제3회 피갤컵 풀이

by

Official Solutions

문제		의도한 난이도	출제자
A	피갤컵 들어왔으면 이글부터 봐라	**Very Easy**	pyb1031
B	지금부터 완장 찬양을 시작하겠습니다	**Easy**	ksoosung77
C	폭탄의 악마	**Easy**	nabina1395
D	Sonic Wave	**Medium**	wapas
E	정사각형 따먹기	**Medium**	bubbler
F	체크무늬 찾기	**Hard**	yoonoomilk
G	0.4% 확률로 애드혹 구성적 문제	**Hard**	lunarlity
H	피돌이 vs 피붕이 2	**Hard**	pyb1031
I	벽력일섬	**Hard**	dong5995
J	"참나... 고작 gpt가 문제를 만들어봤자..."	**Challenging**	pyb1031

A. 피갤컵 들어왔으면 이글부터 봐라

`implementation`
출제진 의도 – **Very Easy**

- 제출 397번, 정답자 308명 (정답률 78.84%)
- 처음 푼 사람: **seawon0808**, 0분
- 출제자: pyb1031

- 이-글
- 길이가 5인 연속부분문자열을 모두 확인하면서 몇글자를 바꿔야 'eagle'이 되는지 확인한 후 그 중 최솟값을 구하면 됩니다.

B. 지금부터 완장 찬양을 시작하겠습니다

Implementation
출제진 의도 – **Easy**

- 제출 438번, 정답자 299명 (정답률 69.63%)
- 처음 푼 사람: **kdg870918**, 1분
- 출제자: ksoosung77

- ✓ 지문에 요구하는 대로 K일 이상 방문하지 않았는지 확인하기 위해서 마지막 1로부터 K일 이상 지났는지 각 날짜가 지날 때마다 확인하면 됩니다.
- ✓ 다른 방법으로는, 0이 연속으로 되는 길이가 K 이상인 경우가 있는지 확인하는 방법도 있습니다.

C. 폭탄의 악마

`linearity_of_expectation`, `prime_factorization`
출제진 의도 – **Medium**

- 제출 353번, 정답자 131명 (정답률 38.81%)
- 처음 푼 사람: **0917ba**, 7분
- 출제자: `nabina1395`

C. 폭탄의 악마

- 기댓값의 선형성에 의해 수열의 합의 기댓값 = 각 원소의 기댓값의 합입니다.
- Subtask 1. 원소의 합을 입력받은 그대로 출력하면 됩니다.
- Subtask 2.
- 한 원소에는 폭발이 0번 혹은 1번 일어납니다.
- 0번 일어난 경우 기댓값은 처음 그대로입니다.
- 1번 일어난 경우 기댓값은, A_i 의 소인수를 $p_1, p_2, p_3 \dots p_k$ 라고 하면, $\frac{\sum_{i=1}^{k} p_i}{k}$ 입니다.
- 폭발의 여부에 따라 각 원소들의 기댓값들을 더해주면 됩니다.
- 에라토스테네스의 체를 사용하거나, 루트 복잡도의 소인수분해를 사용할 수 있습니다.

- Subtask 3.
- 폭발이 일어난 원소는 소수가 됩니다.
- 소수의 소인수는 자기 자신 하나입니다. 따라서, 폭발이 일어나더라도 기댓값이 변하지 않습니다.
- 따라서, 폭발이 2번 이상 일어나더라도 기댓값은 1번 일어난 경우와 같습니다.
- imos법, 스위핑 등으로 각 원소에 폭발이 일어났는지를 확인할 수 있습니다.

D. Sonic Wave

`parity, greedy, constructive`
출제진 의도 – **Medium**

- 제출 164번, 정답자 92명 (정답률 59.14%)
- 처음 푼 사람: **trashmouse0524**, 9분
- 출제자: wapas

D. Sonic Wave

- 시점 T 에서 한 스피커가 어떤 칸들을 영향을 주는지 관찰하면 "자신과 짝수칸 만큼 떨어진 칸만" 혹은 "자신과 홀수칸 만큼 떨어진 칸만" 영향을 줍니다.
- 따라서 배열 y 에 대해서 짝수 칸과 홀수 칸은 독립적인 문제로 쪼개서 풀 수 있습니다.
- 배열 y 에서 홀수 인덱스만 모은 배열 $O = [y_1, y_3, y_5, \cdots]$, 짝수 인덱스만 모은 배열 $E = [y_2, y_4, y_6, \cdots]$ 를 정의합니다.
- 해설은 짝수 기준으로 해설합니다. 홀수의 경우도 풀이가 동일합니다.

