

UCPC 2018

본선

간략한 풀이

스코어보드 : <https://ucpc2018.acmicpc.net/>

J. 아기 석환 뚜루루 뚜루

정답 팀 : 50 (49 + 1)

First Solve : TCG @ t*****, c*****, g*** (2 min.)

출제자 : norang

J. 아기 석환 뚜루루 뚜루

14단어가 계속 반복된다

→ `word[(n-1) % 14]` 를 출력!

출력하려는 단어가 `tururu..ru` 라면

현재 몇 절인지를 구해서

적절히 출력

14 (baby sukhwan tururu turu
very cute tururu turu
in bed tururu turu
baby sukhwan

14 (baby sukhwan turururu tururu
very cute turururu tururu
in bed turururu tururu
baby sukhwan

14 (baby sukhwan tururururu turururu
very cute tururururu turururu
in bed tururururu turururu
baby sukhwan

...

E. Thinking Heap

정답 팀 : 47 (45 + 2)

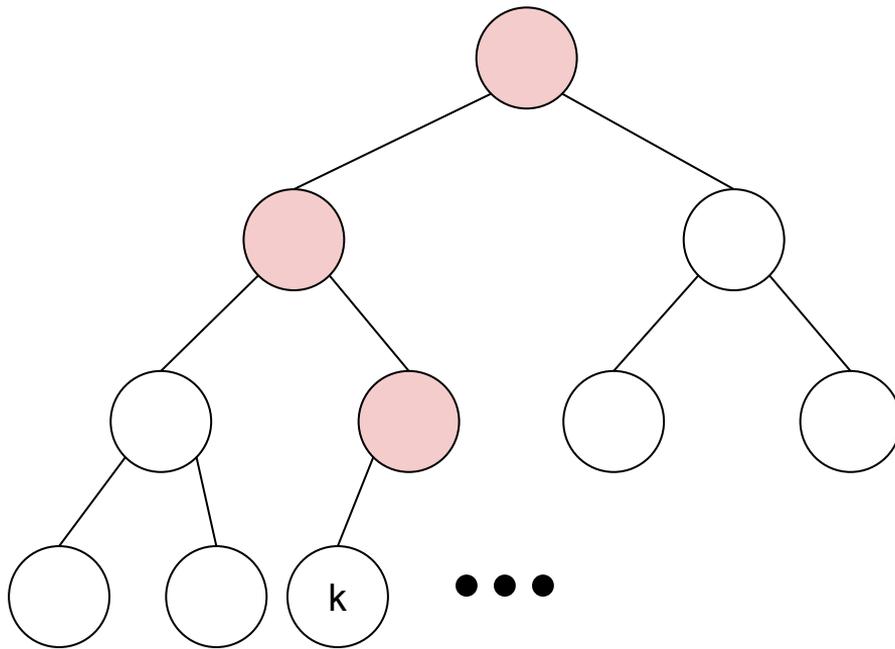
First Solve : Make Appa Soldier Again! @ appa, hongjun-7, h0ngjun-7 (18 min.)

출제자 : ntopia

E. Thinking Heap

빨간 노드 (k 의 모든 조상) 에 오는 수는

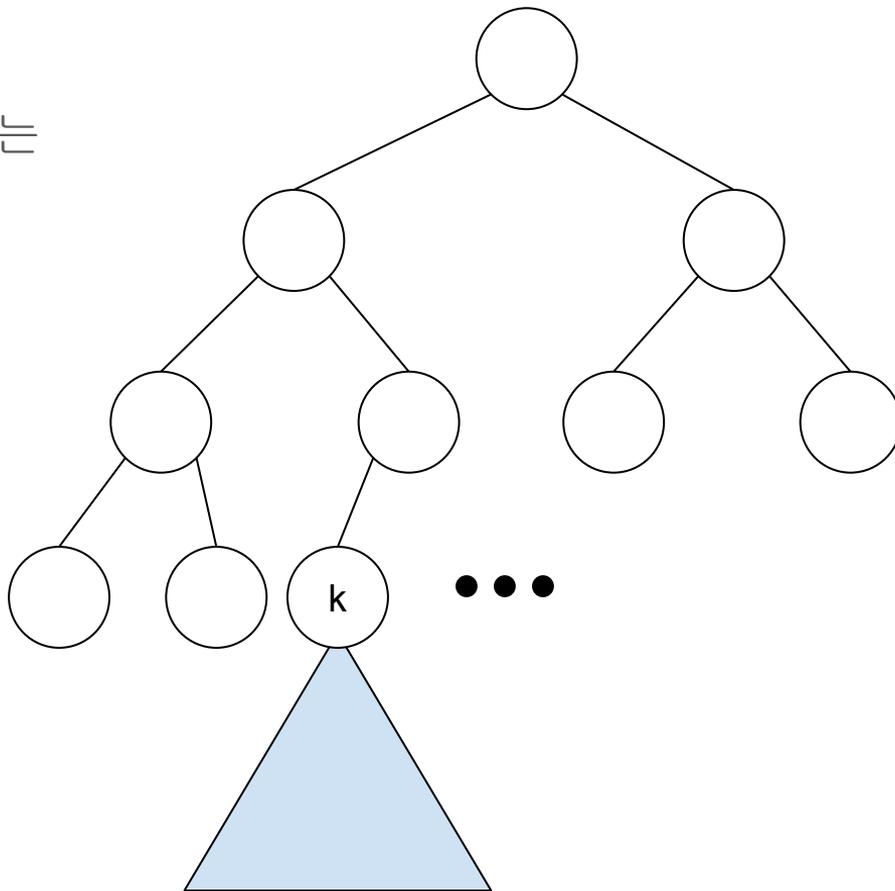
무조건 k 보다 작아야함!



E. Thinking Heap

파란 부분 (k의 모든 자식들) 에 오는 수는

무조건 k보다 커야함!



