PPC 2022

Official Solutions

by

POSCAT

2022년 9월 18일

문제		의도한 난이도	출제자
A	약속 장소	Easy	정윤성slah007
B	X 만들기 (Easy)	Medium	장래오leo020630
B2	X 만들기 (Hard)	Hard	장래오leo020630
C	MMST	Medium	장래오leo020630
D	가채점	Challenging	정윤성slah007
E1	신기한 숫자 1	Medium	장래오leo020630
E2	신기한 숫자 2	Hard	장래오leo020630
F	트리의 MEX	Hard	김범수qjatn0120
G	택배 색칠	Easy	장래오leo020630
H	멋쟁이 포닉스	Easy	장래오leo020630
I	잔디 예측하기	Medium	정윤성slah007
J	단짠단짠 피자	Medium	장래오leo020630
K	포닉스의 신비한 분자 보고서	Hard	정윤성slah007
L	포커	Hard	김범수qjatn0120
M	포스택	Easy	김범수qjatn0120, 장래오leo020630

A. 약속 장소

`implementation`
출제진 의도 – **Easy**

- ✓ 제출 266번, 정답 56명 (정답률 22.56%)
- ✓ First Solve: **jhnah917**, 2분
- ✓ 출제자: 정윤성`slah007`

A. 약속 장소

- 정답 문자열이 될 수 있는 모든 문자열을 시험해보면 됩니다.
- 각각의 가게 이름 후보에서 많아야 한 글자 다른 문자열만이 정답 문자열이 될 수 있고, 문자열 N 개에서 각각 L 개 위치에 A∼Z를 대입해보면 $26NL$ 개입니다. 모든 $26NL$ 개의 문자열이 조건을 만족하지 않는다면 조건을 만족하는 문자열은 존재하지 않습니다.
- 특정 문자열이 실제로 모든 가게 이름 후보와 많아야 한 글자 차이나는지 검사하는 데는 NL 이 걸리므로, 총 $26N^2L^2$ 에 풀 수 있습니다.

A. 약속 장소



- 앞의 풀이로도 충분히 제한 시간 내에 문제를 풀 수 있지만, 생각해보면 더 적은 수의 문자열만 고려해도 됩니다.
- 한 가게 이름 후보에서만 한 글자를 바꿔서 생기는 문자열만 검사해도 충분합니다. 그 중에 답이 존재하지 않는다면 그 이름 후보에 대한 조건을 만족할 수 없기 때문입니다.
- 이 경우 $26L$ 개만 봐도 되므로 총 $26NL^2$ 이 걸립니다.

A. 약속 장소



- 가장 빠른 풀이는 $O(NL)$에 동작합니다.
- 한 후보 문자열 S와 많아야 한 위치에서 다른 정답 문자열이 존재할 조건을 생각해 봅시다.
- 다른 후보 문자열 중에서 S와 3개 이상의 위치에서 다른 것이 존재한다면 정답 문자열이 존재하지 않습니다.
- 모든 후보 문자열이 S와 1개 이하의 위치에서 다르다면, S 자체가 조건을 만족합니다.
- 이외에는 후보 문자열 중 S와 2개 이하의 위치에서 다른 것이 존재하고, 그 2개 위치에서만 문자를 바꿔 보며 가능한지 검사하면 됩니다.

B. X 만들기

`dynamic_programming`
출제진 의도 – **Medium**, **Hard**

- ✓ 제출 34 / 10번, 정답 6명 (정답률 17.65% / 60.00%)
- ✓ First Solve: **aeren**, 22분
- ✓ 출제자: 장래오`leo020630`

B. X 만들기



- 우선, 중심이 되는 점 P를 고정해봅시다.
- 중심으로부터 각 사분면 방향으로 뻗어갈 수 있는 최대 길이를 구해야 합니다.
- 3사분면으로 뻗어갈 수 있는 최대 길이는 점들을 x 좌표로 정렬했을 때 P를 끝 점으로 하는 y 좌표의 최장 증가 부분 수열 (Longest Increasing Subsequence)의 길이와 같습니다.



- 최장 증가 부분 수열 문제는 DP를 통해 $O(N^2)$ 또는 $O(N \log N)$에 해결하는 방법이 알려져 있습니다.
- DP로 해결할 경우 점화식은 다음과 같습니다.
  - $DP[i] = \max_{j<i, x_j<x_i, y_j<y_i}(DP[j]) + 1$
  - 조건을 만족하는 j가 존재하지 않을 경우 $DP[i] = 1$
- 다른 사분면에 대한 최대 길이 역시 같은 방법으로 구해줄 수 있습니다.



