

# UCPC 2018

온라인 예선

간략한 풀이

스코어보드 : <https://ucpc2018p.acmicpc.net/>

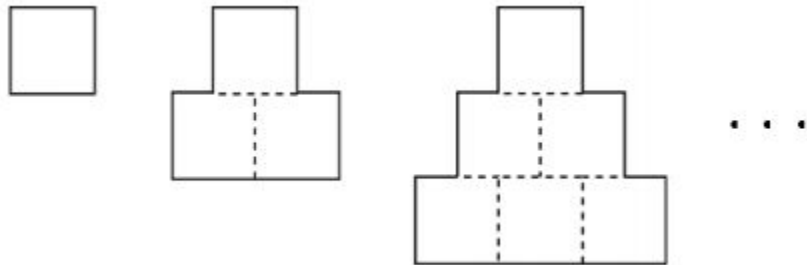
# A. 수학은 체육과목 입니다

정답 팀 : 184

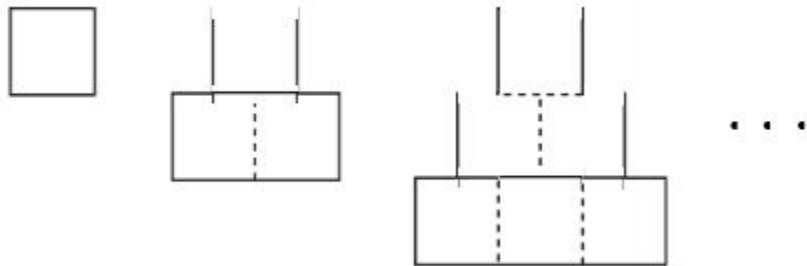
가장 처음 푼 팀 : TCG (t\*\*\*\*\*, c\*\*\*\*\*, g\*\*\*), 0분

출제자 : ntopia

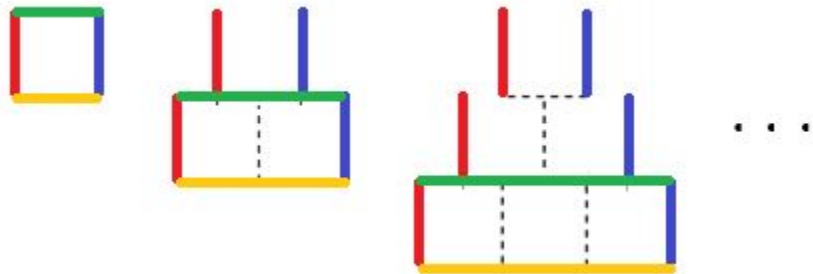
# A. 수학은 체육과목입니다



# A. 수학은 체육과목입니다



# A. 수학은 체육과목입니다



$$4 \times n$$

## G. UCPC는 무엇의 약자일까?

정답 팀 : 167

가장 처음 푼 팀 : 남남서와 씨끄레기들 (신승원, 김동현, 김현수), 1분

출제자 : ntopia

## G. UCPC는 무엇의 약자일까?

공백이 들어간 문자열을 입력받을 수 있는가?

‘U’가 등장하고 그 뒤에 ‘C’가 등장하고

그 뒤에 ‘P’가 등장하고 그 뒤에 ‘C’가 등장하는지 확인하면 끝!

## G. UCPC는 무엇의 약자일까?

```
1 import re
2 ans = re.search('U.*C.*P.*C',input())
3 print('I love UCPC' if ans else 'I hate UCPC')
```



## D. 나무 탈출

정답 팀 : 121

가장 처음 푼 팀 : SC2GodGame (박상수, 김진표, 박성진), 8분

출제자 : ntopia

## D. 나무 탈출

게임말을 하나 골라 부모로 올려봅시다. 무엇이 변할까요?

그 게임말과 루트 노드 사이의 거리가 1 줄어들죠!

그렇다면?

## D. 나무 탈출

$S := \text{sum}(\text{dist}(\text{root}, i\text{번\_게임말의\_위치}))$

플레이어의 행동 한 번은 **S**의 홀짝을 항상 뒤집는다!

따라서 **S**의 초기값이 홀수면 먼저 두는 사람이 필승!

## H. 잘못 구현한 에라토스테네스의 체

정답 팀 : 88

가장 처음 푼 팀 : 남남서와 찌끄레기들 (신승원, 김동현, 김현수), 15분

출제자 : ntopia

## H. 잘못 구현한 에라토스테네스의 체

답은

$$\sum_{k=1}^n \left\lceil \frac{n}{k} \right\rceil$$

그런데 시간내에 계산하려면 어떻게 해야할까?

