



# 풀이

제 1회 생각하는 프로그래밍 대회  
2017년 9월 17일 일요일



## A. 현수막

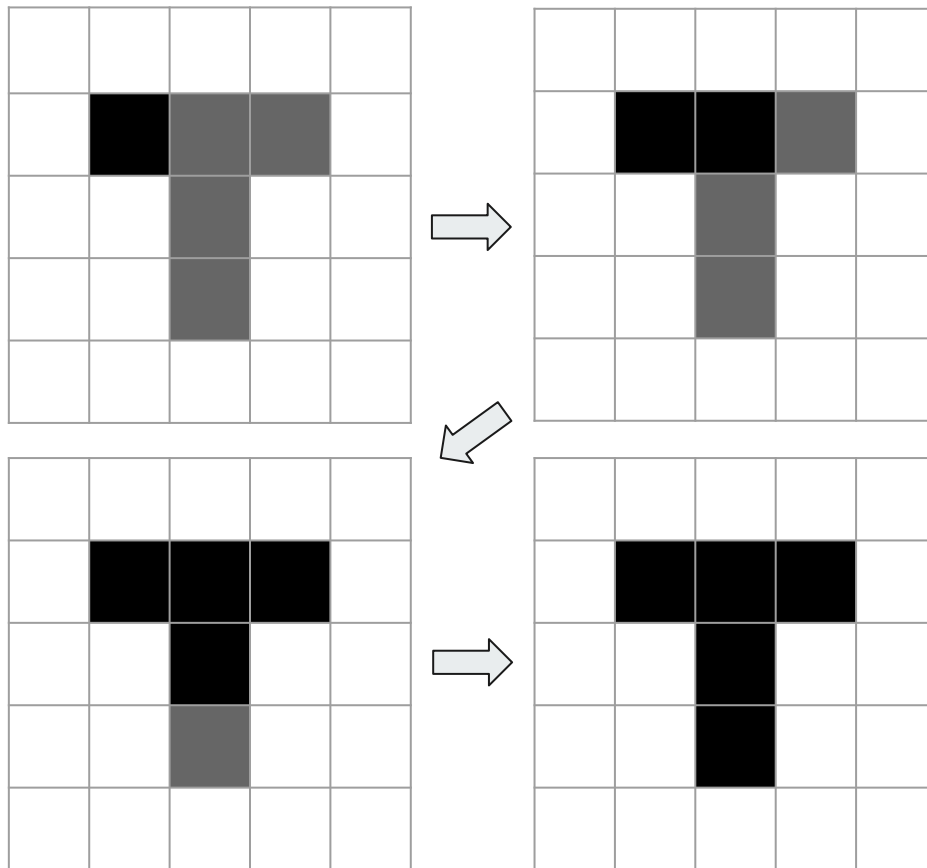
분류: Flood Fill, DFS, BFS

- 1, 0 으로 된 이미지에서 글자의 개수를 구하는 문제
- 상, 하, 좌, 우, 대각선으로 1이 인접해 있으면 하나의 글자로 생각함

Solution: **DFS or BFS**를 통한 그래프 탐색

## A. 현수막

0	0	0	0	0
0	1	1	1	0
0	0	1	0	0
0	0	1	0	0
0	0	0	0	0





## B. 았았다

분류: Mathematics

- 총 20장에서 영학이의 패 2장을 제외하면 18장
- 18장에서 2장을 뽑는 방법의 수 =  $18C2 = 18 \cdot 17 / 2 = 153$
- 상대방의 패는 이 153가지 중 하나
- 153가지의 모든 가능성을 시도해보고 영학이가 이기는 횟수를 센다
- 답: (영학이가 이기는 횟수) / 153



## C. 용감한 용사 진수

분류: Etc.

- 적 병사가 10명 있고, 각각 힘이 10, 20, 30, 40, 100, 200, 300, 400, 500, 600 이라고 하자
- 이 때, 진수가 힘에 80포인트를 투자하는 것은 의미가 있을까?
  - 40포인트만 힘에 쓰고 나머지 포인트는 아끼자!
- 10포인트를 투자하거나, 20포인트를 투자하거나, 30포인트를 투자하거나 .....
- 적 병사가 N명 있으면 N가지 수치만 시도해보면 됨
- 민첩도 마찬가지로



## C. 용감한 용사 진수

분류: Etc.

