 위치를 $i, i+1, ..., j$ 라 해봅시다.
- 최 상병의 열 위치 $k(i \le k \le j)$ 에서 (N, M) 로 이동할 수 있는 모든 열 위치의 집합을 a_k 라 할 때, $a_{k+1} \subset a_k$ 를 항상 만족합니다.
- 따라서 최 상병이 움직일 때, 가능한 오른쪽으로 움직이지 않고 아래쪽으로만 움직이는 것이 최적해이며, 이를 구현해서 (N, M) 에 도착할 수 있는지 판단해주면 됩니다.

D. 잠입

- 이때 M 제한이 10억으로 크다는 점에 유의하며 구현해야 합니다.
- 경계에 레이저 센서가 배치된 유형을 6가지 유형으로 나눌 수 있다는 점을 이용하면 각 경계에 대해 $\mathcal{O}(1)$에 다음 행에서의 최 상병의 위치를 구할 수 있습니다.
- 하나의 경계에 대해 입력이 열 위치의 오름차순으로 주어진다는 점을 이용하면 레이저 센서가 배치된 모든 경우에 대해 case work를 하지 않고도 간단히 그리디하게 현재 위치를 옮겨주면서 구현할 수도 있습니다.
- 총 시간복잡도는 $\mathcal{O}(N)$입니다.

E. 조교의 맹연습

dp , knapsack
출제진 의도 – **Medium**

- 제출 155번, 정답 30명 (정답률 22.581%)
- 처음 푼 사람 : **hijkl2e**, 13분
- 출제자 : 99asdfg

E. 조교의 맹연습

- 왼쪽, 오른쪽, 뒤쪽으로 도는 세 가지 동작을 적절히 사용하여 정확히 K 의 에너지를 사용해야 합니다.
- 또한, 동작을 모두 수행한 뒤에는 처음 바라보고 있는 방향을 바라보고 있어야 합니다.
- 각 동작을 수행하는 순서는 사용된 에너지 양 및 최종 방향에 영향을 끼치지 않습니다.
- 따라서, 순서와 관계 없이 적절한 횟수로 각 동작을 수행하여 조건에 맞출 수 있는지 확인하면 됩니다.

E. 조교의 맹연습

- 모든 동작을 수행 후 처음 바라보고 있는 방향을 바라보려면, 다음의 6가지 동작 조합만을 사용해야 합니다.

1. 좌로 돌아 1회 + 우로 돌아 1회, 사용 에너지양 $A+B$, 동작 수행 횟수 2회
2. 좌로 돌아 2회 + 뒤로 돌아 1회, 사용 에너지양 $2A+C$, 동작 수행 횟수 3회
3. 우로 돌아 2회 + 뒤로 돌아 1회, 사용 에너지양 $2B+C$, 동작 수행 횟수 3회
4. 뒤로 돌아 2회, 사용 에너지양 $2C$, 동작 수행 횟수 2회
5. 좌로 돌아 4회, 사용 에너지양 $4A$, 동작 수행 횟수 4회
6. 우로 돌아 4회, 사용 에너지양 $4B$, 동작 수행 횟수 4회

E. 조교의 맹연습

- 이제 위의 6 가지 동작만으로 정확히 K 만큼의 에너지를 사용할 수 있는지 확인해 봅시다.
- 이는 간단한 knapsack DP로 해결이 가능합니다.
- 총 시간복잡도는 $\mathcal{O}(N)$ 입니다.

- 위에서 제시된 방법 말고도, 4 가지 동작에 대해 방향을 기록하는 2 차원 knapsack DP로도 해결 가능합니다.
- 이 방법의 시간복잡도도 $\mathcal{O}(N)$ 입니다.

F. 통신소

bfs, sweeping
출제진 의도 – **Hard**

- 제출 60번, 정답 11명 (정답률 18.333%)
- 처음 푼 사람: **jhnah917**, 24분
- 출제자: asdarwin03, 99asdfg

F. 통신소

- 단순한 BFS 문제처럼 보이지만, naive하게 구현하면 $\mathcal{O}(NMK)$로 시간 초과를 받게 됩니다.
- 통신소에서 전파가 퍼져 나가는 모습을 생각해 봅시다.
- 세기 p의 전파는 세기가 $p-1$이 되며 상하좌우 네 칸으로 퍼져나갑니다.
- 이때, 만약 퍼져나간 칸에 이미 더 강한 전파가 있었다면, 세기가 더 약한 전파는 더 이상 퍼져나가지 않아도 됩니다.
- 동일한 칸에 전파가 여러번 들어오더라도, 세기가 약한 전파가 퍼지는 칸은 반드시 세기가 강한 전파가 퍼지는 칸과 중복되기 때문입니다.