## H. 잘못 구현한 에라토스테네스의 체

$$\sum_{k=1}^n \left\lceil \frac{n}{k} \right\rceil$$

k가 꽤 커진 상황을 생각해보자.

ceil(n/k) 값은 얼마나 자주 변할까?

ex)  $\text{ceil}(10000/1001) = \text{ceil}(10000/1002) = \dots = \text{ceil}(10000/1111) = 10$

## H. 잘못 구현한 에라토스테네스의 체

$$\sum_{k=1}^n \left\lceil \frac{n}{k} \right\rceil$$

$1 \leq k \leq T$  (적당한 상수) 구간에서는 그냥  $\text{ceil}(n/k)$  값을 직접 구하고,

$T \leq k \leq n$  구간에서는  $\text{ceil}(n/k)$  값이 같아지는  $k$ 의 범위를 구하기

$T = \text{sqrt}(n)$  으로 두면 최적!

## B. 카드 합체 놀이

정답 팀 : 169

가장 처음 푼 팀 : 더불어민규당 (구재현, 조민규, 이종원), 2분

출제자 : ldtl



## B. 카드 합체 놀이

우리의 문제) 카드 2개를 없애고, 2개의 합이 2개 추가된다.

어디서 본 문제) 카드 2개를 없애고, 2개의 합이 1개 추가된다.

## B. 카드 합체 놀이

어디서 본 문제: 허프만 알고리즘!!

= 항상 가장 작은 2개를 합쳐나가면 된다.

이 문제도 가장 작은 2개를 계속 골라나가면 된다.

찍어맞추기는 좋지만 엄밀히 증명하기는 어려운 문제

## B. 카드 합체 놀이

출제자님의 증명과 zlzmsrhak님의 증명 의역

[https://docs.google.com/document/d/1S\\_y3T7whkEcX7XYHExVjyanVuvyWv93W\\_Sd89ALQwl4/edit?usp=sharing](https://docs.google.com/document/d/1S_y3T7whkEcX7XYHExVjyanVuvyWv93W_Sd89ALQwl4/edit?usp=sharing)

## F. 트리와 색깔

정답 팀 : 48

가장 처음 푼 팀 : 남남서와 찌끄레기들 (신승원, 김동현, 김현수), 7분

출제자 : appa

## F. 트리와 색깔

아주 여러가지 해법이 있을 수 있음.

## F. 트리와 색깔

### 1. 오프라인

트리의 정점을 DFS order 순으로 늘어놓고 그 순서로 세그먼트 트리를 만듦

색깔이 작은 순으로 보면서 색깔이  $k$ 인 정점과 쿼리들을 한번에 처리함

-> 색깔이  $k$ 인 정점을 세그트리에 밀어넣고

-> 색깔이  $k$ 인 쿼리를 처리하면 OK!

## F. 트리와 색깔

### 2. 온라인 1

트리의 정점을 DFS order 순으로 늘어놓고 그 순서로 머지소트 트리를 만듦

머지소트 트리의 각 노드엔 트리의 정점이 색깔 순으로 정렬되어 있음

쿼리가 주어지면 그 부트리에 해당하는 머지소트 트리 구간에 쿼리

$O(N \log N + M \log^2 M)$

## F. 트리 와 색깔

### 3. 온라인 2

use Persistent Segment Tree

$O(N \log N + M \log M)$



# I. 피아의 아틀리에 ~신비한 대회 of 연금술사~

정답 팀 : 36

가장 처음 푼 팀 : Make Appa Soldier Again! (appa, hongjun-7, h0ngjun-7), 45분

출제자 : zlzmsrhak

# I. 피아의 아틀리에 ~신비한 대회 of 연금술사~

모든 경우를 다 해볼 수 있을까?

# I. 피아의 아틀리에 ~신비한 대회 of 연금술사~

재료를 선택하는 경우의 수 :  $10 \times 9 \times 8$

각 재료마다 격자에 놓을 수 있는 곳 : 4

각 재료마다 회전하는 경우의 수 : 4

전체 경우의 수 :  $10 \times 9 \times 8 \times (4 \times 4)^3 = 2949120$

하나의 경우마다 필요한 연산의 수 : 대충  $16 \times 3 + 25 = 73$

계산해보면 시간제한 안에 노려볼만 하다!

# I. 피아의 아틀리에 ~신비한 대회~의 연금술사~

하지만 코딩은 매우 어렵다!!