E. Thinking Heap

앞의 두 조건을 만족하기만 하면 무조건 만들 수 있다!

두 조건을 만족하도록 적당히 삽입하면 OK

A. 더위 피하기

정답 팀 : 48 (45 + 3)

First Solve : 박트리와 형님들 @ baactree, smu, usefulhyun (13 min.)

출제자 : gs12117

A. 더위 피하기

T제한이 200입니다. 금방 더위먹어 쓰러져요...

아무리 걸어도 출발지에서 200m 이상 갈 수 없습니다.

중요한 건 출발지 주변 200칸!

A. 더위 피하기

출발지 주변 $401*401$ 칸에 대해서

각 시점마다 이 위치에 있을 수 있는 경우의 수를 동적계획법으로 계산한다

시간복잡도 : $O(T^3+N)$

H. Make Similar

정답 팀 : 30 (19 + 11)

First Solve : 남남서와 찌끄레기들 @ 신승원, 김동현, 김현수 (71 min.)

출제자 : ainta

H. Make Similar

4가지 케이스로 나눌 수 있습니다.

- 1) 양의 정수와 음의 정수가 모두 존재
- 2) 양의 정수가 있고, 음의 정수가 없는 경우
- 3) 음의 정수가 있고, 양의 정수가 없는 경우
- 4) 모두 0

4번째 케이스는 자명하게 0임을 알 수 있습니다.

H. Make Similar

1. 음수와 양수가 모두 존재하는 경우

유클리드 호제법처럼 하여 $g = \gcd(a_1, \dots, a_n)$ 을 만들 수 있습니다. 이를 이용해 $-g$ 역시 만들 수 있습니다. g 와 $-g$ 를 다른 수들에 더하면 $0, 0, 0, \dots, 0, g$ 로 만들 수 있는데, 이제 0 들에 g 를 더하면 모두 같아지므로 답은 0 이 됩니다.

H. Make Similar

2. 양수가 존재하고, 음수가 존재하지 않는 경우

양의 정수 중 최솟값이 a , 최댓값이 b 라고 할 때 답은 $\min(a, b-a)$ 가 됩니다.

(Ex. $\{0,2,5\}$ 인 경우 $\min(2,5-2) = 2$ 가 답)

H. Make Similar

2. 양수가 존재하고, 음수가 존재하지 않는 경우

초기에는 최대-최소가 $b-a$ 입니다.

답이 a 가 되도록 하는 것은 a 이외에 나머지 수들의 차를 $a-1$ 이하로 만들 수 있으므로 가능합니다.

또한 $p+=q$ 를 한 후에 p 와 q 의 차이는 원래 q 의 값이 되고, 모든 수는 초기보다 감소하지 않으므로 1번이상 연산한 후 답이 a 보다 작아지게 만들 수는 없습니다.

H. Make Similar

3. 음수가 존재하고, 양수가 존재하지 않는 경우

2번 경우와 마찬가지로 문제를 해결할 수 있습니다.

시간복잡도 : $O(N)$

L. Piet

정답 팀 : 31 (15 + 16)

First Solve : 그 팀 @ veydpz, portableangel, Jhuni (56 min.)

출제자 : bryanj

L. Piet

블록의 개수의 상한은 NM 개(실제로 블록이 NM 개인 테스트 케이스는 없지만..),
DP와 CC의 조합은 8개이므로 가능한 상태의 수는 최대 $8NM$ 개입니다.

프로그램이 끝나지 않은 입력은 없다고 했으므로 다음 블록을 찾는 과정을 최대 $8NM$ 번 반복하면 됩니다.

다음 블록을 찾는 과정에 **flood-fill**이 들어가는데, 이 과정은 블록의 크기에 비례하고 모든 블록의 크기의 합이 NM 이므로 모든 블록의 다음 블록을 찾는 과정이 **amortized $O(NM)$** 입니다.

즉, 그냥 스펙대로 잘 구현하면 됩니다 :)

G. 성공

정답 팀 : 24 (15 + 9)

First Solve : TCG @ t*****, c*****, g*** (63 min.)

출제자 : jh05013

G. 성공

$D=1$ 이라면 전형적인 0/1 BFS 문제. 인접한 칸을 양방향으로 연결하되, 장애물이 있는 칸이 연결되면 비용은 1, 아니면 0으로 두면 해결.

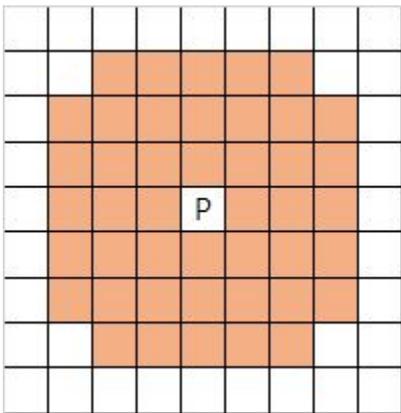
문제는 $D>1$ 일 때이다. 0/1 BFS를 그냥 짜면 $O(NMD^2)$ 로 TLE.

$D=1$ 의 아이디어를 일반적인 D 에도 적용시킬 수 있을까?

G. 성공

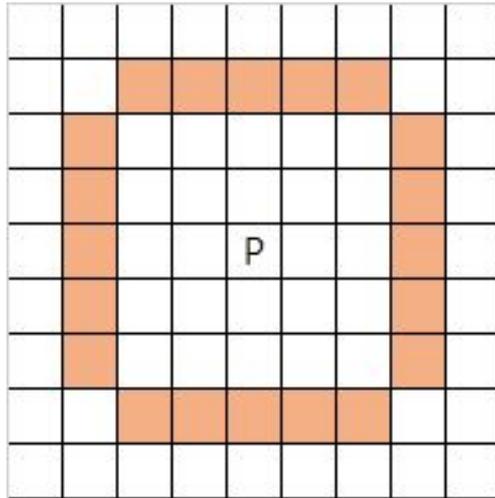
폭파의 조건을 살짝 바꿔서, 현재 위치한 칸에 인접한 $D \times D$ 영역만 폭파할 수 있다고 하자. 그래도 결과가 바뀌지 않음은 자명.