- 각 사분면에 대한 LIS 값을 모두 더한 것이 가장 큰 점이 중심이 되며, 전체 점의 개수에서 해당 값을 뺀 것이 답이 됩니다.
- 각 사분면에 점이 최소 하나 존재해야 한다는 조건은 각 방향으로의 LIS 길이가 2 이상인지 판별해주면 됩니다.

C. MMST

`mst`

출제진 의도 – **Medium**

- 제출 34번, 정답 9명 (정답률 26.47%)
- First Solve: **jhnah917**, 30분
- 출제자: 장래오[leo020630]



- 모든 간선의 가중치가 다르다는 점에 집중해봅시다.
- 이 성질로 인해, 주어지는 그래프의 최소 신장 트리와 최대 신장 트리의 개수는 각각 유일하게 존재합니다.
- 만약 주어지는 그래프가 트리일 경우 최소 신장 트리와 최대 신장 트리가 동일하며, 이때만이 MMST를 만들 수 없는 경우에 해당합니다.



- ✓ $N \leq M$ 인 경우에는 반드시 MMST를 만들 수 있습니다.
- ✓ 가중치의 합이 서로 다른 신장 트리가 3개 이상이며, 이 중 최소 신장 트리와 최대 신장 트리가 각각 1개이므로 MMST가 최소 하나 이상 존재합니다.

C. MMST



- MMST는 다음과 같은 방법을 이용해 구성할 수 있습니다.
- 우선 최소 신장 트리 혹은 최대 신장 트리를 만듭니다.
- $N \leq M$ 이기 때문에 MST에 포함되지 않는 간선이 최소 하나 존재하게 됩니다.
- 이 간선을 MST에 추가하면 사이클이 하나 존재하는 그래프가 되며, 해당 사이클의 길이는 3 이상입니다.

C. MMST



- 이 사이클에서 가중치가 최소도 아니며 최대도 아닌 간선을 하나 제외하면 그래프는 다시 신장 트리가 되며, 이 신장 트리는 항상 MMST입니다.
- 그 이유는 해당 신장 트리보다 가중치의 합이 작은 신장 트리와 큰 신장 트리를 항상 만들 수 있기 때문입니다.
- 추가로 처음에 골랐던 MST에 포함되지 않는 간선 하나를 반드시 포함하는 최소 신장 트리를 구성하는 것으로도 답을 구할 수 있습니다.
- 시간 복잡도는 $O(M \log M)$ 입니다.

C. MMST



- 재미있는 별해 하나를 소개합니다.
- 앞서 소개한 문제의 조건에 의해, $N \leq M$ 이라면 임의의 신장 트리를 구성했을 때 MMST일 확률이 $\frac{1}{3}$ 이상입니다.
- 따라서, 임의의 랜덤 스패닝 트리를 K 번 구성했을 때 구성한 모든 스패닝 트리가 MST일 확률은 $(\frac{2}{3})^K$ 보다 작습니다.
- 이는 $K = 20$ 일 때 MMST를 한 번이라도 구성할 확률이 99.97% 정도임을 의미합니다.
- 시간 복잡도는 $O(KM \log M)$ 입니다.

D. 가채점

`aho_corasick`, `suffix_array`, `binary_search`
출제진 의도 – **Challenging**

- ✓ 제출 16번, 정답 3명 (정답률 18.75%)
- ✓ First Solve: **aeren**, 68분
- ✓ 출제자: 정윤성`slah007`

D. 가채점



- ✓ 문제를 요약하면, $S[1:i]$의 부분 문자열로 들어 있는 T_j의 개수를 $Count(i,j)$라 할 때 $\max_{1 \le j \le M}(Count(i,j) \times B[j])$가 A 이상이 되는 가장 작은 i를 찾으면 됩니다.
- ✓ 항상 $Count(i-1,j) \le Count(i,j)$이므로, i에 대한 이진탐색을 적용할 수 있습니다.

- ✓ i를 고정하면 이제 $\max_{1 \le j \le M} (Count(i, j) \times B[j]) \ge A$인지 판정하는 문제가 됩니다.
- ✓ KMP, Aho-Corasick 등으로 단순히 모든 T_j가 S와 매칭되는 모든 인덱스를 구하면 그 수만 NM개이기 때문에 너무 많은 시간이 걸리게 됩니다.
- ✓ Aho-Corasick을 적용할 때, $S[i]$를 끝으로 매칭되는 모든 T_j를 저장하는 대신 가장 긴 T_j만 저장한다고 생각해 봅시다. T_j의 Suffix에 해당하는 모든 다른 T_k들 역시 $S[i]$를 끝으로 매칭되게 됩니다.



