

제 8회 인하대학교 프로그래밍 경진대회

대회 결과 및 해설

주최



인하대학교 컴퓨터공학과
Department of Computer Engineering

주관



후원

NAVER 



A. 경로당핑크 2077

의도한 난이도 : **EASY**

#구현

#출제 : 김현민

제출 104번, 정답 54명 (정답률 57.69%)

처음 풀 사람 : **강지원**, 3분

- Q. 사용자 이름의 마지막 글자에 받침이 존재하는가?
- 사용자 이름의 마지막 글자 : 문자열 s의 마지막 2글자 혹은 3글자
 - 2글자 : 자음 + 모음 (받침이 없다)
 - 3글자 : 자음 + 모음 + 자음 (받침이 있다)
- 문자열 s의 마지막 알파벳만 봤을 때
 - `yuiophjklbnm` : 모음, 2글자 (받침이 없다)
 - `qwertasdfgzxcv` : 자음, 3글자 (받침이 있다)

B. 너의 평점은

의도한 난이도 : **EASY**

#구현

#출제 : 정치훈

제출 89번, 정답 50명 (정답률 57.30%)

처음 풀 사람 : **최준형**, 5분

B. 너의 평점은

- 조건문과 반복문을 사용할 수 있고,
정수형 · 실수형 · 문자열 변수를 적절하게 활용할 수 있는가?
- 반복문을 이용해 20줄의 입력을 받는다
- 과목명은 의미가 없으니 버리고, 학점과 등급은 실수형 자료형으로 변환한다
 - 단, 등급이 P가 아닌지 조건문으로 판단하자
- 이후 문제에 나온 식에 따라 전공 평점을 계산하고 출력한다

C. 바벨탑의 저주

의도한 난이도 : **CHALLENGING**

#DFS

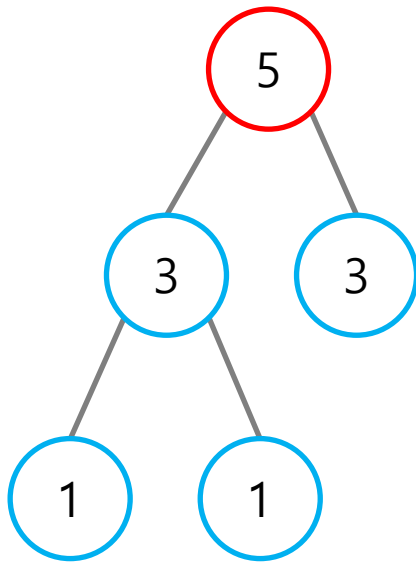
#manacher

#출제 : 김현민

제출 54번, 정답 0명 (정답률 0.00%)

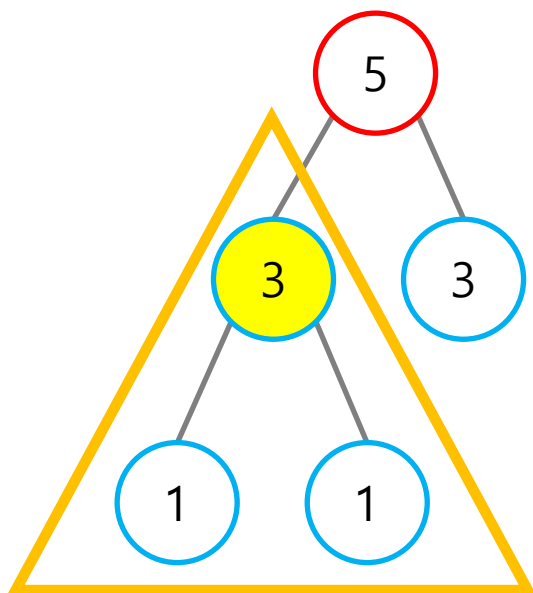
처음 풀 사람 : **N/A**

- DFS로 트리를 순회하며 (적힌 정수, 깊이) 쌍을 배열에 순서대로 기록한다
 - 번호가 작은 자식을 우선으로 순회한다
 - 중위 순회



5	3	1	3	1	3	5	3	5
1	2	3	2	3	2	1	2	1

- 각 정점이 루트인 서브 트리는 배열에서 하나의 연속한 구간으로 나타난다
- Manacher, 해싱 등 $O(N)$ 전처리를 거쳐 각 구간이 대칭인지 $O(1)$ 에 판정할 수 있다