$D=3$ 의 예를 들어 보면 현재 위치 P 를 기준으로 아래의 범위를 폭파 가능.



G. 성공

또한 이 영역 안에서 마지막으로 밟는 칸은 영역의 가장자리이거나 도착 지점이고 (물론 도착지점이 이 영역 안에 있을 때에만 해당), 그 칸에는 한 번의 폭파만으로도 도달할 수 있다. 즉 실제로 비용 1의 간선을 그어 줄 필요가 있는 칸은 다음 $O(D)$ 개 및 도착 지점 밖에 없다.



G. 성공

실제로 간선을 만들 필요는 없고, 약 $8D$ 개의 지점을 직접 순회해줘도 된다.

$O(NMD)$

각각의 행/열을 자료구조로 잘 관리하면 $O(NM \lg(NM))$ 도 가능.

C. 네트워크 해킹

정답 팀 : 13 (9 + 4)

First Solve : 더불어민규당 @ 구재현, 조민규, 이종원 (58 min.)

출제자 : ntopia

C. 네트워크 해킹

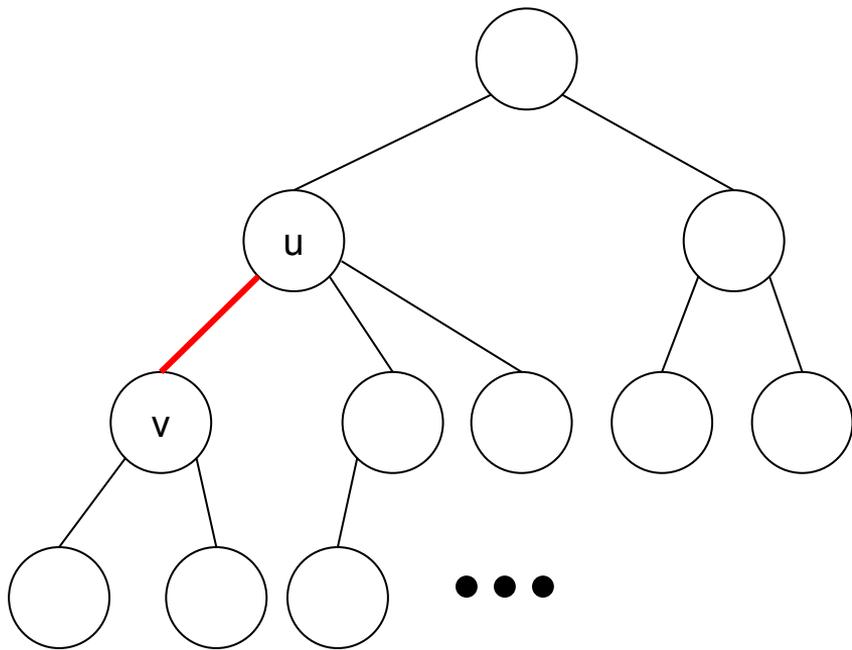
트리의 간선 중 하나를 끊고 그 간선을 다른 곳에 달았을 때,
어떤 선택을 하면 트리의 지름을 제일 크게 만들 수 있는가?

귀찮으니 아무 노드나 루트로 잡고 루트 있는 트리로 생각하자.

C. 네트워크 해킹

u-v 사이의 간선을 끊는다면,

이걸 어디에 연결하는 것이 최적일까?