- ✓ 즉, 매칭되는 횟수를 구할 때 매칭되는 가장 긴 T_j 만 저장한 뒤 failure node를 따라 가면서 짧은 다른 문자열들도 매칭되었다고 체크해주면 됩니다.
- ✓ Aho-Corasick의 모든 노드들을 탐색하는 DP 문제가 되므로, $O\left(\sum L_j\right)$ 에 해결할 수 있습니다.
- ✓ $\log N$ 개의 i 에 대해 판정하므로 총 $O\left(\sum L_i \log N\right)$ 이 됩니다.

D. 가채점



- 이외에도 Suffix Array를 이용하여 풀 수 있습니다. 이 경우 T_j가 온라인으로 주어지고 변경될 때도 $O(L_j \log^2 N)$에 S에 T_j가 몇 번 매칭되는지 셀 수 있습니다.
- S의 Suffix들 중 T_j로 시작하는 문자열은 Suffix Array 상에서 서로 연속된 구간에 속합니다.
- 즉, T_j로 시작하는 문자열들이 Suffix Array 상에서 어느 구간 $[left_j, right_j]$에 속하는지를 $O(L_j \log N)$에 구할 수 있습니다.
  - $left_j$는 사전 순으로 T_j 이상인 첫 지점, $right_j$는 사전 순으로 T_j 이상이면서 T_j로 시작하는 마지막 지점이 됩니다.
  - 변경 쿼리가 주어지는 경우, 두 문자열의 최장 공통 접미사의 길이를 해싱과 이진탐색으로 구할 수 있으므로 $O(\log N \log L_i)$에 처리 가능합니다.

- 이제 $S[1:i]$에서 각각의 T_j가 몇 번 매칭되는지는, $S[1:i]$의 Suffix Array 상에서의 순서 i개 값이 $[left_j, right_j]$에 몇 개 포함되는지와 같습니다.
- 2차원 직사각형 내의 점의 개수이므로 Persistent Segment Tree나 Merge Sort Tree를 이용해서 $O(\log N)$에 구할 수 있습니다.
- M개의 문자열을 모두 봐야 하므로 $O(M \log N)$이 됩니다.
- S의 Suffix Array를 $O(N \log N)$에 구하고 Segment Tree에 업데이트하면 전체 시간복잡도는 $O\left(N \log N + \sum L_j \log N + M \log N\right)$가 됩니다.

E1. 신기한 숫자 1

`number_theory`
출제진 의도 – **Medium**

- ✓ 제출 85번, 정답 24명 (정답률 12.96%)
- ✓ First Solve: **surface_03**, 36분
- ✓ 출제자: 장래오 `leo020630`

- ✓ 소수 p에 대해 $A = p^x$, $B = p^y$라 가정합시다.
- ✓ 1. $x > y$
  - 이 경우 조건을 만족하는 C가 존재하지 않습니다.
  - $GCD(A, B) = B$이므로 $GCD(A, C) = B$이기 위해서는 $B = C$여야 합니다.
  - 하지만 이는 $LCM(A, B) = A \neq B = LCM(B, C)$이므로 두 번째 조건에 위배됩니다.



- 2. $x \leq y$
  - 이 경우 $GCD(A, B) = A, LCM(A, B) = B$ 입니다.
  - 후자에 의해 $C = p^z$ 여야 하며, $x \leq z \leq y$ 여야 합니다.
  - 따라서 조건을 만족하는 C 는 $y - x + 1$ 개 존재합니다.



- 이제 $A = \prod {p_i}^{x_i}, B = \prod {p_i}^{y_i}$ 라 합시다.
- C 가 존재하기 위해서는 모든 i 에 대해 $x_i \leq y_i$ 이어야 합니다. 즉, B 가 A 로 나누어떨어져야 합니다.
- C 가 존재할 경우, 그 개수는 $\prod (y_i - x_i + 1)$ 입니다.
- 이는 $\frac{B}{A}$ 의 약수의 개수와 같습니다.
- 약수의 개수는 소인수분해 등을 통해 $O(\sqrt{B})$ 에 구할 수 있습니다.