5	3	1	3	1	3	5	3	5
1	2	3	2	3	2	1	2	1

D. 새벽의 탐정 게임

의도한 난이도 : **HARD**

#BFS

#출제 : 김현민

제출 78번, 정답 16명 (정답률 20.51%)

처음 푼 사람 : **최을호**, 40분

D. 새벽의 탐정 게임

- 크기가 $6 \times N \times M$ 인 정수형 배열로 감옥의 모든 상태를 표현할 수 있다
 - `int visited[N][M][6]`
- 간선의 가중치가 항상 1이기 때문에 BFS만으로도 해결할 수 있다
- BFS로 각 상태에 도달하기 위한 최소 이동횟수를 기록하여 해결한다

E. 샤카샤카

의도한 난이도 : **CHALLENGING**

#구현

#출제 : 김민겸

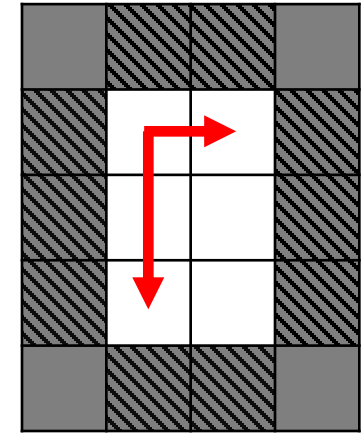
제출 40번, 정답 0명 (정답률 0.00%)

처음 풀 사람 : **N/A**

- 먼저 입력을 쪼개 각 블록의 모양을 알아낸다
- "삼각형 블록의 개수" 조건은 간단히 해결할 수 있다
- 흰색 영역이 직사각형인 경우는 두가지이다
 - 바른 직사각형
 - 기울어진 직사각형
- 각 칸을 1행부터 N열까지, 1열부터 M열까지 순서대로 탐색하자

- **바른 직사각형 판별**

- 바른 직사각형의 시작점은 반드시 흰색 정사각형 블록이다



- 탐색 중에 해당 블록이 나왔을 경우, 아래와 오른쪽 방향으로 탐색하여 직사각형의 크기를 알아낸다

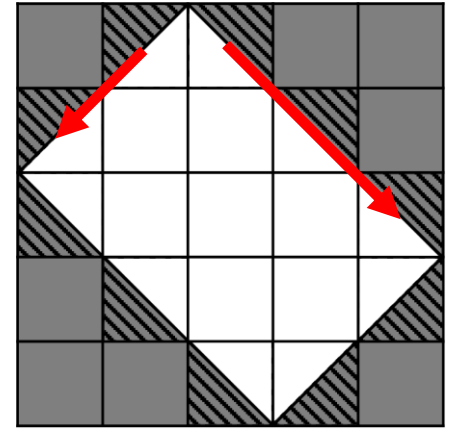
- 크기를 알아냈다면, 해당 직사각형의 외곽 부분이 잘 막혀 있는지, 직사각형의 내부는 흰색 정사각형 블록으로만 이루어져 있는지 확인한다

- 기울어진 직사각형 판별

- 기울어진 직사각형의 시작점은 반드시 왼쪽 위가 검은 삼각형 블록이다

- 탐색 중에 해당 블록이 나왔을 경우, 왼쪽 아래와 오른쪽 아래 방향으로 탐색하여 직사각형의 크기를 알아낸다

- 크기를 알아냈다면, 해당 직사각형의 외곽 부분에 엉뚱한 블록이 없는지, 직사각형의 내부는 흰색 정사각형 블록으로만 이루어져 있는지 확인한다



F. 정사각형 세기

의도한 난이도 : **HARD**

#수학

#출제 : 김민겸

제출 32번, 정답 4명 (정답률 12.50%)