D. Sonic Wave

- 배열 E에 대해 배열 P를 정의합니다.
  - $P_1 = E_1$
  - $P_k = E_{k-1} + E_k$
  - $P = [E_1, E_1 + E_2, E_2 + E_3, \cdots]$
- 배열 P에서 0이 아닌 개수가 곧 최소 스피커 개수입니다.
- 왜냐하면 어떤 스피커든 배열 P에서 설치한 위치 x_j에서만 흔적을 남기기 때문입니다.

- ✓ 홀수칸은 배제했으므로 P에서는 $\phi_j = 1, \phi_j = 3$인(또는 $\phi_j = 0, \phi_j = 2$인) 경우만이 배열 P에 영향을 줄 수 있습니다.
- ✓ 그러므로 탐욕적으로 P_i가 0이 아닐 때, $v_j = P_i$인 스피커를 배치하는게 항상 이득입니다.
- ✓ 그러므로 배열 P를 살펴보며 0이 아닌 지점이 나오면 그 위치에 스피커를 설치하는 식으로 $O(N)$ 시간복잡도로 해를 구성할 수 있습니다.
- ✓ 홀짝$(\mathrm{mod}\ 2)$이 아닌 4주기$(\mathrm{mod}\ 4)$로도 동일한 논리로 풀 수 있습니다. 이는 스피커가 $\sin$ 함수 형태로 각 칸에 영향을 주기 때문입니다.

E. 정사각형 따먹기

`ad_hoc`, `constructive`
출제진 의도 – **Medium**

- ✓ 제출 314번, 정답자 94명 (정답률 30.57%)
- ✓ 처음 푼 사람: **kdy40929**, 6분
- ✓ 출제자: `bubbler`

- 먼저, 더 이상 제거할 정사각형이 없는 상태에서의 총 둘레의 길이는 원래의 둘레와 동일해야 함을 관찰합니다.
- 그러한 둘레의 길이는 $2(W+H)$ 입니다.
- 따로 떨어진 단위 정사각형 하나의 둘레가 4 이므로, 둘레 $2(W+H)$ 를 만들기 위해서는 적어도 $\left\lceil \frac{W+H}{2} \right\rceil$ 개의 단위 정사각형이 필요합니다.

- $W \times 1$ 직사각형에서 시작하여 이만큼의 단위 정사각형을 남기기 위해서는, 왼쪽에서부터 2번째, 4번째, ...의 단위 정사각형을 제거하여 달성할 수 있습니다.
- 이때 W가 짝수이더라도 W번째 단위 직사각형은 제거할 수 없으며, $W-2$번째까지 제거하면 필요한 단위 정사각형의 개수를 얻을 수 있습니다.
- $W \times H$ 직사각형의 경우, 예를 들어 맨 오른쪽 아래에서 시작하여 한 줄씩, 한 줄에서 $W-1$개씩 제거하는 것으로 $W+H-1$개의 단위 정사각형으로 이루어진 띠를 남길 수 있고, 그 띠에 대해 위의 $W \times 1$과 동일한 풀이를 적용하여 최적을 달성할 수 있습니다.

F. 체크무늬 찾기

`stack`
출제진 의도 – **Hard**

- 제출 114번, 정답자 50명 (정답률 43.86%)
- 처음 푼 사람: **dlalswp25**, 21분
- 출제자: `yoonoomilk`

- $(row + column) \mod 2 = 1$ 인 칸의 색을 전부 반전시키면, 같은 색으로만 이루어진 직사각형의 개수를 세는 문제로 바꿀 수 있습니다.
- 각 행을 기준으로 히스토그램을 만들 수 있습니다.
- 히스토그램에서 한 열의 높이는 기준에서 위쪽으로 연속된 같은 색 칸의 개수입니다.
- 히스토그램에서의 모든 직사각형 개수는 단조 스택을 이용하면 $O(N)$ 에 계산할 수 있습니다.
- 총 시간복잡도는 $O(N^2)$ 입니다.

