

2019 UOSPC 문제해설

컴퓨터과학부 16 최문기

출제 후기

1. 좋은 문제를 만드는 것은 쉽지 않다. 평소에 아이디어를 생각해 두어야 한다. 대회에 가까워져서 필요한 난이도에 적당한 알고리즘에 맞춰서 문제를 만들면 문제의 질이 낮아진다.
2. 테스트 케이스를 만드는 건 절대로 간단하지 않다. 랜덤으로 되는게 아니라 잘못된 풀이가 틀리는 테스트 케이스가 철저하게 있어야 하고 솔루션 코드가 정확한지도 중요하다.

(백준 자유게시판) 대회의 질에 대해

<https://www.acmicpc.net/board/view/43681>



Professional way to prepare programming contest problem

Test script

Solutions (AC, WA, TLE)

Checker

Data generator

validator

Tests

[Preview Tests](#) [Add Tests](#)

#	Content	Size	Description	Example	Actions				
					Delete	Copy	Move	Example	
1	data_gen1 1 20 3 5 1				Delete	Edit	Example		<input type="checkbox"/>
2	data_gen1 1 50 10 20 2				Delete	Edit	Example		<input type="checkbox"/>
3	data_gen1 1 100 20 30 3				Delete	Edit	Example		<input type="checkbox"/>
4	data_gen1 1 200 30 40 4				Delete	Edit	Example		<input type="checkbox"/>
5	data_gen1 1 500 40 50 5				Delete	Edit	Example		<input type="checkbox"/>
6	data_gen1 1 700 50 60 6				Delete	Edit	Example		<input type="checkbox"/>
7	data_gen1 1 800 60 70 7				Delete	Edit	Example		<input type="checkbox"/>
8	data_gen1 1 900 80 90 8				Delete	Edit	Example		<input type="checkbox"/>
9	data_gen1 1 1000 90 100 9				Delete	Edit	Example		<input type="checkbox"/>
10	data_gen1 1 1000 99 100 10				Delete	Edit	Example		<input type="checkbox"/>
11	data_gen1 1 1000 99 100 11				Delete	Edit	Example		<input type="checkbox"/>
12	data_gen1 1 1000 99 100 12				Delete	Edit	Example		<input type="checkbox"/>
13	data_gen1 1 1000 99 100 13				Delete	Edit	Example		<input type="checkbox"/>
14	data_gen1 100 1000 99 100 14				Delete	Edit	Example		<input type="checkbox"/>
15	data_gen1 500 1000 99 100 15				Delete	Edit	Example		<input type="checkbox"/>
16	data_gen1 500 1000 99 100 16				Delete	Edit	Example		<input type="checkbox"/>
17	data_gen1 10 10 1 1 17				Delete	Edit	Example		<input type="checkbox"/>
18	data_gen1 10 10 5 5 18				Delete	Edit	Example		<input type="checkbox"/>

[Check generators to be compilable](#)

Script:

```
data_gen1 1 20 3 5 1 > $
data_gen1 1 50 10 20 2 > $
data_gen1 1 100 20 30 3 > $
data_gen1 1 200 30 40 4 > $
data_gen1 1 500 40 50 5 > $
data_gen1 1 700 50 60 6 > $
data_gen1 1 800 60 70 7 > $
data_gen1 1 900 80 90 8 > $
data_gen1 1 1000 90 100 9 > $
data_gen1 1 1000 99 100 10 > $
data_gen1 1 1000 99 100 11 > $
data_gen1 1 1000 99 100 12 > $
data_gen1 1 1000 99 100 13 > $
data_gen1 100 1000 99 100 14 > $
data_gen1 500 1000 99 100 15 > $
data_gen1 500 1000 99 100 16 > $
data_gen1 10 10 1 1 17 > $
data_gen1 10 10 5 5 18 > $
```

Drafts

Generator is a program which outputs single test to the standard output. It is highly recommended to use [testlib](#) to write generators. Each line of the script should have a form:

```
generator-name [params] > test-index
```

or

```
generator-name [params] > $
```

Each dollar signs will be automatically replaced with the smallest available test index.

Other form to use generators is:

```
generator-name [params] > {test-indices}
```

In this case generator should produce one or more files named as test-indices. This form does not support dollars.

First usage example:

```
gen 7 0.2 abaca > 13
```

(generator gen will write test in stdout)

1. 무빙워크 (Div.2)
2. 우리 어디서 만날래? (Div.1)
3. UOS Coder (Div.2, Div.1)
4. 팰린드롬 게임 (Div2, Div.1)

무빙워크

1. 가장 긴 두 건물 사이의 경로의 길이
2. 이때 임의의 두 건물 사이의 경로는 유일하다. 즉, 이동 경로에 사이클이 없다.