C. 네트워크 해킹

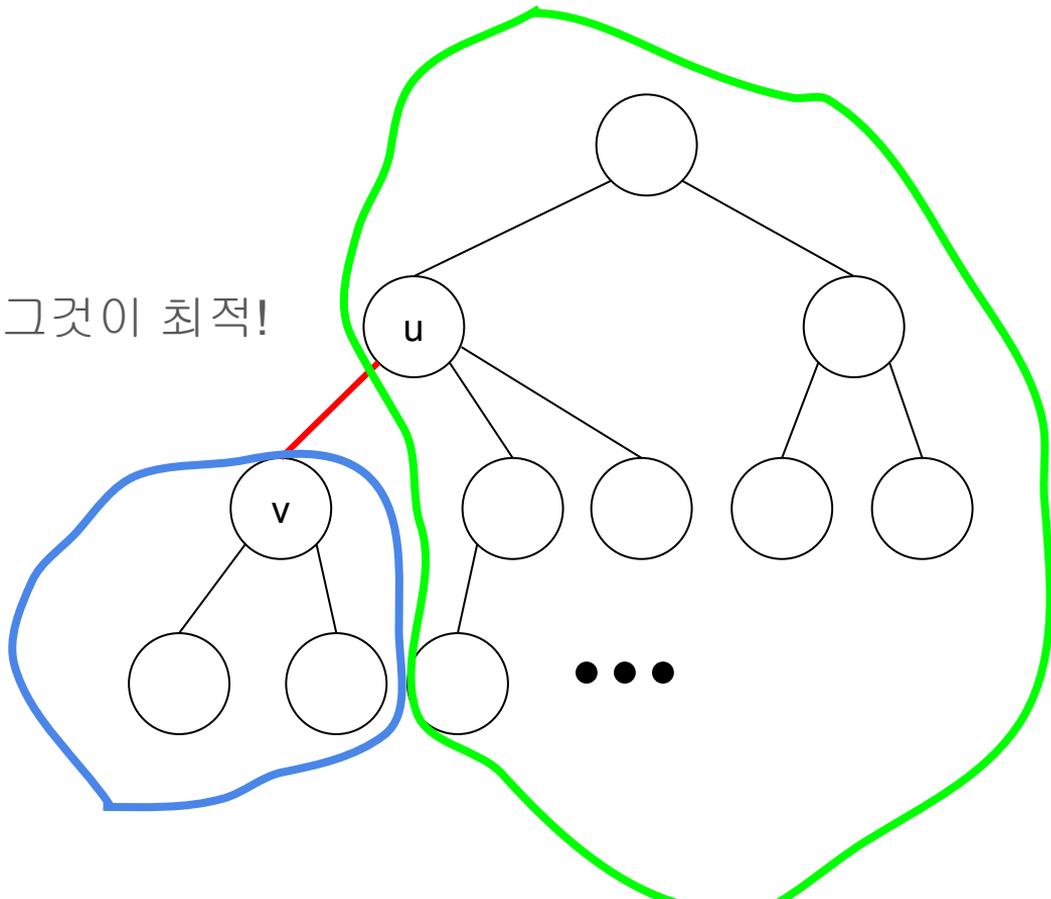
파란 부분의 지름과

초록 부분의 지름을 구한 후

두 지름을 $u-v$ 간선으로 이어주면 그것이 최적!

하지만 지름을 매번 직접 구하면

$O(N^2)$



C. 네트워크 해킹

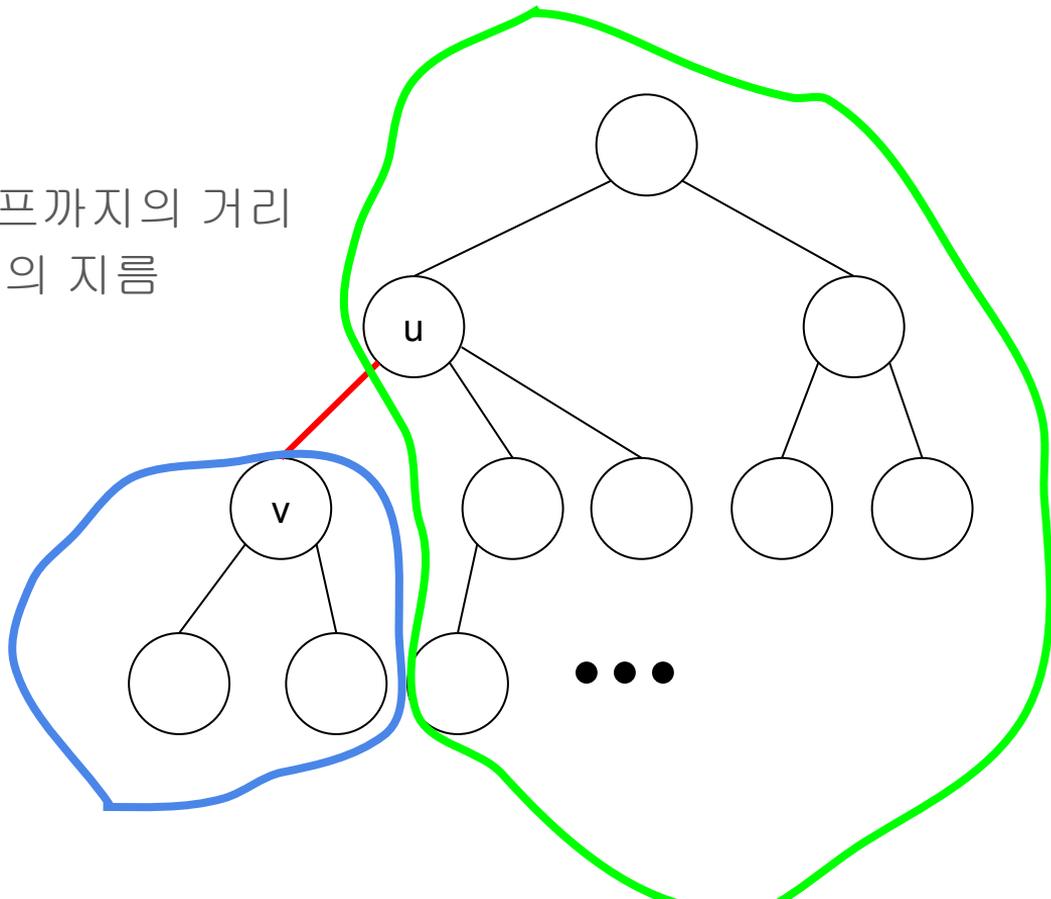
우선 각 노드마다

- 자신부터 1, 2, 3번째로 먼 리프까지의 거리
- 자신을 루트로 하는 서브트리의 지름

을 $O(N)$ DFS로 미리 구해두자.