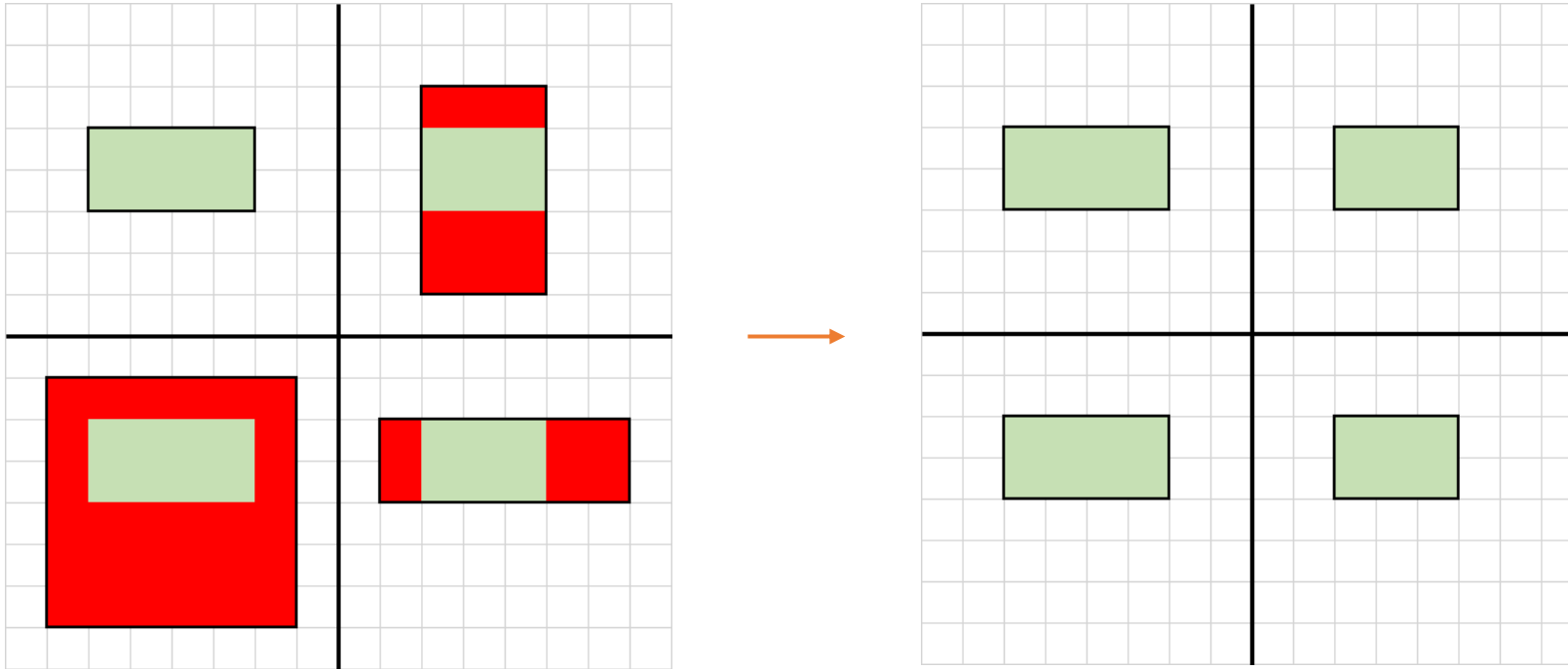
처음 푼 사람 : **황진익**, 60분

F. 정사각형 세기

- 각 직사각형에 대해 모든 점들을 대상으로 모든 사각형을 다 만들어 보는 것은 $O(N^8)$ 의 시간 복잡도를 가진다
 - N = 직사각형의 변의 길이
- 사각형의 대각선 방향의 두 꼭짓점만 결정되면 나머지는 자동으로 정해지기 때문에, 이 관찰을 통해 시간 복잡도를 $O(N^4)$ 로 낮출 수 있다
- 하지만, 이 문제를 풀기 위해선 더 좋은 방법이 필요하다