= 트리의 지름

무빙워크 [풀이 1] 트리 DP

u를 루트로 하는 서브 트리의 지름을 $D(u)$ 라고 하면

i) 자식 노드가 2개 이상

$D(u) = (\text{첫 번째로 큰 리프 노드까지의 거리}) + (\text{두 번째로 큰 리프 노드까지의 거리})$

ii) 자식 노드가 1개

$D(u) = (\text{첫 번째로 큰 리프 노드까지의 거리})$

무빙워크 [풀이 1] 트리 DP

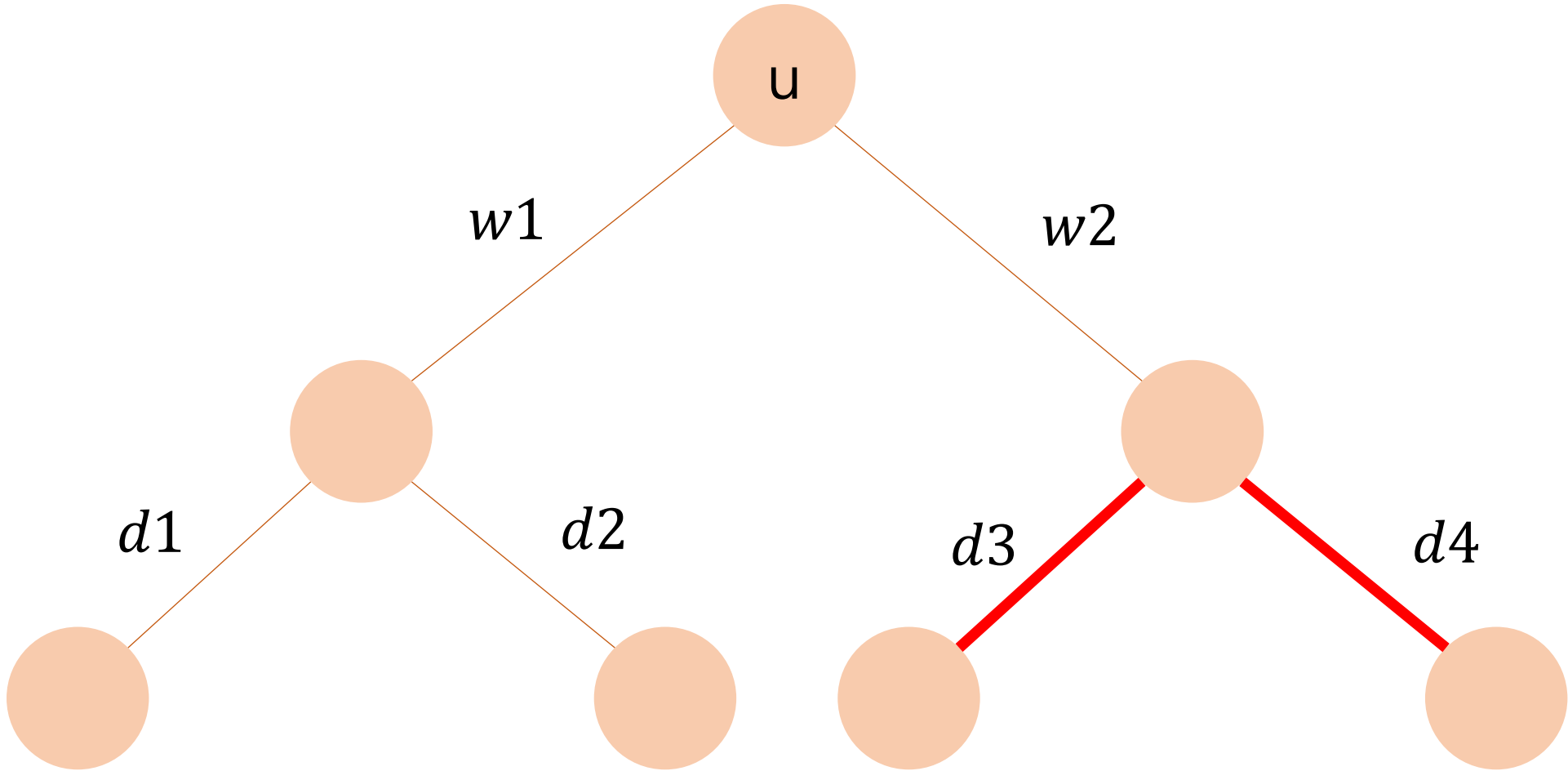
u에서 첫 번째로 먼 리프 노드까지의 거리를 $dp1[u]$
두 번째로 먼 리프 노드까지의 거리를 $dp2[u]$ 라 하면