E2. 신기한 숫자 2

`number_theory`
출제진 의도 – **Hard**

- ✓ 제출 31번, 정답 8명 (정답률 32.26%)
- ✓ First Solve: **riroan**, 67분
- ✓ 출제자: 장래오 `leo020630`



- 신기한 숫자 1의 풀이를 정리해봅시다.

$$f(A,B)=\begin{cases}\tau(\frac{B}{A}), & \text{if } A \mid B\\ 0 & \text{if } A \nmid B\end{cases}$$

- 이므로, $\sum_{i=1}^{N}\sum_{j=1}^{N} f(i,j)=\sum_{i=1}^{N}\sum_{d\mid i}\tau(\frac{i}{d})=\sum_{i=1}^{N}\sum_{d\mid i}\tau(d)$ 입니다.

- ✓ $\sum_{i=1}^{N}\sum_{d|i}\tau(d)$ 에서, 각 d에 대해 $\tau(d)$가 더해지는 횟수는 1 이상 N 이하의 정수 중 d를 약수로 가지는 정수의 개수와 같습니다.
- ✓ d를 약수로 가지는 정수의 개수는 d의 배수의 개수와 같으며, 이는 $\lfloor \frac{N}{d} \rfloor$로 쉽게 계산할 수 있습니다.



- 이제 $\sum_{d=1}^{N} \lfloor \frac{N}{d} \rfloor \tau(d)$ 를 계산합시다.
- 우선, $\lfloor \frac{N}{d} \rfloor$ 중 다른 값의 개수는 $O(\sqrt{N})$ 개임이 알려져 있습니다.
- 같은 $\lfloor \frac{N}{d} \rfloor$ 을 가지는 d끼리 묶은 후, 각 구간에 대해 $\lfloor \frac{N}{d} \rfloor$ 의 값을 a_i, d의 시작값과 끝값을 s_i, e_i 라 합시다.
- 이제 우리가 구해야 하는 값은 $\sum_{i=1}^{t} a_i \sum_{j=s_i}^{e_i} \tau(j)$ 입니다.

- ✓ $\sum_{j=s_i}^{e_i} \tau(j) = \sum_{j=1}^{e_i} \tau(j) - \sum_{j=1}^{s_i-1} \tau(j)$ 입니다.
- ✓ 첫 전개와 같은 원리로, $\sum_{i=1}^{N} \tau(i) = \sum_{i=1}^{N} \lfloor \frac{N}{i} \rfloor$ 이므로 $O(\sqrt{N})$ 시간에 구할 수 있습니다.



- 이제 이 풀이의 시간 복잡도를 구해봅시다.
- $O(\sqrt{N})$ 반복문을 $O(\sqrt{N})$ 번 도므로 $O(N)$ 일까요?
- 그렇지 않습니다. Harmonic Lemma에 의하면 계산량은 $O\left(\sum_{i=1}^{\sqrt{N}} \sqrt{i} + \sqrt{\frac{N}{i}}\right)$ 이며,

  적분판정법에 의해 해당 시간복잡도는 $O(N^{\frac{3}{4}})$ 입니다.

E2. 신기한 숫자 2



- 별해 하나를 소개합니다.
- 신기한 숫자 1 문제의 풀이에 의해, 조건을 만족하는 C 는 $A \mid C \mid B$ 여야 합니다.
- 세 수로 이루어진 순서쌍 $(i, j, k) = (A, \frac{C}{A}, \frac{B}{C})$ 를 생각해봅시다.
- 이 순서쌍이 조건을 만족하는 것은 세 수가 모두 정수이며, 세 수의 곱이 N 이하임과 동치입니다.
- 이러한 순서쌍의 개수를 작은 수부터 고정해나가며 세면 $O(N^{\frac{2}{3}})$ 에 문제를 해결할 수 있습니다.



F. 트리의 MEX

`small_to_large`
출제진 의도 – **Hard**

- ✓ 제출 36번, 정답 12명 (정답률 33.33%)
- ✓ First Solve: **edenooo**, 18분
- ✓ 출제자: 김범수 `qjatn0120`

F. 트리의 MEX



- 배열에 숫자가 계속 추가되는 상황에서, 배열의 MEX는 c++이나 python의 set을 이용하여 관리할 수 있습니다.
- 배열에 값이 추가될 때마다 MEX값은 단조 증가하기 때문에, 현재 MEX 값이 set에 존재한다면 MEX 값에 1을 계속 더해나가는 방식으로 MEX를 계산할 수 있습니다.
- 따라서 특정 서브 트리의 MEX는 서브 트리의 값들을 모두 set에 넣어 구할 수 있습니다.
- 하지만 서브 트리의 MEX를 구할 때마다 서브 트리의 모든 값들을 set에 넣어 MEX를 구한다면, 모든 서브 트리의 MEX를 구하는 데 걸리는 시간복잡도가 $O(N^2 \log N)$으로 시간초과가 나게 됩니다.