F. 통신소

- 그러나, 이것을 naive하게 구현하면 시간 초과를 받게 됩니다. 동일한 격자점에 전파가 너무 여러 번 들어올 수 있기 때문입니다.
- 시간을 줄이기 위해선 동일한 격자점에 전파가 여러 번 들어오지 않게 해야 합니다.
- 전파가 퍼지는 순서는 정답에 영향을 끼치지 않으므로, 항상 세기가 강한 전파가 먼저 퍼져나간다면 지나왔던 격자점으로 다시는 들어가지 않아도 됩니다.
- 따라서, BFS를 돌릴 때 항상 강한 전파가 먼저 퍼져나가도록 구현하면 동일한 격자점에 최대 한 번씩만 접근할 수 있습니다.
- 이 아이디어를 활용하여 코드를 구현해 봅시다.

F. 통신소

- 우선 입력받은 통신소를 전파 세기의 내림차순으로 정렬해 둡시다.
- 전파의 세기는 최대 3000 까지이므로, 전파의 세기가 3000 인 전파부터 1 인 전파까지 순서대로 queue에 넣어주며 BFS를 돌려줍시다.
- 총 시간 복잡도는 $\mathcal{O}\left(NM + K \log K\right)$ 입니다.
- priority queue를 사용하여 구현하면 시간복잡도가 $\mathcal{O}\left(NM \log NM\right)$ 이 되어 시간 초과가 나게 된다는 점에 유의합시다.
- 별해로, 네 방향으로 스위핑하며 모든 칸에 4회씩 방문하는 풀이가 있습니다.
- 이 풀이의 시간 복잡도는 $\mathcal{O}\left(NM\right)$ 이며, 일반적으로 위의 BFS 풀이보다 빠르게 작동합니다.

G. 전투기 출격

floyd_warshall, prefix_sum, parametric_search
출제진 의도 – **Challenging**

- 제출 27번, 정답 6명 (정답률 22.222%)
- 처음 푼 사람: **jhnah917**, 74분
- 출제자: 99asdfg

G. 전투기 출격

- 동료 조종사가 순회하는 구역의 개수를 최소화하기 위해선, 하늘이가 사용하는 연료의 양을 최소화해야 합니다.
- 구역의 개수 $M \leq 100$ 이므로, 플로이드-워셜 알고리즘을 사용하여 $\mathcal{O}\left(M^3\right)$ 안에 각 구역에서 다른 구역으로 갈 때 필요한 연료의 최솟값을 계산할 수 있습니다.
- 이때, $dist[i][j]$ = 구역 i 에서 구역 j 로 갈 때 필요한 연료의 최솟값으로 정의합시다.

G. 전투기 출격

- 구간 합을 활용하여 각 경로까지 갈 때 필요한 연료의 합을 구해둡시다.
- 즉, $sum[i]$ = 구역 1 부터 i 번째 경로의 구역까지 갈 때 필요한 연료의 합으로 정의합시다.
- 이제 동료 조종사가 순회하는 경로의 구간을 $[l, r]$ 이라고 합시다. ($1 < l \leq r \leq N$)
- 또한, 문제에 주어진 것과 똑같이 i 번 구역에 착륙할 때 드는 연료의 양을 $f[i]$, i 번째 경로의 구역 번호를 $a[i]$ 라고 합시다.
- 그러면 구간 $[l, r]$ 을 동료 조종사가 순회해줄 때, 하늘이가 사용하는 총 연료의 양은 $sum[n] - (sum[r+1] - sum[l-1]) + dist[a[l-1]][a[r+1]] + f[a[l-1]]$ 입니다.

G. 전투기 출격

- 플로이드-워셜 알고리즘으로 모든 구역의 이동 거리를 최소화시켰기 때문에, 동일한 l 값에 대해 r 값이 늘어나면 늘어날수록 반드시 사용하는 총 연료의 양은 줄어들게 됩니다.
- 따라서, 2 부터 N 까지 총 $N-1$ 개의 l 값에 대해, Parametric Search를 활용하여 남아 있는 연료 R 보다 연료를 적게 사용하는 가장 작은 r 값을 구할 수 있습니다.
- 위 방식으로 구한 모든 (l, r) 쌍에 대해 $r - l + 1$ 값이 가장 작으면서, 사용하는 연료의 양이 최소인 값을 출력하면 됩니다.
- 총 시간복잡도는 $\mathcal{O}\left(M^3 + N \log N\right)$ 입니다.

H. 위문 공연

permutation_cycle_decomposition, combinatorics, math, ad_hoc
출제진 의도 – **Challenging**

- 제출 6번, 정답 4명 (정답률 66.667%)
- 처음 푼 사람: **edenooo**, 106분
- 출제자: 99asdfg

H. 위문공연

- 문제 상황이 상당히 복잡하므로, 이를 그래프로 단순화시켜 생각해 봅시다.
- i 번째 좌석을 i 번 노드라고 하고, i 번 노드에서 a_i 번 노드로 단방향으로 연결된 그래프를 생각해 봅시다.
- 위 방식대로 그래프를 구성하면, N 개의 노드와 N 개의 간선으로 이루어지며 모든 노드의 indegree와 outdegree가 1 인 그래프가 됩니다.
- 이때, 그래프가 두 개 이상의 연결 요소(Connected Component)들로 이루어질 수도 있습니다.