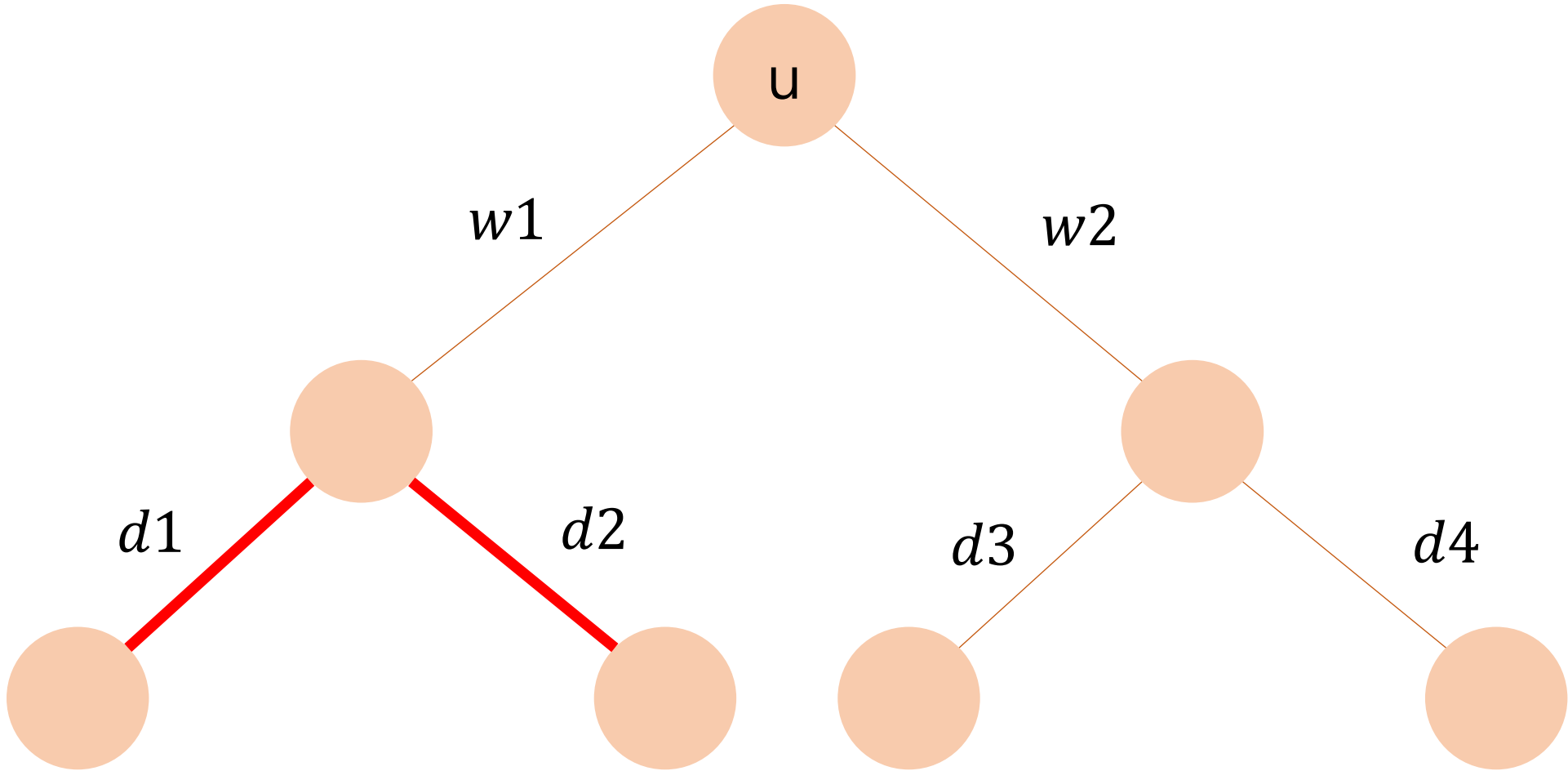
i) 자식 노드가 2개 이상

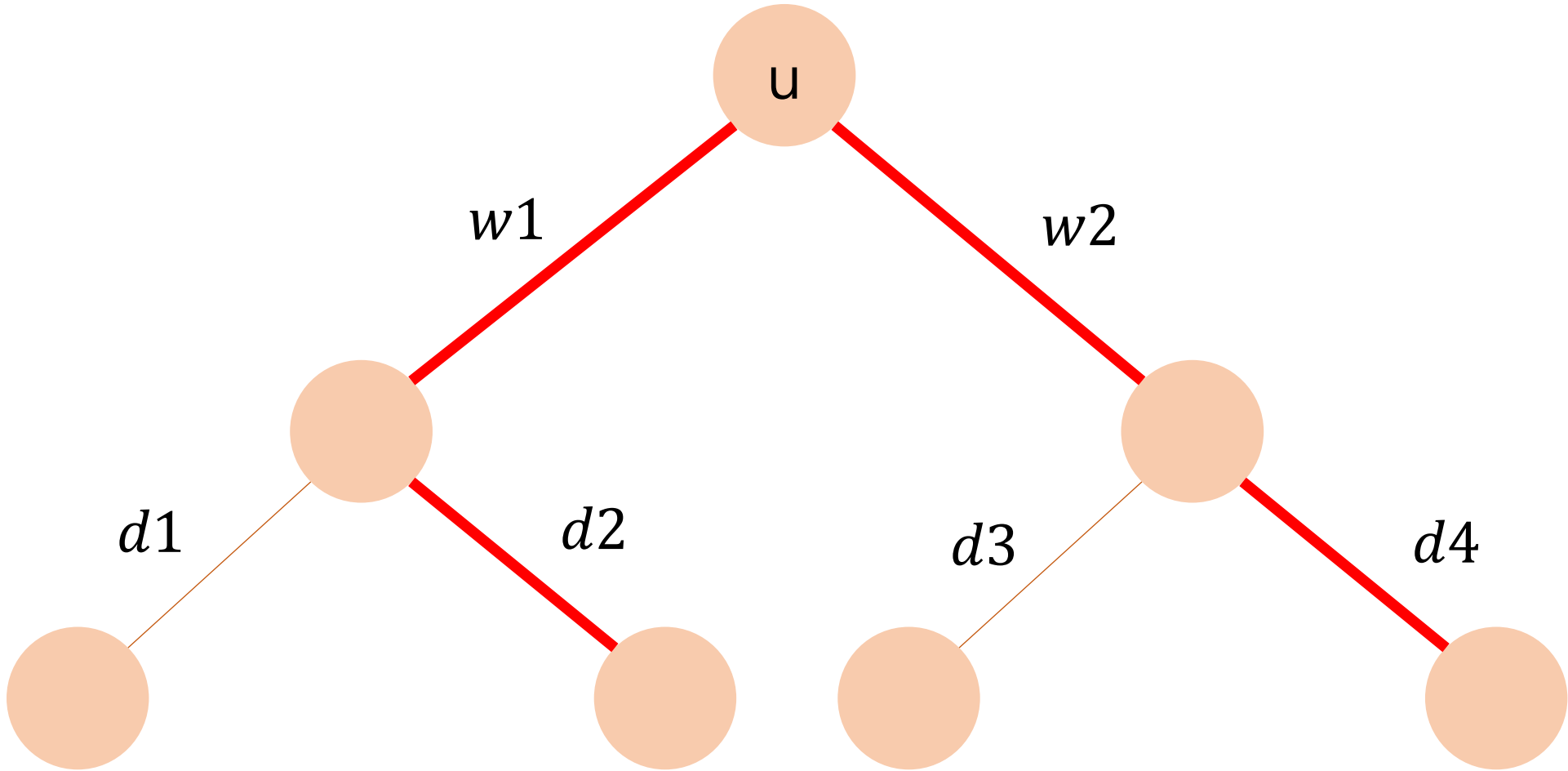
$$D(u) = dp1[u] + dp2[u]$$

ii) 자식 노드가 1개

$$D(u) = dp1[u]$$







```

def dfs(v):
    global ans
    vst[v] = True
    weight = [] # 리프 노드까지의 거리
    for u, w in g[v]:
        if not vst[u]:
            weight.append(dfs(u) + w)
    weight.sort()

    if len(weight) > 1:
        ans = max(ans, weight[-1] + weight[-2]) # 최댓값 갱신
        return weight[-1]

    elif len(weight) == 1:
        ans = max(ans, weight[-1]) # 최댓값 갱신
        return weight[-1]

    return 0

dfs(0)
print(ans)

```

$D[u]$, $dp1[u]$, $dp2[u]$ 를
메모이제이션 할 필요는 없다.

무빙워크 [풀이 2]

1. 어떤 정점 u 에서 가장 먼 정점 v 을 찾는다.
2. 정점 v 에서 가장 먼 정점 t 를 찾는다.
3. $v - t$ 는 트리의 지름이다.

가장 간결한 풀이. 생각해내기 어렵다.

정당성 증명 (및 참고자료)

-> <https://www.weeklyps.com/entry/%ED%8A%B8%EB%A6%AC%EC%9D%98-%EC%A7%80%EB%A6%84>

BOJ #1967 트리의 지름

<https://www.acmicpc.net/problem/1967>

BOJ #3482 Labyrinth

<https://www.acmicpc.net/problem/3482>

BOJ #17969 Gene Tree (2019 ICPC Seoul Regional)

<https://www.acmicpc.net/problem/17969>

우리 어디서 만날래?