F. 트리의 MEX



- 서브 트리의 MEX를 구할 때, 자식 노드의 서브 트리 MEX 결과를 재활용하여 계산할 수 있습니다.
- 자식 노드의 서브 트리 값들이 담겨있는 set에 나머지 자식 노드의 서브 트리 값들을 추가해주면서 MEX를 계산해주면 됩니다.
- 이때 재활용할 자식 노드를 서브 트리가 가장 큰 자식 노드로 선택하는 small to large trick을 사용한다면, 각 값은 모든 서브 트리의 MEX를 추가하는 과정에서 set에 최대 $\log N$ 번만 추가됩니다.
- 따라서 set에 값을 추가하는 과정이 최대 $N \log N$ 번 발생하므로, 최종 시간복잡도는 $O(N \log^2 N)$ 으로 시간 제한안에 문제를 풀 수 있습니다.

G. 택배 색칠

`case_work`
출제진 의도 – **Easy**

- 제출 47번, 정답 18명 (정답률 38.30%)
- First Solve: **edenooo**, 32분
- 출제자: 장래오 `leo020630`



- 각 칸에 대해 색칠되지 않는 상자의 수를 구해봅시다.
- 이는 상하좌우 칸의 택배 상자 수에만 영향을 받습니다. 만약 상하좌우 칸이 격자 밖일 경우 상자 0 개가 쌓인 칸으로 생각하면 됩니다.
- 해당 칸에 쌓인 상자의 개수를 x_C, 상하좌우 칸에 쌓인 상자의 개수를 각각 x_U, x_D, x_L, x_R 이라 합시다.

G. 택배 색칠



- 각 칸의 색칠되지 않는 상자의 수를 계산하기 위해 경우를 나누어봅시다.
- 1. $x_C = 0$
  - 이 경우 상자가 없으므로 자명하게 색칠되지 않는 상자의 개수도 0 입니다.
- 2. $x_C \leq \min(x_U, x_D, x_L, x_R)$
  - 이 경우 해당 칸에 있는 상자들은 옆면이 다른 상자로 인해 막혀 있습니다.
  - 따라서 가장 위 상자의 윗면만 색칠되므로 답에 $\min(x_U, x_D, x_L, x_R) - 1$ 만큼이 추가됩니다.
- 3. $x_C > \min(x_U, x_D, x_L, x_R)$
  - 이 경우 높이가 $\min(x_U, x_D, x_L, x_R)$ 이하인 박스들이 색칠되지 않습니다.
  - 따라서 답에 $\min(x_U, x_D, x_L, x_R)$ 만큼이 추가됩니다.



- 각 칸에 대해 계산한 값을 모두 더해주면 정답을 구할 수 있습니다.
- 시간복잡도는 $O(NM)$ 이며, 잘못된 구현으로 인한 $O(NM \max(A_i))$ 풀이는 시간 초과를 받게 됩니다.
- 이외에도 답의 상한이 크기 때문에 integer overflow에 유의하여 문제를 해결해야 합니다.

H. 멋쟁이 포닉스

`case_work`
출제진 의도 – **Easy**

- ✓ 제출 32번, 정답 17명 (정답률 53.13%)
- ✓ First Solve: **dbgusdn012**, 53분
- ✓ 출제자: 장래오 `leo020630`



- 문제의 조건에 따라 포닉스가 고를 수 있는 옷의 조합은 아래의 두 가지 경우 중 하나입니다.
  - 흰색 마스크 - 검은색 티셔츠 - 흰색 바지
  - 검은색 마스크 - 흰색 티셔츠 - 검은색 바지
- 전자의 경우를 패턴 A, 후자의 경우를 패턴 B라고 하겠습니다.
- 포닉스는 두 가지 패턴을 번갈아서 선택해야 합니다.

H. 멋쟁이 포닉스



- ✓ 각 패턴을 고를 수 있는 최대 일수는 패턴에 속한 세 옷의 개수 중 최솟값입니다.
- ✓ 패턴 A의 최대 등장 가능 일수를 x, 패턴 B의 최대 등장 가능 일수를 y라고 합시다.
- ✓ 즉, $x = \min(M_W, T_B, P_W), y = \min(M_B, T_W, P_B)$ 입니다.
- ✓ 각 경우에 대해 아래와 같은 전략으로 패턴을 선택하는 것이 최적입니다.
  - $x > y$: A - B - A - $\cdots$ - A - B - A
  - $x = y$: A - B - A - $\cdots$ - A - B 또는 B - A - B - $\cdots$ - B - A
  - $x < y$: B - A - B - $\cdots$ - B - A - B
- ✓ 이를 종합하면 $x \neq y$일 경우 $2 \times \min(x, y) + 1$이, $x = y$일 경우 $2 \times x$가 답이 됩니다.

