2025 KPSC
Summer Algorithm Challenge

by

2025 KPSC Summer Algorithm Challenge 출제진 및 검수진

문제		의도한 난이도	출제자
A	/gg	Easy	haru_101
B	공급 횡령	Medium	captain6700
C	스트릭 채우기	Medium	saywoo
D	서로소 그래프 게임	Hard	tapris
E	걸어가요	Hard	young_out
F	데이트 약속	Hard	young_out
G	우혁이와 엘리베이터	Hard	kangsm02
H	마이마이 순회 돌기	Challenging	chan120714
I	오버클럭	Challenging	tapris
J	회전체와 쿼리	Challenging	tapris

A. /gg

#string, #implementation

출제진 의도 - **Easy**

- 제출 285번, 정답 166명 (정답률 58.947%)
- 처음 푼 사람: seawon0808, 0분 45초
- 출제자: haru_101

A. /gg

- 이 문제에는 함정이 존재합니다.
- 팀원의 수를 N이라고 할 때, “팀원의 절반”은 N이 홀수냐 짝수냐에 따라 다릅니다.
  - N이 짝수면 $\frac{N}{2}$는 정수이므로 0의 개수가 $\frac{N}{2}$ 이상인지 검사하면 됩니다.
  - N이 홀수면 $\frac{N}{2}$는 $X.5$이므로 0의 개수가 $\lceil \frac{N}{2} \rceil$ 이상인지 검사하면 됩니다.
- 시간복잡도는 $O(N)$입니다.

A. /gg

- 각 언어별 내장 함수를 사용하면 편하게 문제를 풀 수 있습니다.
  - C++의 경우 `std::count(s.begin, s.end, '0')`
  - Python의 경우 `s.count('0', start_index, end_index)`
    - 이 문제에서는 `s.count('0')` 만 사용해도 됩니다.
  - Java의 경우 `s.replace("X", "").length()`

B. 공금 횡령

#data_structures, #hash_set, #set

출제진 의도 - **Medium**

- 제출 146번, 정답 94명 (정답률 65.068%)
- 처음 푼 사람: riroan, 2분 37초
- 출제자: captain6700

B. 공금 횡령

- 각 물품의 가격을 map 등을 활용해 key-vaule로 저장하고, 저장한 값을 기반으로 105%를 초과하는지 확인하면 됩니다.
- 다만, 입/출력값이 많아, 빠른 입출력을 사용하지 않는다면 TLE가 발생하는 문제입니다.
- 시간 복잡도 $O(N)$에 해결할 수 있습니다.

C. 스트릭 채우기

#greedy, #sorting

출제진 의도 - **Medium**

- 제출 118번, 정답 50명 (정답률 42.373%)
- 처음 푼 사람: urd05, 7분 12초
- 출제자: saywoo

- 스트릭 프리즈를 적게 쓰기 위해선 K일 동안 최대한 많은 문제를 풀어야 합니다.
- 푸는 데 걸리는 시간이 적은 문제부터 푸는 전략이 최적인 것을 알 수 있습니다.
- 문제를 푸는 데 걸리는 시간 기준으로 오름차순으로 정렬한 후, 순서대로 문제를 풀어 제출하고, 문제를 제출하지 않은 날은 스트릭 프리즈를 사용하여 스트릭 K일을 채웁니다.
- 이때, 필요한 스트릭 프리즈가 M개보다 많으면, 스트릭 K일을 채우는 게 불가능하므로 -1 을 출력합니다.
- 정렬하는 데 $O(N \log N)$, 스트릭 K일을 채울 수 있는지 확인하는 게 $O(N)$으로, 총 시간복잡도는 $O(N \log N)$입니다.