정해는 9중포문으로 되어있다.

```
for(int i = 1; i <= n; i++){
    for(int j = 1; j <= n; j++){
        for(int k = 1; k <= n; k++){
            if(i == j || i == k || k == j) continue;
            for(int x = 0; x < 4; x++){
                for(int y = 0; y < 4; y++){
                    for(int z = 0; z < 4; z++){
                        for(int p = 0; p < 4; p++){
                            for(int q = 0; q < 4; q++){
                                for(int r = 0; r < 4; r++){
                                    init();
                                    put_elm(D[i], x/2, x%2);
                                    put_elm(D[j], y/2, y%2);
                                    put_elm(D[k], z/2, z%2);
                                    if(ans < score()){
                                        trace = {i, j, k, x, y, z, p, q, r};
                                        ans = score();
                                    }
                                    D[k].r90();
                                }
                                D[j].r90();
                            }
                        }
                    }
                }
            }
            D[i].r90();
        }
    }
}
```

# I. 피아의 아틀리에 ~신비한 대회 of 연금술사~

(주의) 상위권 팀에 낚이지 말자.

상위권 팀은 보통 코딩만 어려운 문제를 늦게 푼다.

코딩만 어려운 문제는 컴퓨터가 놓면 시도해보자.

ICPC는 한 문제 차이가 매우 크다!!



**zizmsrhak** 2:45 PM

근데 소각로보다 코딩 한참 쉬운데

왜 소각로 풀지



**ainta**  2:45 PM

ㅇ지합니다

!푸는데는

10분이면

충분

# J. &+ +&

정답 팀 : 19

가장 처음 푼 팀 : TCG (t\*\*\*\*\*, c\*\*\*\*\*, g\*\*\*), 32분

출제자 : moonrabbit2

# J. &+ +&

1번 문제

$(A_i \& B_j)$  의  $k$ 번 비트가 1이려면?

->  $A_i$  와  $B_j$  모두  $k$ 번 비트가 1이어야 함

각 비트  $k$ 에 대해,  $k$ 번 비트가 켜져있는 수의 개수를 수열  $A, B$ 에 대해 저장하면

전체 합에  $k$ 번 비트가 기여하는 양은  $\text{cntA}[k] * \text{cntB}[k]$

# J. &+ +&

2번 문제

최종 답의 k번 비트가 1이려면

->  $C_i = A_i \bmod 2^{k+1}$ ,  $D_j = B_j \bmod 2^{k+1}$  이라 할 때, 모든  $i, j$ 에 대해

$$2^k \leq C_i + D_j < 2 \cdot 2^k \text{ 이거나, } 3 \cdot 2^k \leq C_i + D_j < 4 \cdot 2^k$$

를 만족해야 함.



# J. &+ +&

## 2번 문제

모든  $C_i$ 에 대해 이를 만족하는  $D_j$ 의 개수를 이분탐색을 이용해 알 수 있음.

모든  $C_i$ 에 대해 이를 만족하는  $D_j$ 의 개수가  $n$ 개여야 OK!

$C_i, D_j$  를 계산하는건 아래 비트부터 위로 올라가면서 기수 정렬을 같이 해주면 됨.

시간복잡도  $O(30*n*\log n)$

하지만 출제자님은 이 솔루션은 TLE 나게 하고 싶었다고 합니다.

# J. &+ +&

2번 문제

앞의 관찰을 좀 더 들여다보면

$C_i$ 에 대해 조건을 만족하는  $D_j$ 의 범위를

$i$ 가 증가할 때마다 sliding window로 관리할 수 있음을 알 수 있다.

총 시간복잡도  $O(30*n)$

이 외에도 수많은  $O(30*n)$  풀이가 존재합니다.

## C. Split and Merge

정답 팀 : 17

가장 처음 푼 팀 : 남남서와 씨끄레기들 (신승원, 김동현, 김현수), 26분

출제자 : ainta

## C. Split and Merge

격자의 길이가  $L$ 일 때, 격자 사이의  $L-1$ 개의 세로선에 대해서

나뉜 것을 1, 아닌 것을 0으로 보면 상태를 길이  $L-1$ 의 **bitstring**으로 생각할 수 있다.

모든 조각의 길이가 1또는 2이므로 연속한 0은 존재하지 않아야 한다.

## C. Split and Merge

먼저 필요한 연산의 최소 횟수를 구해보자.

초기 상태에서 0인데 나중 상태에서 1인 부분을 모두 쪼갬 후,

초기 상태에서 1인데 나중 상태에서 0인 부분을 모두 합치면 조건을 만족한다.