이를 이용하면

파란 부분의 지름은 OK

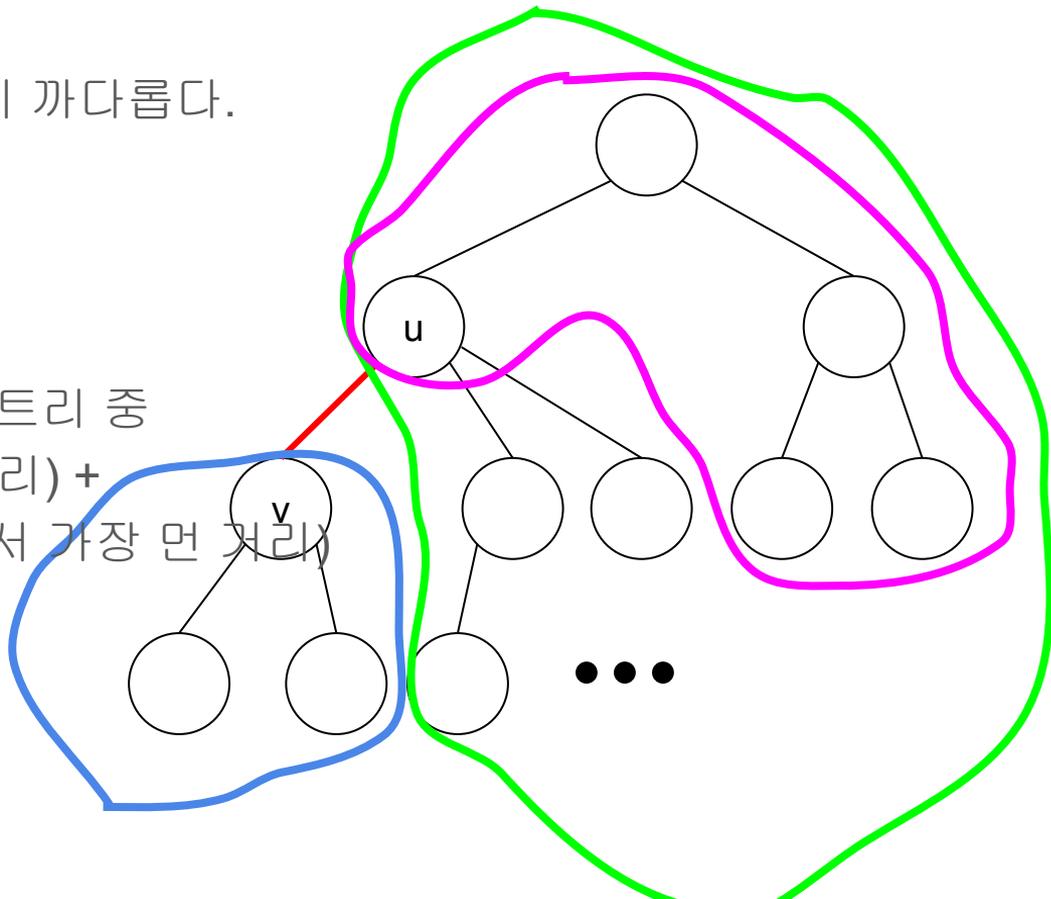


C. 네트워크 해킹

초록 부분의 지름을 구하는 부분이 까다롭다.

1. 자주색 부분의 지름
2. v -서브트리를 제외한 u -서브트리의 지름
3. (v -서브트리를 제외한 u -서브트리 중 u 에서 가장 먼 리프까지의 거리) + (u 에서 시작해 자주색부분에서 가장 먼 거리)

위 3가지 값 중 제일 큰 것이 초록 부분의 지름!



C. 네트워크 해킹

앞에서 전처리로 구한 값들을 이용하면

DFS를 해나가면서

파란 부분과 초록 부분의 지름을 $O(1)$ 에 계산할 수 있다.

K. 간단한 문제

정답 팀 : 14 (8 + 6)

First Solve : 더불어민규당 @ 구재현, 조민규, 이종원 (11 min.)

출제자 : ldtl

K. 간단한 문제

어차피 $A_i = 1$ 인 경우도 풀 수 있어야 하므로 결국 풀어야 하는 문제는 다음과 같습니다.

$$1 + \frac{2^k - 1}{n} = \prod_{i=1}^k \frac{1 + B_i}{B_i}$$

보통 이런 문제를 보면 k 가 커질 때의 답은 k 가 작을 때의 답을 이용해서 만들 수 있는 경우가 많습니다. 적당히 실험해서 패턴을 파악해 보거나 수식을 잘 때려맞추면 되겠습니다. 작은 k 에서 확장 가능한 것만 나열하도록 실험해보면 규칙이 잘 보입니다.

K. 간단한 문제

n 이 홀수인 경우 $b_k = n$

$$1 + \frac{2^k - 1}{n} = \frac{n + 1}{n} \prod \frac{1 + b_i}{b_i}$$

$$1 + \frac{2^{k-1} - 1}{(n + 1)/2} = \prod \frac{1 + b_i}{b_i}$$

K. 간단한 문제

n 이 짝수인 경우 $b_k = n + 2^k - 2$

$$1 + \frac{2^k - 1}{n} = \frac{n + 2^k - 1}{n + 2^k - 2} \prod \frac{1 + b_i}{b_i}$$
$$1 + \frac{2^{k-1} - 1}{n/2} = \prod \frac{1 + b_i}{b_i}$$

K. 간단한 문제

n 이 홀수든 짝수든 k 를 하나 줄일 수 있습니다.

n 은 계속 작아지므로 출력 범위는 만족할 수 있습니다.

B. 쉬운 최단경로 문제

정답 팀 : 4 (3 + 1)

First Solve : 남남서와 찌끄레기들 @ 신승원, 김동현, 김현수 (153 min.)

출제자 : ainta

B. 쉬운 최단경로 문제

다들
타

ainta  2:12 AM

2시간을솔나올까바 N을늘렸어요



ㅎ

$N \leq 400, Q \leq 400 \rightarrow N \leq 5000, Q \leq 10000$

☆최고의 선택☆

B. 쉬운 최단경로 문제

쿼리로 주어진 점이 p, q 라고 합시다.

p 와 q 가 모두 다각형 외부에 있는 경우, 답은 0이 됩니다.

p 와 q 중 외부에 있는 점이 있는 경우(일반성을 잃지 않고 q), p 에서 다각형 외부로 나갈 때 지나는 최소 밧줄의 개수를 구하는 문제가 됩니다.

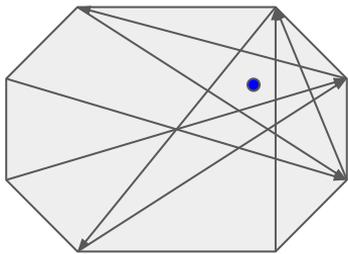
p 와 q 가 모두 내부에 있는 경우, \min (두 점에서 다각형 내부로 나갈 때 필요한 넘어야 하는 밧줄의 개수의 합, p 와 q 를 잇는 선분과 만나는 대각선의 개수)가 답이 됩니다.