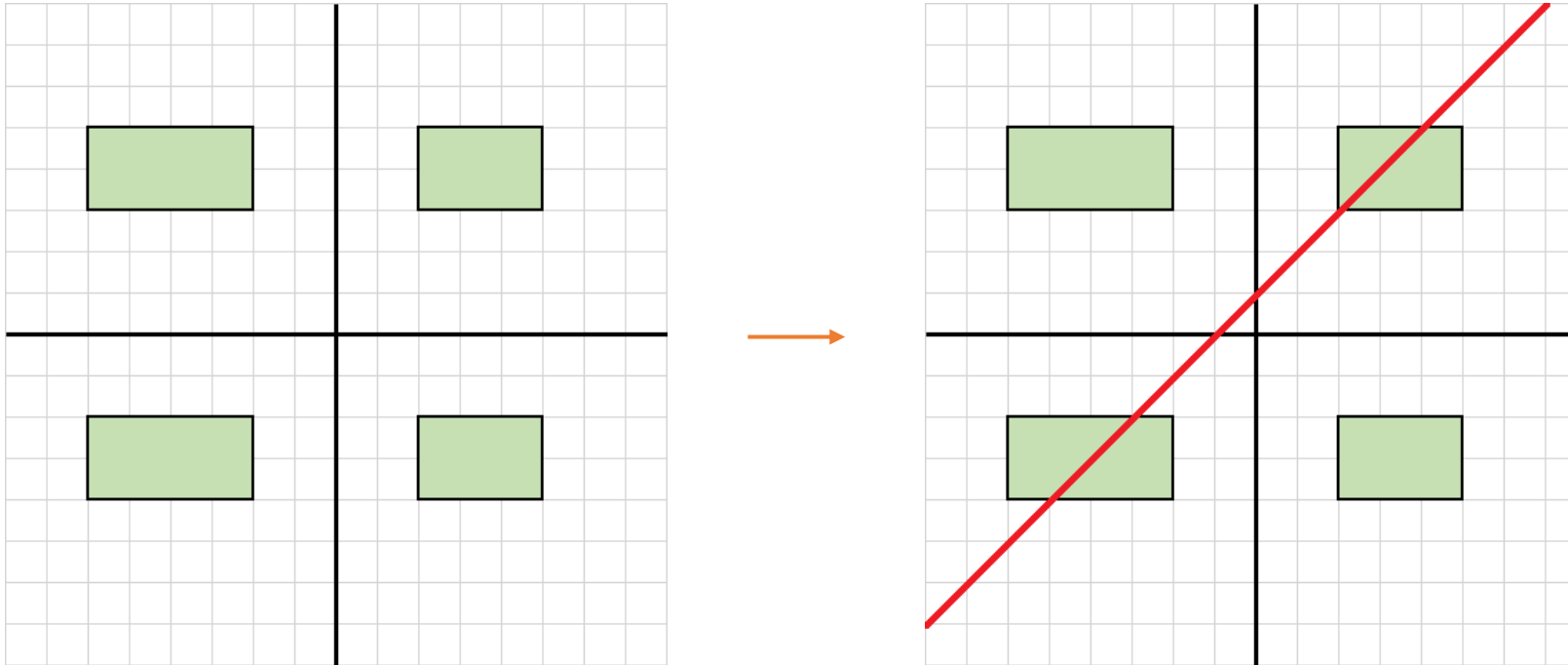
F. 정사각형 세기

- 먼저, 색칠된 부분은 사각형의 꼭짓점으로 사용될 수 없으므로 잘라낸다



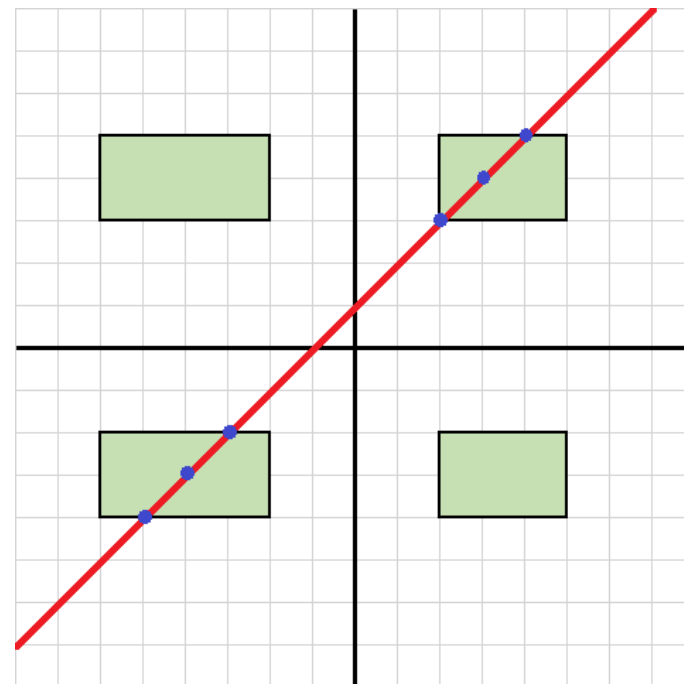
F. 정사각형 세기

- 여기에 $y = x + k$ (단, k 는 정수) 꼴의 대각선을 하나 그어보자



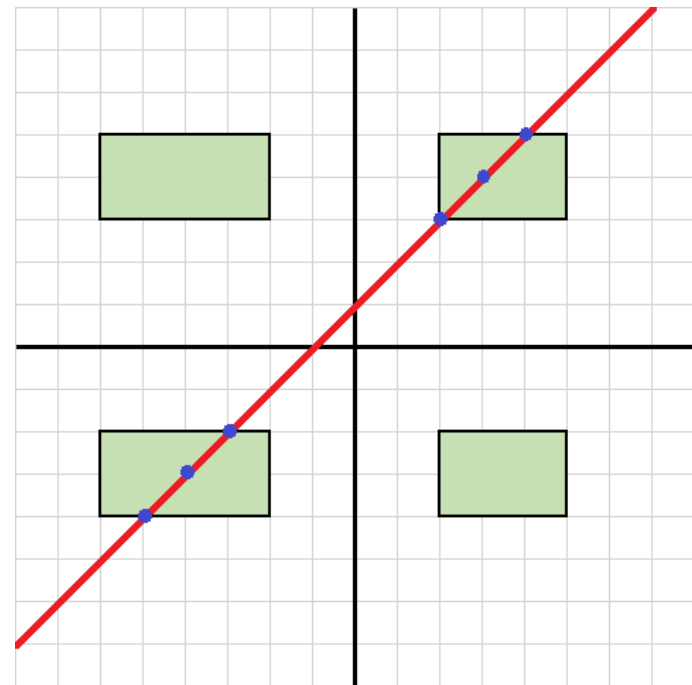
F. 정사각형 세기

- 해당 대각선이 각각의 두 사각형과 겹치는 격자점의 개수를 세자.
- 각 사각형에서 점을 하나씩 택하여 정사각형의 대각선 방향 꼭짓점으로 사용한다면 반드시 조건에 맞는 정사각형을 만들 수 있다.
- 이 때, 정사각형의 개수는 두 점의 개수의 곱이다.
- 따라서, -10^5 이상 10^5 이하의 모든 k 에 대해 $y = x + k$ 꼴의 그래프를 만들어 보면 모든 정사각형의 개수를 셀 수 있다.



F. 정사각형 세기

- 격자점의 개수를 셀 때,
- 사각형 내부의 점 개수를 일일이 세면?
 - 시간 복잡도는 대각선 하나당 $O(N)$, 최종적으로 $O(N^2)$ 이다.
- $y = x + k$ 식을 이용해 부등식 형태로 정리하면?
 - 대각선 하나당 $O(1)$, 최종적으로 $O(N)$ 이다.
- + $O(N)$ 보다 빠르게 해결할 수 있을까요?
가능하면 어떻게 풀어야 할지도 생각해 보세요.