I. 잔디 예측하기

`graph_traversal`
출제진 의도 – **Medium**

- ✓ 제출 31번, 정답 11명 (정답률 33.33%)
- ✓ First Solve: **edenooo**, 42분
- ✓ 출제자: 정윤성[slah007]



- 우선 현재 잔디가 존재하는 모든 위치에는 미래에도 잔디가 위치해야 합니다. 그렇지 않다면 답은 항상 NO입니다.
- 현재에는 잔디가 존재하지 않았던 곳에 미래에 잔디가 존재할 때, 규칙에 따라 퍼져나간 결과로 가능한지 판정해야 합니다.

- 특정 칸에 잔디가 퍼지기 위해서는 거리가 D 이하인 칸들 중 하나가 그 이전에 잔디가 퍼져 있어야 합니다.
- 즉, 미래에 잔디가 퍼져 있는 모든 칸들은, 그 칸에서 거리가 D 이하인 잔디가 있는 칸들을 따라가면 현재 잔디가 있는 칸으로 도달할 수 있어야 합니다.
- 잔디가 위치하는 칸을 정점으로 하고 거리가 D 이하인 칸끼리만 이동이 가능한 간선이 있을 때 현재 잔디가 존재하는 칸들에서 시작하여 미래에 잔디가 퍼진 모든 칸을 도달 가능한지 묻는 그래프 탐색 문제를 풀면 됩니다.



- ✓ BFS, DFS 등의 그래프 탐색 알고리즘을 사용하여 풀면 됩니다.
- ✓ 정점의 수는 최대 N^2 개이고 각 정점마다 인접한 정점의 수는 최대 $2D(D+1)$ 개이므로 $O(V+E) = O(N^2D^2)$ 에 문제를 해결할 수 있습니다.

J. 단짠단짠 피자

`prefix_sum`, `case_work`
출제진 의도 – **Medium**

- ✓ 제출 14번, 정답 4명 (정답률 28.57%)
- ✓ First Solve: **edenooo**, 64분
- ✓ 출제자: 장래오[leo020630]



- 초기 상태에서 i 번 조각에서 시작해 K 개 조각의 합을 더한 값을 $f_0(i)$ 라 합시다.
- 또한, i 번 조각부터 K 조각동안 등장하는 홀수 번호 조각의 개수를 $o(i)$, 짝수 번호 조각의 개수를 $e(i)$ 라 합시다.
- 1번 쿼리로 인한 변화량의 합을 x, 2번 쿼리로 인한 변화량의 합을 y 라 하면 특정 시점에서 시작 조각을 i 로 하는 K 조각만큼의 맛의 합은 $f_0(i) + o(i) \times x + e(i) \times y$ 입니다.
- 즉, $(o(i), e(i))$ 값이 같은 시작 조각들끼리는 $f_0(i)$ 가 가장 큰 값만 고려해주어도 됩니다.

- 놀랍게도, 서로 다른 $(o(i), e(i))$ 쌍의 개수는 최대 2가지입니다.
- 1. $N =$ 짝수, $K =$ 짝수
  - 모든 i에 대해 $(o(i), e(i)) = \left(\frac{K}{2}, \frac{K}{2}\right)$ 입니다.
- 2. $N =$ 짝수, $K =$ 홀수
  - 이 경우 i의 홀짝성에 의해 두 가지 경우로 나뉩니다.
  - i가 홀수인 경우 $(o(i), e(i)) = \left(\frac{K+1}{2}, \frac{K-1}{2}\right)$
  - i가 짝수인 경우 $(o(i), e(i)) = \left(\frac{K-1}{2}, \frac{K+1}{2}\right)$

- ✓ N 이 홀수인 경우 N 번 조각과 1 번 조각이 모두 홀수이기 때문에 조금 복잡해집니다.
- ✓ 3. $N =$ 홀수, $K =$ 짝수
  - $i + K - 1 > N$ 이며 i 가 홀수인 경우 $(o(i), e(i)) = \left(\frac{K}{2} + 1, \frac{K}{2} - 1\right)$
  - 그 외의 경우 $(o(i), e(i)) = \left(\frac{K}{2}, \frac{K}{2}\right)$
- ✓ 4. $N =$ 홀수, $K =$ 홀수
  - $i + K - 1 \leq N$ 이며 i 가 짝수인 경우 $(o(i), e(i)) = \left(\frac{K-1}{2}, \frac{K+1}{2}\right)$
  - 그 외의 경우 $(o(i), e(i)) = \left(\frac{K+1}{2}, \frac{K-1}{2}\right)$