D. 서로소 그래프 게임

#game_theory

출제진 의도 - **Hard**

- 제출 56번, 정답 31명 (정답률 55.357%)
- 처음 푼 사람: yaksha, 10분 12초
- 출제자: tapris

- n이 작은 경우에서 직접 게임을 진행해 보며 각 플레이어의 필승전략을 찾아내면 됩니다.
- 같은 연결 요소에 속한 두 정점은 크기가 서로소가 아니므로 연결할 수 없습니다.
  - $n = 2$인 경우 선공이 두 정점을 잇는 간선을 추가하면 두 정점이 같은 연결요소에 속해 후공은 더 이상 서로소인 두 정점쌍을 선택할 수 없으므로 패배합니다.
  - $n \geq 3$인 경우 n이 홀수일 때와 짝수일 때로 나누어 생각해 볼 수 있습니다.

- 1이 아닌 연결 요소가 한 개 이하 일때 선공은 1이 아닌 연결 요소가 한 개인 상태를 만들거나 크기가 각각 홀수,2 인 두 1이 아닌 연결 요소가 존재하는 상태를 만들 수 있습니다.
  - n이 홀수일 경우 후공은 항상 자신의 차례에 크기가 1이 아닌 연결 요소가 한 개가 되도록 만들 수 있습니다. 따라서 모든 정점이 하나의 연결 요소를 이룰 때까지 반복해 후공은 게임에서 승리할 수 있습니다.
  - n이 짝수일 경우 또한 후공이 $n-5$번째 차례까지 홀수인 경우와 같은 전략을 취하고 $n-3$번째 차례에 크기가 짝수인 두 연결 요소를 만들어 승리할 수 있습니다.

E. 걸어가요

#bruteforcing, #math, #number_theory

출제진 의도 - **Hard**

- 제출 127번, 정답 17명 (정답률 13.386%)
- 처음 푼 사람: urd05, 25분 31초
- 출제자: young_out

E. 걸어가요

- 사람이 너무 많으므로, 두 사람만 있다고 가정해 봅시다. (X_1, S_1), (X_2, S_2)
- 만약 두 사람이 Y위치에서 만날 수 있다면, 두 사람은 또한 $Y + S_1 \times S_2$위치에서 만날 수 있음을 알 수 있습니다.
- 조금 더 엄밀하게는 $Y + \mathrm{lcm}(S_1, S_2)$에서 만날 수 있습니다.
- 그러므로 $S_1 \times S_2$범위만큼만 탐색한다면 만날 수 있는지 여부를 알 수 있습니다.

E. 걸어가요

- $X_1 + n \times S_1 \geq X_2$를 만족하는 가장 작은 정수 n에 대해 $Y = \{X_1 + nS_1, X_1 + (n+1)S_1, \ldots, X_1 + (n + S_2 - 1)S_1\}$의 모든 위치를 검사합니다. $S_i \leq 100$이므로 최대 $O(100)$이 걸립니다.
- 검사한 위치 중 만날 수 있는 위치가 있다면, $X_1 = Y, S_1 = \mathrm{lcm}(S_1, S_2)$로 갱신하고, 만날 수 없다면 -1 을 출력하고 즉시 종료합니다.
- 다음에는 갱신된 X_1, S_1 그리고 X_3, S_3에 대해 같은 작업을 합니다. 이를 모든 i에 대해 수행하면 됩니다.
- 시간 복잡도는 $O(100 \times N)$으로 해결할 수 있습니다.
- 여담으로, 중국인의 나머지 정리를 이용한 풀이도 있습니다.

F. 데이트 약속

#dp, #prefix_sum

출제진 의도 - **Hard**

- 제출 60번, 정답 14명 (정답률 23.333%)
- 처음 푼 사람: yaksha, 20분 7초
- 출제자: young_out

F. 데이트 약속

- DP를 이용하면 해결 가능합니다.
- 문제의 핵심은 N이 아닌 M을 기준으로 점화식을 세우는 것입니다.
- $dp[j]$를 1일부터 X_j일까지의 데이트를 끝마쳤을 때, 최대로 얻을 수 있는 애정의 총합으로 정의합시다.
- $dp[j]$를 갱신할 때, 모든 $k(k < j)$에 대하여 X_k일 이후로 연속으로 쭉 만났을 때를 고려합니다.
- X_j일에 만났을 때 얻는 애정량보다, 저주로 잃는 애정량 Y_j가 더 클 수도 있으므로 (X_k, X_j), $(X_k, X_j]$날 연속으로 만났을 때 중 더 큰 값으로 결정합니다.