G. 0.4% 확률로 애드혹 구성적 문제

`ad_hoc`, `constructive`, `binary_search`, `sqrt_decomposition`
출제진 의도 – **Hard**

- 제출 62번, 정답자 2명 (정답률 3.22%)
- 처음 푼 사람: **jjang36524**, 117분
- 출제자: `lunarlity`

G. 0.4% 확률로 애드혹 구성적 문제 (Subtask 1/2)

- ✓ Subtask 1. 가능한 $N!$개의 순열을 인터랙션해보며 결과가 0인 것을 찾으면 됩니다.
- ✓ Subtask 2. 가능한 $N!$개의 순열을 나열 후, 랜덤한 순열을 인터랙션하여 해당 결과가 나올 수 없는 숨겨진 순열을 지우면 생각보다 매우 빠른 횟수(일반적으로 약 3회) 안에 유일하게 순열이 결정됩니다.

- ✓ 약 $2N$ 번의 쿼리 횟수를 사용하는 풀이부터 보겠습니다.
- ✓ 인터랙션을 통해 얻을 수 있는 값은 덧셈 구조이므로, Q 의 많은 위치를 동시에 바꾸는 것은 좋지 않습니다.
- ✓ 또한 절댓값은 다루기 까다롭기 때문에, 절댓값의 영향을 단순하게 만드는 $Q_i = 1,\ Q_j = N$ 과 같은 위치가 중요한 역할을 할 것임을 예상할 수 있습니다.

G. 0.4% 확률로 애드혹 구성적 문제 (Fulltask)

- ✓ 순열 Q에 대해 인터랙션의 결과를 $[Q]$라고 표기하겠습니다.
- ✓ 이제 $Q = [\cdots, 1, \cdots, N, \cdots]$, $Q' = [\cdots, N, \cdots, 1, \cdots]$ 처럼 두 위치 i, j에서만 값을 바꾼 두 순열을 생각합시다. 즉, $Q_i = 1,\ Q_j = N$ 이고 $Q'_i = N,\ Q'_j = 1$ 이며, 나머지 위치는 모두 동일합니다.
- ✓ 이때 1과 N을 제외한 나머지 위치의 값은 동일하므로, 해당 항들은 모두 상쇄됩니다.
- ✓ 따라서 식을 정리하면 $[Q] - [Q'] = 2(P_i - P_j)$를 얻을 수 있습니다.
- ✓ 이를 이용하면 $P_1 - P_2, P_1 - P_3, \cdots, P_1 - P_N$을 얻어줄 수 있고, P는 순열이므로 P의 모든 원소 값을 구할 수 있습니다.
- ✓ 사용되는 쿼리 수는 $2 \times (N-1) = 2N - 2$입니다.

G. 0.4% 확률로 애드혹 구성적 문제 (Fulltask)

- ✓ 이전 방법에서 쿼리 수가 많이 필요했던 이유는, 어떤 순열 Q에 대한 $[Q]$ 값을 비교하기 위해 매번 새로운 순열을 추가로 인터랙션해야 했기 때문입니다.
- ✓ 하나의 순열로 더 많은걸 비교할 순 없을까요?
- ✓ N 대신 어떤 수 x를 사용한다고 생각하겠습니다.
- ✓ 식을 정리하면, $[Q] - [Q'] = (P_i - 1) + |P_j - x| - |P_i - x| - (P_j - 1)$가 됩니다.
- ✓ 여기서 $i = 1$로 고정하면 $(P_1 - P_j) + |P_j - x| - |P_1 - x|$가 되는데, P_1 값을 알고있다면 식이 더 쉽게 정리가 될 것 같습니다.

G. 0.4% 확률로 애드혹 구성적 문제 (Fulltask)

- P_1 을 알기 위해 이분탐색을 사용합니다.
- 우리는 $P_1 - P_2$ 를 알 수 있습니다. $(2N$ 풀이 참고$)$
- 위 식 $(P_i - P_j) + |P_j - x| - |P_i - x|$ 에서 값이 0 이 되려면, 식을 정리하면 $x \leq \min(P_i, P_j)$ 여야 합니다.
- 이를 이분탐색하면 $\min(P_1, P_2)$ 를 알 수 있고, $P_1 - P_2$ 를 알기 때문에 P_1, P_2 를 확정적으로 구할 수 있습니다.
- 이때 사용되는 쿼리 수는 $\log N + 2$ 번입니다.