- 투자할 만한 힘 수치는  $N$ 가지
- 투자할 만한 민첩 수치는  $N$ 가지
- $N*N$ 가지 경우에 대해 모두 시도해보자!
- 힘을  $x$ 로 고정, 민첩을  $y$ 로 고정
  - 이제 지능이 얼마나 높아야  $K$ 명의 적을 이길 수 있을까?
  - 적을 지능 순으로 정렬해두고  $K$ 명을 이기기 위해 필요한 최소의 진수의 지능을 찾는다
- $O(N^3)$



## D. 빗물

- 예를 들어 설명
- 2차원 세계의 가로 길이  $W = 500$  이라고 하자
- 가장 높은 블록이 위치한  $x$ 좌표가 100, 200, 300, 400 이라고 하자
- [101, 199] 구간, [201, 299] 구간, [301, 399] 구간에 물을 짹 채운다
  - [1, 99] 구간과 [401, 500] 구간이 남았다
    - 두 구간을 푸는 방법은 같다
    - [1, 99] 구간을 푸는 방법만 설명
- [1, 99] 구간에 가장 높은 블록이 위치한  $x$ 좌표가 50, 60, 70 이라고 하자
- [51, 59], [61, 69], [71, 99] 구간에 물을 짹 채운다
  - [1, 49] 구간이 남았다
- [1, 49] 구간에 가장 높은 블록이 위치한  $x$ 좌표가 30, 40 이라고 하자
- [31, 39], [41, 49] 구간에 물을 짹 채운다
  - [1, 29] 구간이 남았다
- 반복



## D. 빗물

- 다양한  $O(W)$  시간 풀이가 있다
  - 스택을 이용한 풀이
  - $[1, 99]$  구간을 1부터 99까지 스캔하면서 최대값을 갱신하며 답을 구하는 풀이
  - $[1, x-1]$  구간의 최대값,  $[x+1, 500]$  구간의 최대값을 이용한 풀이
- 다른 시간복잡도를 가진 풀이
  - 2차원 세계의 위에서부터 아래로 내려오면서 각 층별로 물이 얼마나 차는지 계산





## E. 우유 축제

분류: Greedy

- 우유가게가 일렬로 늘어서 있을 때 지나가면서 우유를 먹는 문제
- 딸기 -> 초코-> 바나나
- 일렬로 지나갈 때 해당 순서대로 우유를 마실 수 있으면 무조건 선택

Solution: 우유가게를 지나는 순간 자신이 만족할 수 있는 가장 **Best**를 선택하는 **Greedy** 풀이



## F. 성적표

분류: Brute Force, Simulation, Linear Regression, Mathematics

- 가능한  $a$ 의 값 100개  $[1, 100]$
- 가능한  $b$ 의 값 100개  $[1, 100]$
- 10000가지의 경우에 대해,  $(a, b)$ 를 고정하고 RSS를 계산
  - $N$ 개의 데이터를 훑으면서 계산하기
    - 이 방법으로도 충분히 시간 안에 통과
  - 적절한 전처리를 통해  $O(1)$ 에 계산하기!



## G. 우유 도시

분류: Dynamic Programming

- $DP[i][j][t]$  := 영학이는  $t$ 번 맛을 먹을 차례이고,  $(i, j)$ 에 위치해있다.  $(i, j)$ 에서 파는 우유를 아직 먹지는 않은 상태이다. 이 때 영학이가 마실 수 있는 우유의 최대 개수.
- 답:  $DP[0][0][0]$
- $DP[i][j][t] = \max(\text{남쪽으로 가기}, \text{동쪽으로 가기})$
- 현재 위치에서 파는 우유를 먹을 수 있으면 항상 먹는 게 최적
- (현재 위치에서 파는 우유를 먹음)  $DP[i][j][t] = \max(DP[i+1][j][t+1], DP[i][j+1][t+1])$
- (현재 위치에서 파는 우유를 못 먹음)  $DP[i][j][t] = \max(DP[i+1][j][t], DP[i][j+1][t])$



## H. 이산수학 과제

분류: Simulation, Mathematics

- $g = \left\lfloor \frac{\sqrt{8N-7}-1}{2} + 1 \right\rfloor$
- $m = \frac{g(g+1)}{2}$

N번째 유리수가 a/b 일 때,

- $a = m - N + 1$
- $b = N - m + g$

Solution:

- 반복문 사용하여 시뮬레이션 (군수열)
- 식을 유도하여 풀이  $O(1)$



# I. 관리자는 누구?

분류: Maximum Value

- 각 동아리에서 동아리원 N명 중 가장 문제 풀 개수가 많은 사람이 후보 (최대값)
- 각 동아리에서 뽑은 후보 중 가장 문제 풀 개수가 많은 사람이 최종적으로 관리자 (최대값)

Solution: 전체 동아리원 중에서 **최대값**을 찾고 해당하는 동아리명을 출력



## J. 개미굴

분류: Graph, Tree

- 개미굴의 구조를 **Tree** 형태로 출력하는 문제
  - 로봇개미가 주어주는 정보는 **Root**부터 모든 **Leaf**까지 가는 개수
  - 주어지는 입력으로 순서대로 **Tree**를 구성하면 된다.
- 
- 사전순서 출력? **Ordered Map** 자료구조 (TreeMap(JAVA), Map(C++))  
Or 출력하기 전 직접 **정렬**을 수행하여도 무방하다.

Solution: 디렉토리 트리 만들기