동창 친구들이 모두 최단 시간으로 약속 장소에 이동한다고 할 때

모두 d 안에 이동할 수 있으면 Y 를 출력하시오.

d 안에 이동할 수 없거나 누군가

약속 장소로 이동 할 수 있는 경로가 없으면 N 을 출력하시오.

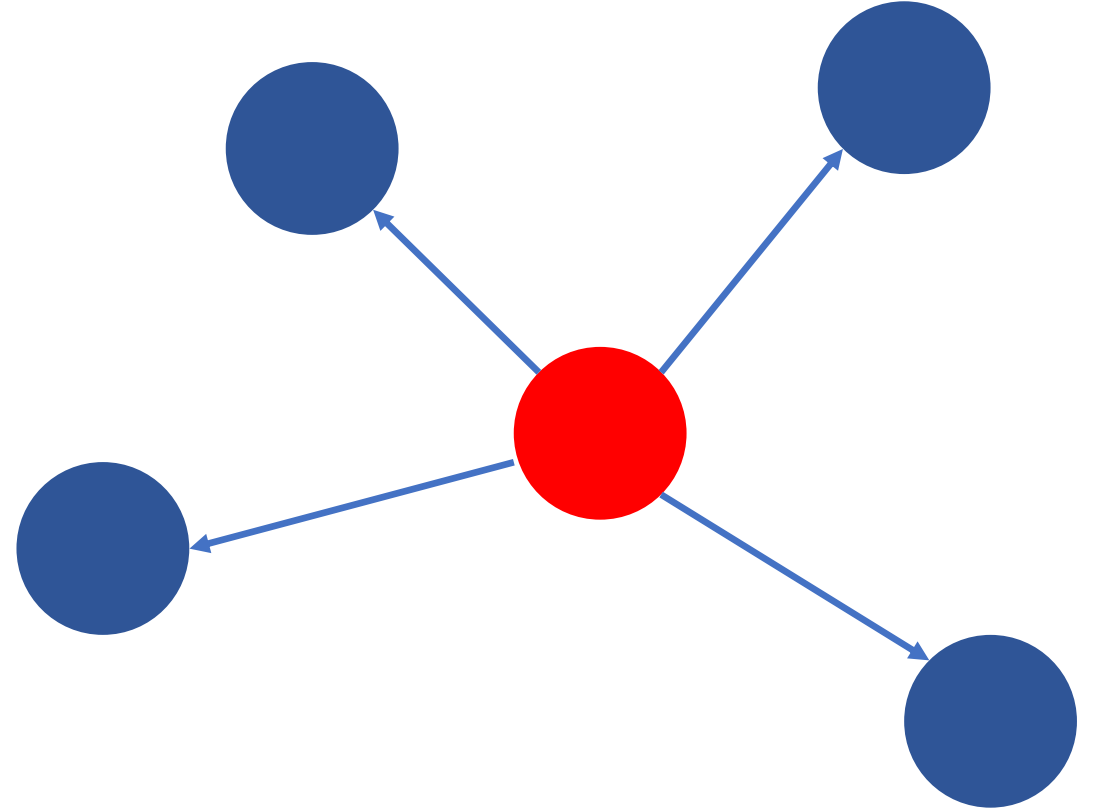
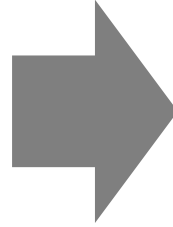
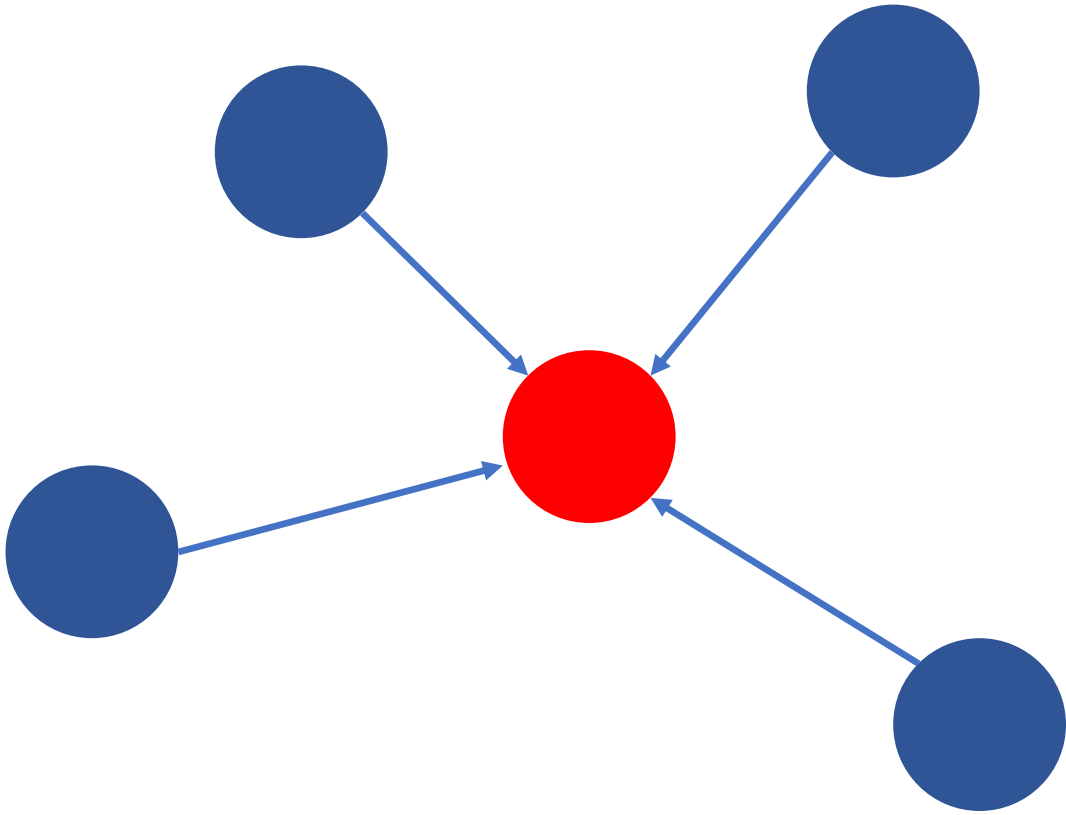
우리 어디서 만날래?