G. 조각 케이크

의도한 난이도 : **NORMAL**

#완전탐색

#출제 : 정치훈

제출 64번, 정답 33명 (정답률 51.56%)

처음 풀 사람 : **권준혁**, 33분

- 2^N 가지의 모든 경우를 다 살펴보고, 이 중 케이크의 크기가 0.99 이상 1.01 이하인 경우가 몇 가지인지 세면 된다
- DFS를 이용한 백트래킹이나 비트마스킹을 이용하여 모든 경우를 완전탐색한다.
- 분수의 합은 유리수를 직접 구현할 수도 있지만, N 이 충분히 작기 때문에 C++의 실수 자료형 (double)로도 충분히 정답을 맞을 수 있다.
- +만약 N 이 훨씬 더 크다면 어떻게 문제를 풀 수 있을까요?

H. 크림 파스타

의도한 난이도 : **NORMAL**

#애드_혹

#DP

#출제 : 김민겸

제출 208번, 정답 35명 (정답률 17.31%)

처음 푼 사람 : **조은정**, 41분

- 각 답을 구할 때, 마지막으로 추가된 원소를 사용할 것인지 말 것인지 두 가지 경우가 존재한다.
 - 사용하는 경우, 해당 수가 추가되기 전 가장 작은 수와의 차가 답이다.
 - 사용하지 않는 경우, 직전에 구한 답이 답이다.
 - 두 경우 중 더 큰 값이 문제의 답이 된다.
- 답을 구하는 과정에서, 여태 추가된 수 중 가장 작은 수와 직전에 구한 답을 변수로 저장하고 사용하면 $O(N)$ 풀이를 만들 수 있다.

I. 타이핑

의도한 난이도 : **NORMAL**

#DP

#그리디_알고리즘

#출제 : 김민겸

제출 155번, 정답 32명 (정답률 20.65%)

처음 풀 사람 : **이준서**, 26분

- 동적 계획법을 이용한 풀이

- i 번째 글자까지 타이핑을 진행했을 때,
- 마름모 버튼의 활성화 여부가 j 인 최소 입력 횟수를 $dp[i][j]$ 라고 하자

- 점화식은 아래와 같다

- $dp[i][0] = \min(dp[i-1][0] + isupper(cur), dp[i-1][1] + 1) + 1$
- $dp[i][1] = \min(dp[i-1][1] + islower(cur), dp[i-1][0] + 1) + 1$

- 시간 복잡도 $O(N)$

- **그리디 알고리즘을 이용한 풀이**
- 현재 문자를 바로 입력할 수 있는 경우 입력한다.
- 현재 문자, 다음 문자의 대소문자 여부가 모두 마름모 버튼의 활성화 상태와 반대일 경우 마름모 버튼을 누르는 게 반드시 이득이다.
- 현재 문자만 반대인 경우 별 버튼을 누르는 게 반드시 이득이다.

J. 파밍 루트

의도한 난이도 : **HARD**

#DP

#이분탐색

#출제 : 김경태

제출 50번, 정답 1명 (정답률 2.00%)

처음 풀 사람 : **전현수**, 289분

- 플레이어 방어력이 d 로 고정되어 있다면?
- $c[u] := u$ 번 구역에 위치한 몬스터를 쓰러트렸을 때 얻을 수 있는 금화의 개수
- $dp[t][u] :=$ 남은 시간이 t 분이고 플레이어가 u 번 구역에 위치해 있을 때 얻을 수 있는 금화의 최대 개수
- $$dp[t][u] = \begin{cases} c[u] + \max_v(dp[t-1][v]), & \text{플레이어가 몬스터를 쓰러트리는 경우} \\ 0, & \text{플레이어가 몬스터에게 쓰러지는 경우} \end{cases}$$
- d 를 매우 큰 값으로 두고 플레이어가 얻을 수 있는 금화의 최대 개수 $c_{\max}$ 를 구한다.
- 이분탐색을 통해 금화 $c_{\max}$ 개를 얻을 수 있는 d 의 최솟값을 구한다.