따라서, 두 상태에서 값이 다른 위치의 개수가 답이 된다.

## C. Split and Merge

두 상태에서 값이 다른 위치의 개수가 답이라는 것은

값이 같은 위치는 변하지 않아야 한다는 뜻이다.



## C. Split and Merge

따라서, 둘 모두 1인 부분으로 나누면 각각의 조각들은 독립적으로 계산 가능하다.



## C. Split and Merge

따라서, 둘 모두 1인 부분으로 나누면 각각의 조각들은 독립적으로 계산 가능하다.

각 조각들은 한쪽이 01010...이고 다른 한쪽은 10101....이라는 성질을 가진다.





## C. Split and Merge

따라서, 둘 모두 1인 부분으로 나누면 각각의 조각들은 독립적으로 계산 가능하다.

각 조각들은 한쪽이 01010...이고 다른 한쪽은 10101....이라는 성질을 가진다.

길이 N의 bitstring 01010...을 10101...로 중간에 0이 연속하지 않도록 N번만에 바꾸는 방법의 경우의 수를  $D[N]$ 이라고 하면

$$D[0] = D[1] = 1$$

$$D[i] = D[0]*D[N-1]*\binom{N-1}{0} + D[2]*D[N-3]*\binom{N-1}{2} + \dots$$

(0 하나를 1로 바꾼 후 같은 성질의 2조각으로 나뉘기 때문)

## C. Split and Merge

$$D[0] = D[1] = 1$$

$$D[i] = D[i] = D[0]*D[N-1]*N-1C0 + D[2]*D[N-3]*N-1C2 + \dots$$

의 점화식으로 DP 배열을 채운 후, 각 조각의 **bitstring** 길이가  $L_1, \dots, L_k$ 이면

$D[L_1] * \dots * D[L_k] * (L_1 + \dots + L_k)! / (L_1! L_2! \dots L_k!)$ 이 최종 경우의 수가 된다.

## E. 소각로

정답 팀 : 13

가장 처음 푼 팀 : 더불어민규당 (구재현, 조민규, 이종원), 39분

출제자 : moonrabbit2

## E. 소각로

1. 소각로 상의  $[L, R]$  구간에 소각 작업을 진행한다.  
-> 소각하는 양 만큼 대기열에서 소각로로 이동한다.
2.  $i$ 번째 소각로 칸에 놓여있는 쓰레기의 종류를 출력한다.
3.  $p$ 번 종류의 쓰레기  $q$ 개를 현재 대기열 뒤에 넣는다.
4. 재활용을 위해 대기열 맨 앞의 쓰레기  $t$ 개를 제거한다.

## E. 소각로

아무 생각없이 하면 시간이 매우 오래걸린다!!

“3. p번 종류의 쓰레기 q개를 현재 대기열 뒤에 넣는다.”를 빠르게 해보자!

-> 연속한 같은 종류의 쓰레기를 (종류, 개수)로 관리하자!!

-> 소각로도 (종류, 개수)로 관리하자!!!

예시) 소각로의 상태가 1 1 2 2 0 0 1 1 1 2 2라면,

-> (1, 2) (2, 2) (0, 2) (1, 3) (2, 2)

## E. 소각로

3.  $p$ 번 종류의 쓰레기  $q$ 개를 현재 대기열 뒤에 넣는다.
4. 재활용을 위해 대기열 맨 앞의 쓰레기  $t$ 개를 제거한다.

(종류, 개수) 정보를 Deque로 관리하면  $O(Q)$ !

## E. 소각로

1. 소각로 상의  $[L, R]$  구간에 소각 작업을 진행한다.
2.  $i$ 번째 소각로 칸에 놓여있는 쓰레기의 종류를 출력한다.

1번 쿼리:  $[L, R]$  구간에 완벽히 포함되는 (종류, 개수)는 삭제, 걸치는 경우는 최대 2개.

+소각로에 추가되는 구간은 Deque에 추가되는 원소 수를 넘을 수 없다!

2번 쿼리: 이진탐색으로  $O(\lg N)$

->  $O((Q+N) \lg N)$

## E. 소각로

이진트리 종류를 쓰거나, **set**을 쓰는 식으로 구현이 가능하다.

단, **set**은 (종류, 개수) 보다는, (첫 번째 쓰레기 위치, 개수)가 구현이 더 편하다.

구현이 매우 복잡하지만, 전형적이기 때문에 익숙한 만큼 쉬워지는 문제.



끝

감사합니다!