N번 다익스트라를 하면 시간초과

간선의 방향을 반대로 해서

목적지에서 한번의 다익스트라

우리 어디서 만날래?




```
cin >> N >> M >> C;
C -= 1;

for (int i = 0, u, v, w; i < M; i++) {
    cin >> u >> v >> w;
    adj[v - 1].push_back({ u - 1, w });
}
cin >> D;
fill(dist, dist + N, 1e9);
```

u -> v 를

v -> u 로 저장

UOS Coder

온라인 저지 사이트 'UOS Coder'에서는 정기적으로 프로그래밍 대회를 연다.

UOS Coder에는 자신의 실력을 나타내는 지표인 레이팅이 있다.

매번 열리는 프로그래밍 대회에서 학생들은 경쟁하고 등수에 따라 레이팅이 올라가거나 내려간다.

UOS Coder의 운영자 MK는 새로운 포인트 시스템을 도입하려고 한다.

분기별로 유저는 목표 레이팅을 설정하고 레이팅을 달성할 때마다 10 포인트를 얻을 수 있다.

즉, k 번째 대회 후 레이팅을 R_k , 목표 레이팅을 R_g 라 하면 다음 식이 성립할 때 대회가 끝나고 10 포인트를 얻는다. ($2 \leq k \leq N$)

$$R_{k-1} < R_g \leq R_k$$

MK는 새로운 포인트 시스템을 도입하기 전에 지난 이용자가 얻을 수 있었던 최대 포인트를 계산해보려고 한다.

MK를 도와서 이용자가 얻을 수 있었던 최대 포인트를 계산하는 프로그램을 작성해보자.

UOS Coder (Div.2)

$$1 \leq N \leq 100 \quad 1 \leq R_i \leq 1000$$

목표 레이팅을 1부터 1000까지 설정해서 완전탐색하면

최대 100000이므로 시간 내로 가능

(풀이 생략)

UOS Coder (Div.1)

$$1 \leq N \leq 100,000$$

$$1 \leq T \leq 10$$

$$1 \leq R_i \leq 1,000,000,000$$

완전탐색하면 시간초과

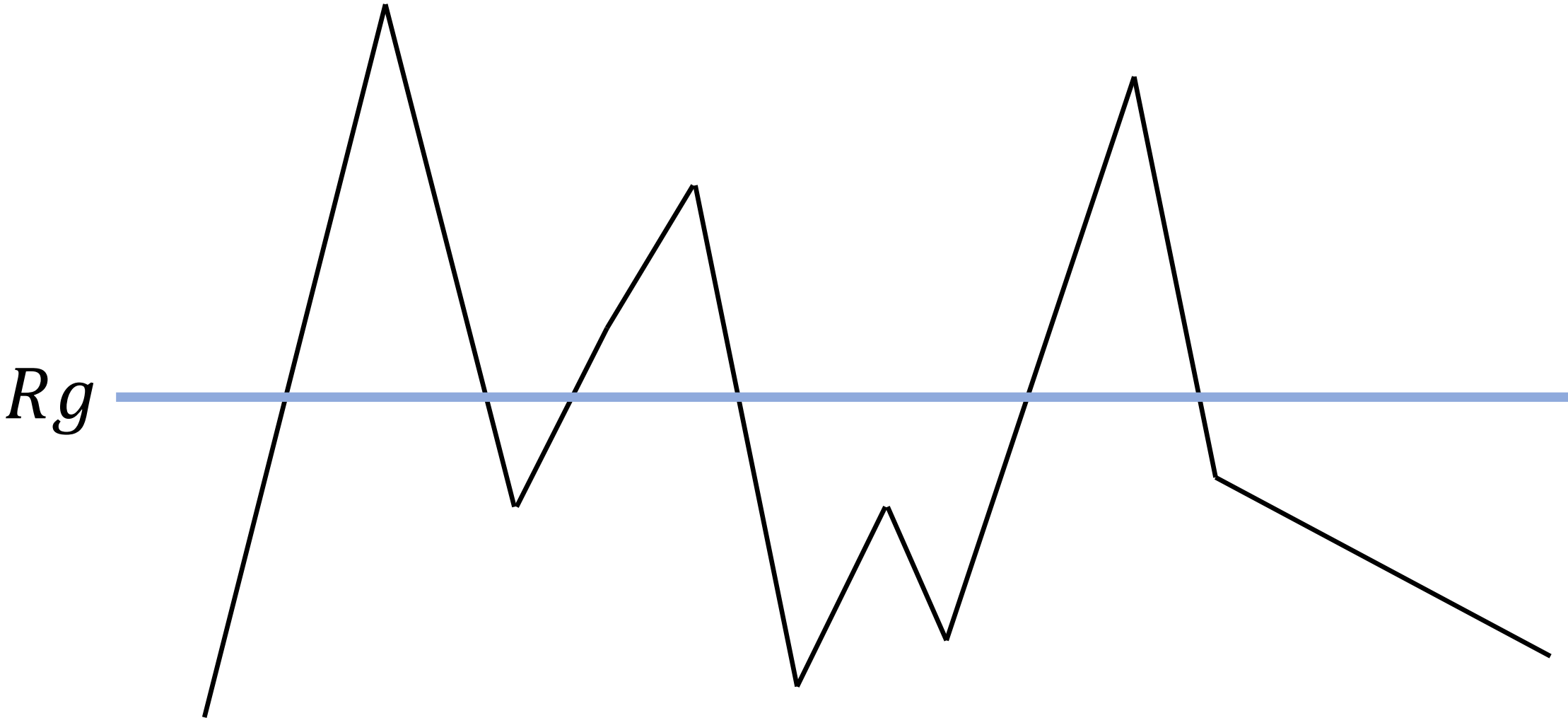
UOS Coder (Div.1)