B. 쉬운 최단경로 문제

다각형 내부에 점 p 가 주어졌다고 해봅시다.

다각형의 각 점 A_i 에 대해, $ccw(A[i], A[j], p) > 0$ 인 경우(반시계방향인 경우)의 j 의 index는 어떤 k 에 대해 $[i+1, k]$ 가 됩니다. (cyclic한 numbering에서)

각 i 에 대해 해당하는 k 를 pivot을 움직이면서 $O(N)$ 에 구해줄 수 있습니다. (이를 $F(i)$ 라 합시다)



B. 쉬운 최단경로 문제

1. 점 p 와 q 를 잇는 선분과 만나는 대각선 개수 구하기

p 와 q 에 대해 각각의 꼭지점 A_i 에서 $F_p(i)$ 와 $F_q(i)$ 를 구해놓았다고 할 때,

i 에서 출발하는 대각선 중 pq 와 만나는 것은 $|F_p(i) - F_q(i)|$ 개 입니다.

따라서, 이를 더해주면 됩니다.

B. 쉬운 최단경로 문제

2. 점 p 에서 다각형 외부로 갈 때 지나야 하는 최소 밧줄의 개수 구하기

$F(p)$ 를 구해놓으면 점 p 에서 출발하여 다각형의 각 변으로 빠져나갈때 지나야 하는 최소 밧줄의 개수를 누적합 배열을 이용해 구할 수 있습니다.

시간복잡도는 $F(p), F(q)$ 를 계산하는것이 $O(N)$ 에 가능하므로 $O(QN)$ 이 됩니다.

I. 피아의 아틀리에 ~신비한 생명의 연금술사~

정답 팀 : 6 (3 + 3)

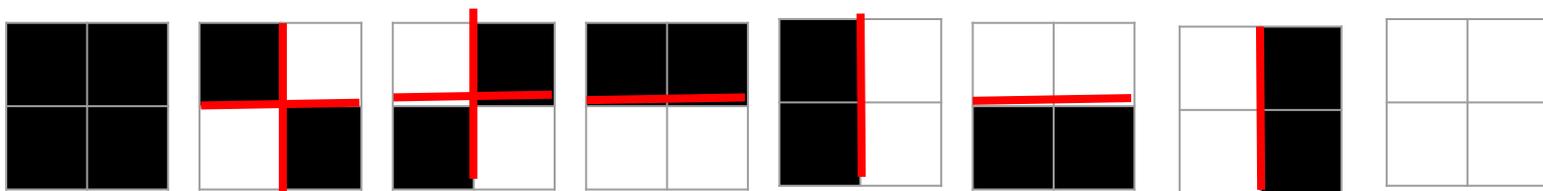
First Solve : 더불어민규당 @ 구재현, 조민규, 이종원 (133 min.)

출제자 : zigui

I. 피아의 아틀리에 ~신비한 생명의 연금술사~

1. 모든 칸이 짝수인 경우

2x2 배치로는 다음 8가지가 가능하다.

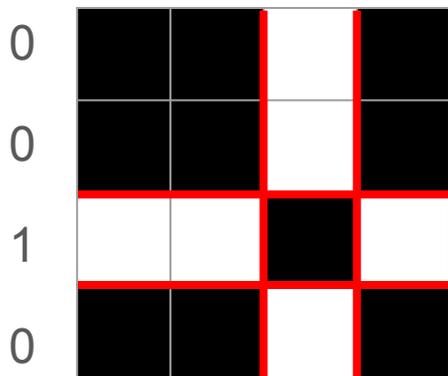


이어진 가로선이나 모든 세로선은 “상태”가 같다!!

(상태: 양쪽 칸이 서로 다른가)

I. 피아의 아틀리에 ~신비한 생명의 연금술사~

1 0 0 1 0



모든 가로선과 세로선 상태를 a_i, b_i 로 결정 $\rightarrow c_{ij} = a_i \wedge b_j \wedge d$

일부 칸이 결정되어 있는 것은 $a_i \wedge b_j \wedge d$ 값이 결정되는 것과 같다!

I. 피아의 아틀리에 ~신비한 생명의 연금술사~

하루만 구하기 : $2n, 2n+1$ 정점을 만든 뒤, Union & Find

(a, b)가 같으면 $(2a, 2b)$ 와 $(2a+1, 2b+1)$, 다르면 $(2a, 2b+1)$ 과 $(2a+1, 2b)$ 가 같은 상태

(a, b)를 합친 뒤, $(2a, 2a+1)$ 이나 $(2b, 2b+1)$ 이 같은 상태인지 확인.

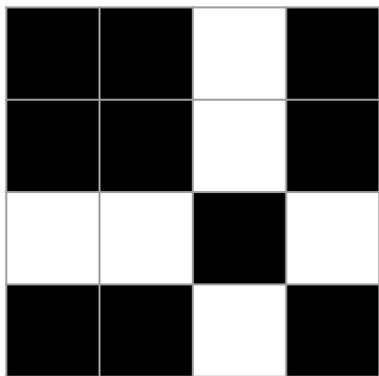
이제 Offline Dynamic Connectivity를 섞으면 문제를 풀 수 있다.

시간복잡도: $O(N \lg N \lg T)$

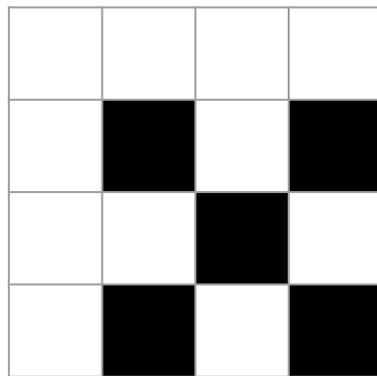
1. 피아의 아틀리에 ~신비한 생명의 연금술사~

2. 일부 칸이 홀수인 경우

(모든 칸이 짝수)



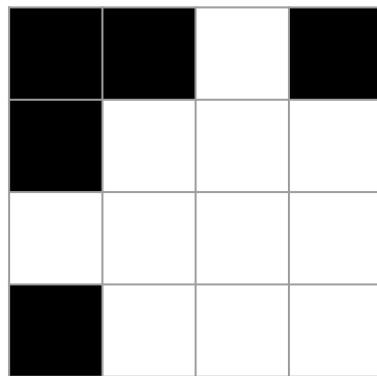
(조건을 만족하는 특수해)



+

=

(모든 상태)



조건을 만족하는 특수해 구하기는 매우 쉽기 때문에 생략합니다.