F. 데이트 약속

- 중간에 여러 Y값들을 만날 수 있으므로, $P(a,b)$를 a일부터 b일 사이에 있는 Y값의 합이라고 정의합니다. 누적합을 이용하면 간단하게 구할 수 있습니다.
- n일간 연속으로 만났을 때 얻을 수 있는 애정의 합은 $S(n) = \frac{n\times(n+1)}{2}$입니다.
- 따라서 점화식은 다음과 같습니다.
- $dp[j] = \max\big(dp[k], dp[k] + S\big(X_j - X_k\big) + P\big(X_k + 1, X_j\big), dp[k] + S\big(X_j - X_k - 1\big) + P\big(X_k + 1, X_j - 1\big)\big)$
- 이를 모든 j에 대해 수행하면 $O(M^2)$에 해결 가능합니다.

F. 데이트 약속

- X_j에 $0, N$을 추가한다면 구현하기 더 간단합니다.
- 정답이 0이 되거나, INT 범위를 넘을 수 있음에 유의해주세요.

G. 우혁이와 엘리베이터

#graph_traversal, #shortest_path, #dijkstra

출제진 의도 - **Hard**

- 제출 40번, 정답 8명 (정답률 11.268%)
- 처음 푼 사람: forpractice, 47분 43초
- 출제자: kangsm02

G. 우혁이와 엘리베이터

- 주어지는 간선을 모두 저장을 하게 된다면 제한 내에 문제를 풀 수 없습니다.
- 따라서 간선 중 필요한 간선만 남겨 저장해야 합니다.
- 각 엘리베이터에서 이동할 수 있는 인접 층을 잇는 간선만을 남깁니다.
- 그 후 시작점과 도착점이 같은 간선 중 이동 시간이 가장 작은 간선만 저장합니다.
- 이렇게 최소한의 간선만 남긴 후 다익스트라를 진행해야 합니다.

G. 우혁이와 엘리베이터

- 현재 층수와 계단을 사용한 횟수를 하나의 정점으로 보고 다익스트라를 진행하면 됩니다.
- 계단을 최대 K번 사용할 수 있음에 유의해야 합니다.
- $(E,0),(E,1),\ldots,(E,K)$ 중 최솟값이 결과값입니다. 이 중 방문한 정점이 존재하지 않을 경우에는 -1을 출력하면 됩니다.
- $K' = K + 1$이라 정의할 때, 다익스트라의 시간복잡도는 $O\left(NK'\left(\min\left(\sqrt{M}, N\right) + 1\right)\right)\log(NK')$입니다.
- 정답이 INT 범위를 넘길 수 있음에 유의해야 합니다.

H. 마이마이 순회돌기

#segtree, #binary_search

출제진 의도 - **Challenging**

- 제출 16번, 정답 11명 (정답률 68.750%)
- 처음 푼 사람: sadtreap, 28분 17초
- 출제자: chan120714

H. 마이마이 순회돌기

- 승민이는 클리어 시간이 짧은 순서대로 플레이해야 가장 많은 노래를 클리어할 수 있습니다.
- 곡들을 클리어 타임이 작은순서대로 정렬하면 이 문제에서 요구하는 답을 구할 수 있습니다. 시간복잡도는 $O(NQ\log N)$ 입니다.
- 위의 방식대로 하면, TLE를 받게 됩니다.
- 곡 길이를 리프노드로 둔 구간 합 세그먼트트리를 작성하여, 이분탐색을 이용하여 값을 구하면 시간복잡도 $O(N+Q\log^2 N)$에 이 문제를 해결할 수 있습니다.
- 제곱근 분할법을 이용한 $O\left(N+Q\sqrt{\max(a_i)}\right)$ 풀이도 존재합니다.