증가하는 직선과

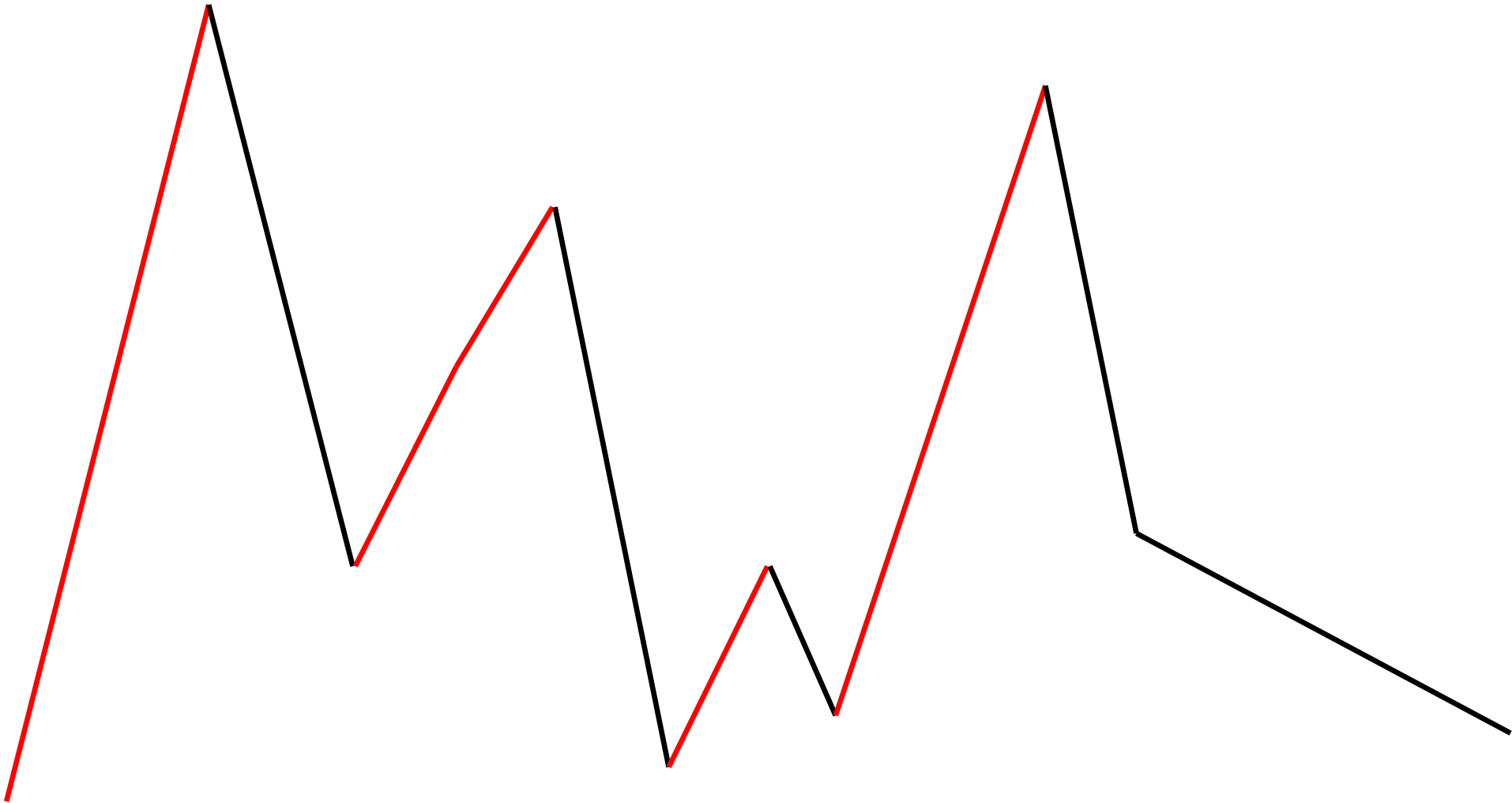
x 축에 평행한 직선(목표레이팅)과의

교점의 수를 최대로

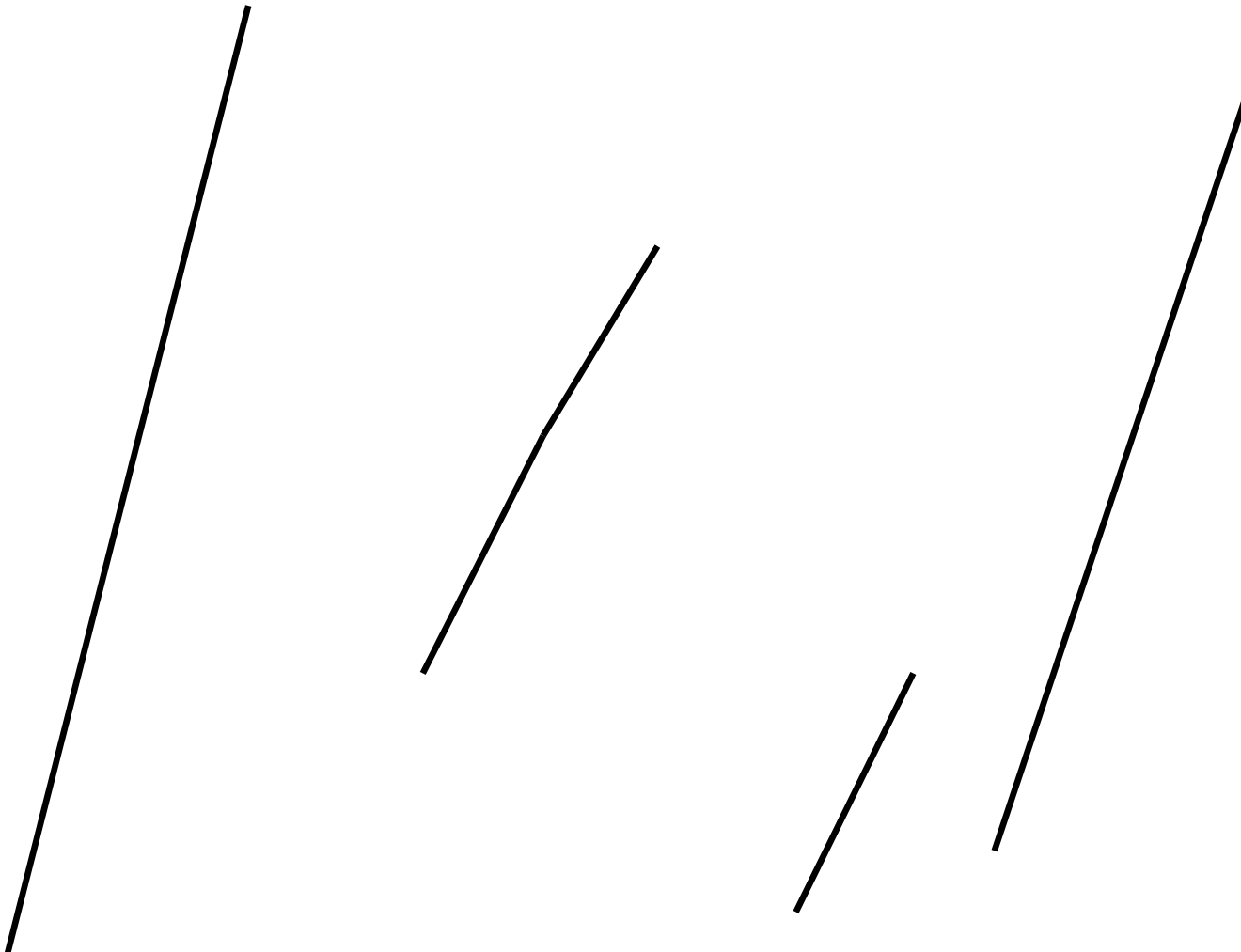
UOS Coder (Div.1)