G. 0.4% 확률로 애드혹 구성적 문제 (Fulltask)

- 다시 원래 문제로 돌아와봅시다.
- 우리는 $(P_1 - P_j) + |P_j - x| - |P_1 - x|$ 에서 P_1 값을 알고있으므로, 값을 알고 있는 상수 C 를 가정하고 식을 정리해보면 $|P_j - x| - P_j + C$ 가 됩니다.
- $P_j \geq x$ 라면 $C - x$ 가 되서 의미가 없지만, $P_j < x$ 라면 $-2P_j + (x + C)$ 가 되서 P_j 값을 알 수 있게 됩니다.
- 이를 이용하면, $P_j < x$ (즉, 차이가 $C - x$ 가 아님)인 모든 x 에 대해 P_j 값을 구해낼 수 있고, 그게 아니라면 $P_j \geq x$ 라는 것도 알 수 있습니다.
- 적당한 수 B 를 잡고, x 를 $N - B + 1, N - B + 2, \cdots, N$ 으로 잡으면, 최악의 경우에도 $N - B$ 개의 원소는 확정지을 수 있습니다.

G. 0.4% 확률로 애드혹 구성적 문제 (Fulltask)

- 이를 이용하기 위해선 $Q = [1, \cdots, (N-B+1, \cdots, x, \cdots, N), \cdots]$로 잡고, $Q' = [x, \cdots, (N-B+1, \cdots, 1, \cdots, N), \cdots]$를 쿼리하면 됩니다. $()$는 이해의 편의를 위해 넣었습니다.
- 비교 대상이 되는 Q는 N/B번 쿼리하면 되며, Q'은 N번 쿼리하면 됩니다.
- 총 $N + N/B$번의 쿼리를 사용하며, $N - B$개의 값을 확정적으로 알 수 있습니다.
- 이제 모르는 B개의 원소들은, 적당히 잘 하면 약 B개의 쿼리만을 사용하여 구할 수 있습니다.
- 전체 쿼리 수는 $N + N/B + B + \log N$이며, $B = \sqrt{N}$으로 잡으면 $N + 2\sqrt{N} + \log N$번의 쿼리에 풀 수 있습니다. $N = 1000$을 대입했을 때, 약 1073번의 쿼리를 사용합니다.
- TMI) 무작위화를 사용하면, 쿼리 수는 평균적으로 $N + \sqrt{2}\sqrt{N} + \log N$번입니다.

H. 피돌이 vs 피붕이 2

`game thoery`, `prefix sum`
출제진 의도 – **Hard**

- 제출 65번, 정답자 17명 (정답률 26.15%)
- 처음 푼 사람: **elaina_cpp**, 30분
- 출제자: pyb1031

H. 피돌이 vs 피붕이 2

- 먼저 x 이상인 수를 1, x 미만인 수를 0이라고 하고 최적으로 플레이할 때 1이 남는다면 답은 x 이상이 됩니다.
- 따라서 0과 1로만 이뤄진 수열에서 문제를 풀 수 있다면 전체 문제를 풀 수 있습니다.
- 먼저 나이브를 돌려보면 0과 1의 개수가 다른 경우, 더 많은 쪽이 답이 된다는 강력한 심증을 얻을 수 있습니다.
- 따라서 N 이 홀수일 때는 $(N+1)/2$ 를 출력하면 되지만 짝수일 때 $N/2$ 와 $N/2+1$ 중에서 판단을 해야합니다.
- 0과 1의 개수가 같은 경우를 생각해봅시다.
- 두 수열로 나눌 때 두 수열 모두 0과 1의 개수가 같도록 나눠야 합니다. 그렇지 않다면 상대가 둘 중 자신한테 유리한걸 가져갈 수 있습니다.

- 즉, 1을 +1, 0을 -1로 하는 누적합을 생각해보면 누적합이 0이 되는 부분을 기준으로 나눠야 합니다.
- 이 때 중요한것은 누적합에서 0이 나오는 위치의 개수뿐입니다. 잘 생각해보면 그 개수의 홀짝성에 따라 결과가 바뀜을 알 수 있습니다.
- 결론적으로, 1의 개수가 0의 개수보다 많거나 1의 개수가 0의 개수와 같고 누적합의 0의 개수가 홀수이면 결과가 1이 됩니다. 이는 어렵지 않게 증명할 수 있습니다.
- 따라서 N 이 짝수일 때도 답을 구할 수 있습니다.