H. 위문공연

- 이제 각 노드를 서로 다른 세 가지 상태로 나눠봅시다.
  1. 누군가 착석해 있으며, 착석한 사람이 해당 좌석의 티켓을 가지고 있는 상태
  2. 누군가 착석해 있으며, 착석한 사람이 해당 좌석의 티켓을 가지고 있지 않은 상태
  3. 누구도 착석해 있지 않은 상태
- 위 정의에 따라, i 번째 좌석에 착석하고자 하는 사람이 공연장에 진입하기 전까지는 i 번 노드는 항상 3 번 상태를 유지합니다.

H. 위문공연

- ✓ i 번째 좌석을 원하는 사람이 입장하는 경우, 다음과 같은 일들이 벌어집니다.
  - i 번 노드에서 출발하여 3 번 상태가 아닌 노드를 만나기 전까지 방문한 모든 노드를 1 번 상태로 바꿉니다.
  - 만약 처음으로 만난 3 번 상태의 노드가 i 번 노드이면 i 번 노드를 1 번 상태로, 아니라면 2 번 상태로 바꿉니다.
  - i 번 노드에서 출발하는 연결을 끊은 뒤, 처음으로 만난 3 번 상태의 노드로 재연결합니다.
- ✓ 이때, i 번째 사람의 이동 횟수는 i 번 노드에서 처음으로 만난 3 번 상태의 노드까지 거쳐간 노드의 개수와 동일합니다.
- ✓ 또한, 처음으로 만난 3 번 상태의 노드가 i 번 노드이려면, i 번 노드는 반드시 i 번 노드를 포함하는 연결 요소 중 가장 높은 번호의 노드여야 합니다.

- 위 관찰을 토대로, 다음과 같은 사실들을 알아낼 수 있습니다.
  - 1번 상태의 노드는 다른 사람의 이동 횟수에 영향을 끼치지 않습니다.
  - 2번 상태의 노드는 다른 사람의 이동 횟수를 1 늘린 뒤 1번 상태로 바뀝니다.
  - 3번 상태의 노드는 각 이동의 처음과 끝을 나타내며, i번 노드가 해당 연결 요소 중 가장 높은 번호의 노드라면 이동 횟수는 $+1$, 아니라면 이동 횟수는 $+2$가 됩니다.

H. 위문공연

- 위 사실을 토대로, 크기가 s 인 하나의 연결 요소에 있는 노드들의 총 이동 횟수를 구해봅시다.
- 연결 요소 중 가장 마지막에 공연장으로 입장하는 하나의 노드를 제외한 모든 $s-1$ 개의 노드들은 2 번 상태를 거쳐므로, 2 번 상태가 증가시키는 이동 횟수는 $s-1$ 회입니다.
- 연결 요소 중 가장 마지막에 공연장으로 입장하는 하나의 노드는 $+1$ 의 이동 횟수를, 나머지 모든 $s-1$ 개의 노드는 $+2$ 의 이동 횟수를 가집니다.
- 2 번 상태가 증가시키는 이동 횟수는 $s-1$ 이고, 각 노드가 가지는 이동 횟수의 합은 $2(s-1)+1$ 이므로 한 연결 요소에서의 총 이동 횟수는 $3s-2$ 회 입니다.
- 따라서, 연결 요소의 개수를 x 라 하면, 총 이동 횟수는 $3N-2x$ 가 됩니다.

- 이제 티켓이 바뀌는 $\frac{N(N-1)}{2}$ 가지의 경우에 대한 이동 횟수의 합을 계산해 봅시다.
- i 번 티켓과 j 번 티켓이 서로 교환되었다면, i 번 노드는 a_j 번 노드로, j 번 노드는 a_i 번 노드로 연결되게 됩니다.
- 만약 초기 그래프에서 i 번 노드와 j 번 노드가 하나의 연결 요소에 있었다면, 해당 연결 요소는 두 개의 연결 요소들로 나눠지게 됩니다.
- 만약 초기 그래프에서 i 번 노드와 j 번 노드가 서로 다른 연결 요소에 있었다면, 그 두 개의 연결 요소들은 하나의 연결 요소로 합쳐지게 됩니다.

H. 위문공연

- i번 노드가 포함된 연결 요소의 크기가 s라고 할때, i번 노드와 다른 무작위 노드를 골랐을 때 이동 횟수의 총합은 다음과 같이 구할 수 있습니다.
  - 초기 그래프에서의 연결 요소 개수를 x라 합시다.
  - 선택한 다른 노드가 동일한 연결 요소에 있는 경우의 수를 $y = s - 1$라 합시다.
  - 선택한 다른 노드가 다른 연결 요소에 있는 경우의 수 $z = N - s$라 합시다.
  - 총 이동 횟수는 $\{3N - 2(x+1)\}y + \{3N - 2(x-1)\}z$입니다.
- 이때, 위에서 구한 총 이동 횟수는 중복을 포함한 값이므로, 계산을 완료한 후 값을 2로 나누어 주어야 합니다.
- 총 시간복잡도는 $\mathcal{O}(N)$입니다.