D. TV 동물 농장

정답 팀 : 1 (0 + 1)

First Solve : 더불어민규당 @ 구재현, 조민규, 이종원 (261 min.)

출제자 : zigui

D. TV 동물 농장

할 수 있는 것: i 번째 개와 $(j, j+1)$ 번째 고양이 + $(i, i+1)$ 번째 개와 j 번째 고양이

격자로 바꾼 뒤, 개를 행으로, 고양이를 열로 바꾸면 인접한 칸이 된다!

	$(i-1, j)$	
$(i-1, j)$	(i, j)	$(i, j+1)$
	$(i+1, j)$	

D. TV 동물 농장

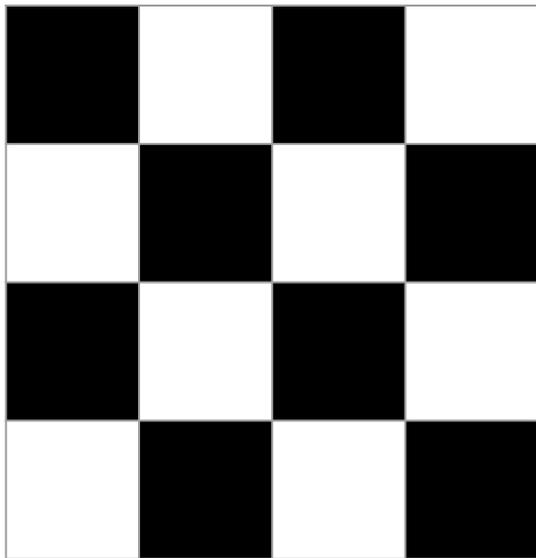
인접하면서 상태가 같은 두 칸을 바꿔야 하는데, 너무 어렵다. 하나만 더 바꿔보자!!

(검은색, 검은색) -> (흰색, 흰색)

체스판을 뒤집으면 다음 연산으로 바뀐다.

(검은색, 흰색) -> (흰색, 검은색)

이제 검은색을 말로 보고 열심히 옮겨보자!!!



D. TV 동물 농장

검은색 말의 출발점과 도착점이 정해져 있고, 한 번에 하나의 말만 움직일 수 있을 때,

이동하는 최소 횟수와 그 방법을 찾으시오.

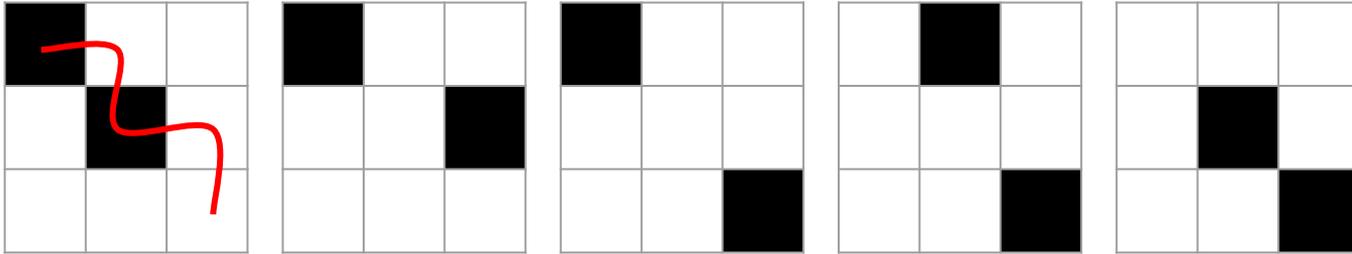
참고: 관심 없음 상태는 처음 상태와 끝 상태가 같아야 하며, 출발점과 도착점 수가 동일해야 함

그럴듯한 답: 가중치 매칭으로 최소 횟수를 찾자. (하한임은 자명하다)

하지만 말이 한 칸에 두 개 이상 있을 수 없다. 항상 가능할까?

D. TV 동물 농장

어떤 경로를 찾아도 그 거리만큼의 연산으로 옮길 수 있다!!



경로에 위치한 “뒤쪽 말”부터 경로를 따라 움직이면 **PROFIT!!**

이렇게 상한과 하한을 둘 다 증명했다.

(정해코드는 4.43 kB로 매우 길다)

F. parentheses recover

정답 팀 : 0 (0 + 0)

First Solve :

출제자 : ainta

F. parentheses recover

N 제한이 늘어난 문제 2 (제잘못아님)

F. parentheses recover

$O(N^3)$ 풀이

문자열 T가 주어져있을때, S와 T가 합쳐서 올바른 괄호 문자열이 될 수 있는지를 어떻게 판단할 수 있을까요?

F. parentheses recover

$$|S| = n, |T| = m$$

문자열 S 의 Sum배열을 $S1$, T 의 Sum배열을 $S2$ 라 할 때,

$(0,0)$ 에서 (n, m) 로 오른쪽 또는 위로만 가는 경로가 존재하여

지나는 모든 격자점 (x,y) 에 대해 $S1[x]+S2[y] \geq 0$ 을 만족하고 $S1[|S|] + S2[|T|] = 0$ 이면 S 와 T 를 합쳐 올바른 괄호 문자열을 만들 수 있고, 또한 그런 경우에만 만들 수 있습니다.