I. 벽력일섬

`segtree`, `lazyprop`, `difference_array`, `binary_search`, `euler_characteristic`, `offline_queries`, `grid_graph`

출제진 의도 – **Hard**

- 제출 39번, 정답자 5명 (정답률 15.39%)
- 처음 푼 사람: **leinad2**, 90분
- 출제자: dong5995

- 칸 A의 승객이 칸 B의 승객보다 많거나 같다면, 승객이 희생되더라도 그 관계는 유지됩니다.
- 따라서, 쿼리를 전처리하면 승객이 희생되지 않았을 때의 lo, hi 값으로 환원할 수 있습니다.
- 이 전처리에는 차분배열트릭을 이용한 세그먼트 트리 또는 레이지 세그먼트 트리, 이분 탐색을 사용할 수 있습니다.
- 그러면 $2 \times N$ 크기의 격자 그래프에서 연결 요소의 개수를 세는 문제가 됩니다.

- 평면 그래프에서 오일러 지표를 생각하면, (연결 요소의 개수) = (정점의 수) - (간선의 수) + (면의 수)입니다.
- $2 \times N$ 크기의 격자 그래프에서 면은 정사각형 형태밖에 없습니다.
- 전처리해둔 쿼리들을 정렬하여 세그먼트 트리를 이용해 스위핑해주면 필요한 값들을 구할 수 있습니다.
- 한 칸의 최대 승객 수를 K라 하면, $O((N+M)logK + MlogM + K)$에 풀 수 있습니다.
- 이 풀이는 오프라인 쿼리이나, 퍼시스턴트 세그먼트 트리를 이용하면 온라인 쿼리로도 풀이가 가능합니다.

J. "참나... 고작 gpt가 문제를 만들어봤자..."

`dp, greedy`
출제진 의도 – **Challenging**

- ✓ 제출 12번, 정답자 0명 (정답률 0.00%)
- ✓ 처음 푼 사람: -, -분
- ✓ 출제자: pyb1031

- 우선 $f(S)$의 값을 빠르게 구할 필요가 있습니다.
- 어떤 두 위치 i, j를 한 획으로 칠할 때 i와 j를 잇는 선을 잇는다고 생각해봅시다.
- 잘 생각해보면 이런 선들은 서로 포함되거나 완전히 겹치지 않거나 둘 중 하나입니다.
- 선들로 이어진 컴포넌트 내에서 인접한 정점을 잇는 선만 남긴다고 생각해봅시다. 예를들어 i, j, k가 한 컴포넌트에 있고 $i < j < k$라면 i와 j를 잇는 선과 j와 k를 남기는 선만 남기는 식입니다.
- $f(S) = |S|$-(그은 선의 개수) 이므로 목표는 이런 선을 최대한 많이 긋는것입니다.

J. "참나... 고작 gpt가 문제를 만들어봤자..."

- 우선 S의 인접한 문자가 모두 다르다고 가정합시다.
- aba의 꼴, 즉 $s_i = s_{i+2} \neq s_{i+1}$인 i가 있다면 i와 $i+2$를 잇는것이 항상 이득입니다.
- 손해인 경우는 $i+1$번째와 다른 위치에 2개가 연결되어 있는 경우인데, 이 경우 $i+1$번째와 연결돼있는 두 위치를 $j, k\ (j<k)$라 하면 $j<i, k>i+2$인데 $i+1$번째 위치의 연결을 끊고 j번째와 k번째, i번째와 $i+2$번째를 연결해주면 손해가 아니게 됩니다.
- aba의 꼴에서 두 a를 연결하는것은 뒤의 ba를 제거하는 것으로도 볼 수 있습니다.
- R, G, B를 각각 0, 1, 2로 보고 $(s_{i+1} - s_i) \bmod 3$의 값을 생각해봅시다. 그러면 인접해있으며 차이가 1, 2인것을 없앤다 라고도 볼 수 있습니다.
- S에서 차이가 1인 위치의 개수를 x, 차이가 2인 위치의 개수를 y라 합시다.

- ✓ 1과 2가 모두 남아있다면 그런 곳은 항상 존재하므로 그런 시행은 $\min(x, y)$ 번 할 수 있습니다.
- ✓ 이 후 남은 S 는 ...RGBRGB... 또는 ...RBGRBG... 의 형태인데 이 경우에서 그을 수 있는 최대 횟수는 $\lfloor (|S| - 1)/3 \rfloor$ 이 됩니다.
- ✓ 결과적으로, 식을 정리해보면 $f(S) = \lceil (\max(2x + y, x + 2y))/3 \rceil + 1$ 이라는 사실을 알 수 있습니다.
- ✓ x, y 에 각각 가중치가 $2, 1$ 인 경우, $1, 2$ 인 경우 모두 dp로 계산해 최댓값을 구하면 $O(N)$ 에 문제를 해결할 수 있습니다.