- ✓ 따라서 각 경우에 대해 $f_0(i)$ 의 최댓값을 저장해준 후 각 질문에서의 x, y 값에 따라 최대 2가지 중 더 큰 것을 출력해주면 됩니다.
- ✓ $f_0(i)$ 는 누적 합 또는 슬라이딩 윈도우 기법으로 구해줄 수 있으며, 비슷한 원리를 이용하면 $o(i), e(i)$ 또한 조건문을 사용하지 않고 편하게 구할 수 있습니다.
- ✓ 시간 복잡도는 $O(N+Q)$ 입니다.



K. 포닉스의 신비한 분자 보고서

`geometry`
출제진 의도 – **Hard**

- ✓ 제출 13번, 정답 3명 (정답률 23.08%)
- ✓ First Solve: **jhnah917**, 166분
- ✓ 출제자: 정윤성[slah007]



- ✓ 두 단순 다각형이 평행이동과 회전이동만을 통해 서로 같아질 수 있는지를 묻는 문제입니다.
- ✓ 실수 연산을 쓰면 오차 때문에 틀리거나 속도가 느려 **시간 초과**를 받기 쉽기 때문에 정수로만 값을 비교하는 것이 좋습니다.



- ✓ 두 단순 다각형이 평행이동과 회전이동만을 통해 같아질 수 있다면, 두 다각형의 꼭짓점의 개수가 같고, 모든 변의 길이가 순서대로 일치하고, 모든 각이 순서대로 일치해야 합니다.
- ✓ 꼭짓점의 수를 n이라 할 때, 두 다각형의 꼭짓점이 반시계방향으로 $P_1, P_2, \dots P_n$과 $Q_1, Q_2, \dots Q_n$이라 합시다.
- ✓ 일치하게 되는 두 꼭짓점을 P_1과 Q_i라 하면 모든 $0 \le k < n$에서
  - 변 $P_{k+1}P_{k+2}$와 변 $Q_{k+i}Q_{k+i+1}$의 길이가 같아야 합니다.
  - $\angle P_{k+1}P_{k+2}P_{k+3}$과 $\angle Q_{k+i}Q_{k+i+1}Q_{k+i+2}$의 크기와 방향성이 같아야 합니다.
- ✓ 모든 $1 \le i \le n$을 보며 만족하는 Q_i가 존재하는지 검사하면 됩니다.



- ✓ 변의 길이는 실제로 두 점 사이의 길이의 제곱을 비교하면 됩니다.
- ✓ 각의 크기와 방향성이 같은지는 실제로 실수인 각도를 구하는 대신 삼각형의 합동과 외적으로 판정할 수 있습니다.
- ✓ $\angle P_{k+1}P_{k+2}P_{k+3}$ 과 $\angle Q_{k+i}Q_{k+i+1}Q_{k+i+2}$ 의 크기가 같은 조건은 $\triangle P_{k+1}P_{k+2}P_{k+3}$ 와 $\triangle Q_{k+i}Q_{k+i+1}Q_{k+i+2}$ 가 합동인 조건과 필요충분하고, 이미 두 변이 같다면 선분 $P_{k+1}P_{k+3}$ 과 선분 $Q_{k+i}Q_{k+i+2}$ 까지 세 변이 모두 같은지 판정하면 됩니다.
- ✓ 세 점 $P_{k+1}P_{k+2}P_{k+3}$ 과 세 점 $Q_{k+i}Q_{k+i+1}Q_{k+i+2}$ 의 방향성은 벡터 $P_{k+1}P_{k+2}$ 와 $P_{k+2}P_{k+3}$ 의 외적의 z성분, 벡터 $Q_{k+i}Q_{k+i+1}$ 와 $Q_{k+i+1}Q_{k+i+2}$ 의 외적의 z성분의 부호가 같은지 판정하면 됩니다.
- ✓ 이외에도 다양한 다른 방식으로 판정 가능합니다.