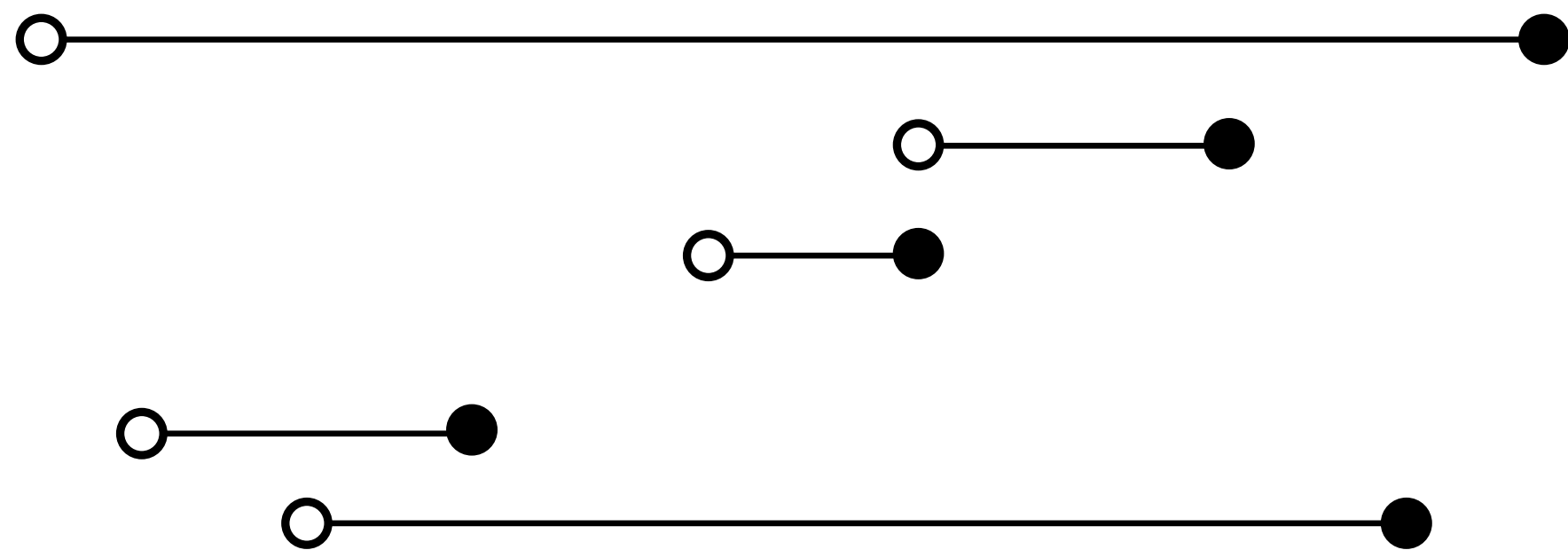
UOS Coder (Div.1)



UOS Coder (Div.1)



UOS Coder (Div.1)



UOS Coder (Div.1)