F. parentheses recover

$S1[0\sim n]$ 중 가장 큰 것을 $S1[p]$, $S2[0\sim m]$ 중 가장 큰 것을 $S2[q]$ 라고 합시다.

<정리> $(0,0)$ 에서 (n,m) 으로 조건을 만족하면서 갈 수 있다면, $(0,0)\rightarrow(p,q)\rightarrow(n,m)$ 으로 조건을 만족하면서 가는 경로가 존재한다.

증명) **the proof is left as an exercise to the reader**

F. parentheses recover

$S1[0\sim n]$ 중 가장 큰 것을 $S1[p]$, $S2[0\sim m]$ 중 가장 큰 것을 $S2[q]$ 라고 합시다.

<정리>(0,0)에서 (n,m)으로 조건을 만족하면서 갈 수 있다면, $(0,0)\rightarrow(p,q)\rightarrow(n,m)$ 으로 조건을 만족하면서 가는 경로가 존재한다.

이를 이용하면 $O(N^3)$ DP를 할 수 있습니다. ($0\sim p$, $0\sim q$ 부분과 $p\sim n$, $q\sim m$ 부분을 독립적으로 할 수 있기 때문)

F. parentheses recover

$S1[0\sim n]$ 중 가장 큰 것을 $S1[p]$, $S2[0\sim m]$ 중 가장 큰 것을 $S2[q]$ 라고 합시다.

<정리>(0,0)에서 (n,m)으로 조건을 만족하면서 갈 수 있다면, (0,0)->(p,q)->(n,m)으로 조건을 만족하면서 가는 경로가 존재한다.

이를 이용하면 $O(N^3)$ DP를 할 수 있습니다. (0~p, 0~q 부분과 p~n, q~m 부분을 독립적으로 할 수 있기 때문)

$N \leq 3000??$

F. parentheses recover

$O(N^2)$ 풀이

ainta : $O(N^3)$ 어려움

zigui : $O(N^2)$ 쉬움

zigui처럼 잘하시면 풀수있습니다.

화이팅!

F. parentheses recover

문자열 S , T 가 조건을 만족할 때, 이 두 문자열을 “ainta 문자열쌍”이라고 하자.

$D[i][j]$:= T 의 글자 개수가 i 개이고, $S[1, k]$ 와 T 가 “ainta 문자열쌍”이 되는 k 의 최댓값이 j 일 때의 T 의 가짓수.

ex) $S = “(())”$, 길이 3

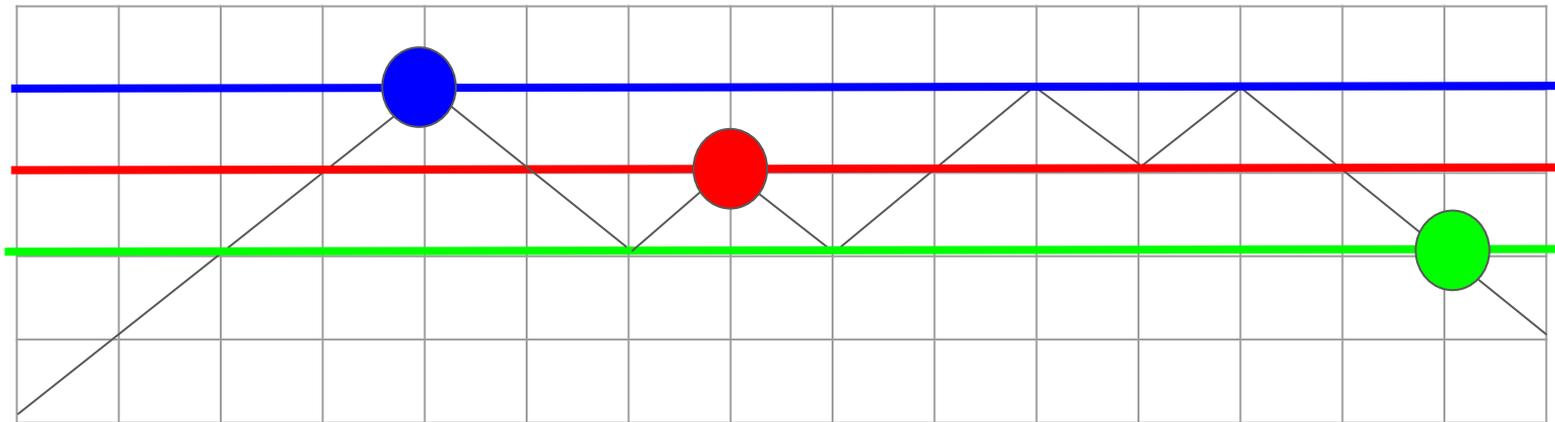
0	0	1
0	1	0
0	0	3

“))”인 경우 j 의 최댓값은 2이므로 $D[2][2] = 2$

F. parentheses recover

$next[i] := (i\text{보다 큰 위치 중 } sum[j]\text{가 } sum[i]-2\text{보다 작거나 같은 최소 위치}) - 1$

$prv[i] := (i\text{보다 작은 위치 중 } sum[j]\text{가 } sum[i]+1\text{보다 크거나 같은 최대 위치})$



(수정) $D[i+1][next[i]] += D[i][j]$, $D[i+1][prv[i]] += D[i][j]$

F. parentheses recover

답은? 먼저 S 뒤에 ')'를 엄청나게 많이 붙이면 $D[L][|S|]$ 가 답이 된다.

최댓값이 $|S|$ 보다 큰 경우: '(', ')' 개수가 안맞음

최댓값이 $|S|$ 보다 작은 경우: 올바른 괄호 문자열을 만들 수 없음

prv, nxt를 전처리하면, $O(N^2)$ 에 해결할 수 있다!

심지어 1KB 내에 된다!!!

(추가) <http://zigui.tistory.com/27> 에 자세한 풀이를 작성하였습니다.

끝

감사합니다!