- ✓ 최종적으로 회전이동과 평행이동을 통해 같아지는 집합끼리 분류하면 됩니다.
- ✓ 모든 다각형 쌍이 서로 같은지 다른지 판정하고 서로 같다면 간선으로 연결하면 최종 그래프에서 각각의 Connected Component가 다각형의 한 종류가 됩니다.
- ✓ Union-Find Set이나 단순 그래프 탐색으로 모든 Connected Component의 개수와 크기를 구할 수 있습니다.
- ✓ 모든 다각형의 꼭짓점의 수가 n으로 같을 때 가장 많은 비교를 해야 하고, 이때 다각형 한 쌍을 비교하기 위해 시작 꼭짓점 n개에 대해 각각 변과 각 $O(n)$개를 비교해야 합니다.
- ✓ 모든 꼭짓점 수의 합을 C라 하면 다각형의 수는 $\frac{C}{n}$이므로 비교해야 하는 쌍의 수가 $O\left(\left(\frac{C}{n}\right)^2\right)$, 각각 비교에 $O(n^2)$이 걸리므로 총 $O(C^2)$에 해결할 수 있습니다.

L. 포커

`dynamic_programming`
출제진 의도 – **Hard**

- ✓ 제출 8번, 정답 4명 (정답률 50.00%)
- ✓ First Solve: **jgs03177**, 131분
- ✓ 출제자: 김범수`qjatn0120`



- 먼저 플러시를 만들기 위해 중요한 건, 페어, 트리플, 포카드를 만들고 문양 별로 몇 장의 카드가 남았냐는지 입니다. 만약 같은 문양이 10장 이상 남았다면 플러시를 2족보 만들 수 있고, 5장 이상 남았다면, 플러시를 1족보 만들 수 있습니다.
- 따라서 페어, 트리플, 포카드를 만들 때, 문양 별로 몇 장의 카드를 사용하였는지가 중요합니다. 그러므로 아래와 같이 문양 별 남은 개수를 이용하여 dp식을 세웁니다.
- $dp[n][a][b][c][d] =$ 적혀있는 숫자가 1이상 n이하인 카드 중 스페이드를 a장, 클로버를 b장, 다이아몬드를 c장, 하트를 d장, 사용하여 만들 수 있는 최고 점수

- 숫자가 N인 카드를 활용하여 만들 수 있는 족보(페어, 트리플, 포카드 중 하나)를 모두 고려해야 합니다.
- 숫자가 N이면서 스페이드인 카드를 a_n, 클로버인 카드를 b_n, 다이아몬드인 카드를 c_n, 하트인 카드를 d_n 장 활용하여 점수가 x 인 족보를 만들 수 있다고 합시다. 물론 a_n, b_n,c_n, d_n 은 모두 0 또는 1입니다. 그렇다면 아래와 같은 점화식을 얻을 수 있습니다.
-

$$dp[n][a][b][c][d] = \max(dp[n-1][a-a_n][b-b_n][c-c_n][d-d_n] + x)$$

- ✓ 점화식을 이용하여 dp field를 모두 채워넣어 플러시를 제외한 족보로 얻을 수 있는 최대 점수를 알 수 있습니다. 따라서 각 dp값에서 남은 카드로 플러시를 만들어 얻을 수 있는 점수를 구한 뒤 이 중 최댓값을 구하여 답을 구할 수 있습니다.
- ✓ 계산해야 할 dp field의 크기는 약 $13^5 \approx 400\,000$이고, 각 숫자마다 가능한 페어, 트리플, 포카드의 종류는 11종류 이므로 대략 $4\,000\,000$번의 연산 안에 dp field를 모두 채울 수 있습니다. 따라서 제한 시간 안에 정답을 구할 수 있습니다.

M. 포스택

`stack`, `greedy`
출제진 의도 – **Easy**

- 제출 36번, 정답 24명 (정답률 66.67%)
- First Solve: **dbgusdn012**, 43분
- 출제자: 김범수 `qjatn0120`, 장래오 `leo020630`



- 스택에서 수를 꺼내는 상황을 먼저 생각합시다.
- 네 개의 스택에서 수를 적절히 꺼내 오름차순 정렬할 수 있음은 각 스택에 수들이 오름차순으로 저장되어 있음과 동치입니다.
- 따라서, 모든 스택을 오름차순으로 유지하며 모든 수를 스택에 넣을 수 있을지 판별하는 문제가 됩니다.



- 다음과 같은 전략을 반복하는 것이 최적입니다.
- 만약 현재 넣어야 하는 수를 넣을 수 있는 스택이 있다면, 그러한 스택 중 top에 위치한 값이 가장 큰 스택에 수를 삽입한다. (빈 스택의 top 값은 0으로 생각한다.)
- 이를 반복하며 스택에 수를 넣을 수 없는 상황이 생긴다면 정답은 NO, 모든 수를 넣을 수 있다면 YES입니다.
- 시간 복잡도는 $O(N)$ 입니다.