I. 오버클럭

#implementation, #graph_traversal

출제진 의도 - **Challenging**

- 제출 20번, 정답 2명 (정답률 10.000%)
- 처음 푼 사람: urd05, 176분 57초
- 출제자: tapris

- 먼저 K가 0 이상 이면서 주어진 문제를 만족하는 상황을 생각해 봅시다.
- 이 상황에서 조건을 만족시키는 두 상태의 각 오버클럭 배율을 더한 결과 또한 위 조건을 만족합니다.
- 따라서 모든 출발점에서 그래프를 탐색하며 부분 그래프의 오버클럭 배율을 조정하고 이를 각각 더해주는 것으로 답을 구할 수 있습니다.
- 모든 공장에서 최대 진출차수가 1이므로 그래프는 트리이거나 사이클, 혹은 한개 이상의 트리가 사이클에 연결된 경우밖에 없습니다.
- 가장 간단한 일직선으로 이어진 경우를 생각해봅시다.

I. 오버클럭

- 시작점을 앞쪽, 종점을 뒤쪽으로 볼 때 각 공장의 오버클럭 배율이 (앞에 있는 공장들의 산출량의 곱) × (뒤에 있는 공장들의 투입량의 곱) 이면 조건을 만족함을 알 수 있습니다.
- 트리의 경우 여러 개의 일직선 형태의 부분 그래프로 나눠 처리한 후 더해주는 방법을 쓸 수 있습니다.
- 외부의 연결이 없는 사이클은 모든 투입량의 곱과 모든 산출량의 곱이 같으면 조건을 만족시킬 수 있으며 일직선 형태처럼 구할 수 있습니다.
- 외부의 연결이 있을 때는 사이클 내부 임의의 공장을 기준으로 투입량과 한 바퀴 돌아 나오는 산출량을 계산한 후 (투입량)>(산출량)이라면 조건을 만족하게 할 수

있으며 (투입량-산출량)만큼을 투입하고 아이템을 산출하지 않는 공장으로 생각하여 트리 형태와 같은 방법으로 풀 수 있습니다.

- 위와 같이 구현하였을 때 시작점이 하나이거나 전체가 하나의 사이클을 이루고 있다면 특정 공장에서 최대 9개의 다른 공장의 계수가 곱해지기에 오버클럭 배율이 10^9를 넘지 않습니다.
- 시작점이 두개 이상일 때에도 각 시작점마다 최대 10^8의 오버클럭 배율이 더해지므로 10^9를 넘을수 없습니다.
- 주어진 수식대로 연립방정식을 세워 양의 정수해를 찾는 풀이도 존재합니다.

J. 회전체와 쿼리

#geometry, #calculus, #prefix_sum

출제진 의도 - **Challenging**

- 제출 52번, 정답 6명 (정답률 13.462%)
- 처음 푼 사람: redcube231, 98분 57초
- 출제자: tapris

- d가 각각 단면의 무게중심이 회전축까지 떨어진 거리이고 s가 단면의 넓이일때, 회전체의 부피는 파푸스-굴딘 정리에 따라 $2\pi ds$임이 알려져 있습니다.
- 따라서 단면의 넓이와 무게중심을 구하면 부피를 구할 수 있습니다.
- 무게중심은 각 도형의 넓이를 가중치로 한 무게중심의 가중평균으로 구할 수 있습니다.
- 따라서 다각형의 넓이를 구하는 과정에서 (각 삼각형의 무게중심) $\times$ (그 삼각형의 넓이) 를 더해 다각형의 넓이로 나눠주면 다각형의 무게중심을 구할 수 있습니다.

- 이때 $6 \times$ (격자점 위 다각형의 무게중심) 이 항상 정수라는 점을 이용하여 실수오차를 줄일 수 있습니다.
- 이러한 방법으로 각 쿼리마다 넓이와 무게중심을 구해주면 각 쿼리당 $O(N)$으로 총 시간복잡도가 $O(NQ)$입니다.
- 여기서 다각형 테두리 부분의 넓이,무게중심을 누적합으로 전처리하여 쿼리당 $O(1)$으로 계산할 수 있습니다.
- 전처리에 $O(N)$이 걸리니 총 시간복잡도는 $O(N+Q)$가 됩니다.