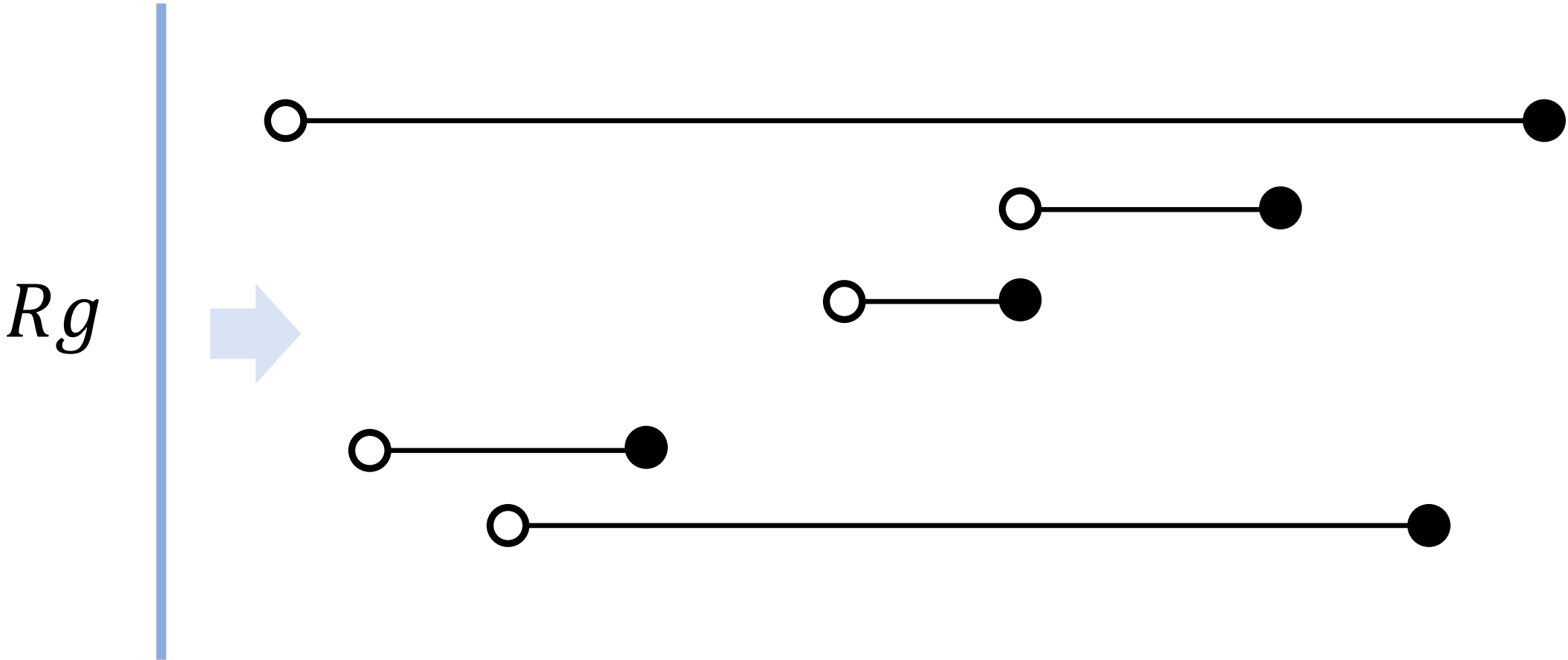
레이팅이 a 에서 b 로 증가했다면

points 배열에 $(a, 1), (b, -1)$ 저장

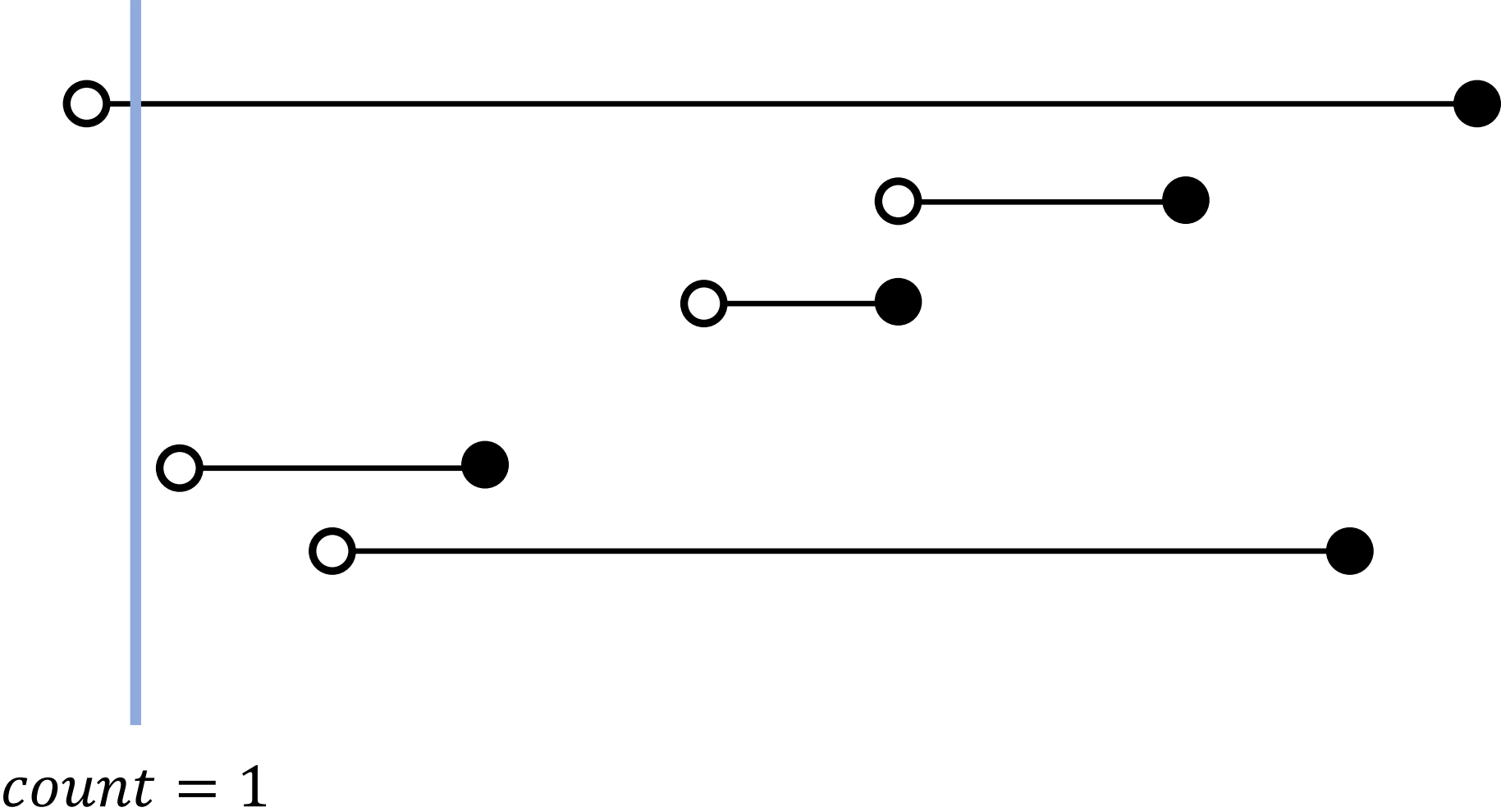
그리고 오름차순으로 정렬 ($O(n \log n)$)

```
while (T--) {  
    vector<int> rate(N);  
    for (int i = 0; i < N; i++)  
        cin >> rate[i];  
  
    vector<point> v;  
    for (int i = 0; i < N - 1; i++) {  
        int a = rate[i];  
        int b = rate[i + 1];  
        if (a <= b) {  
            v.push_back(point(a, 1));  
            v.push_back(point(b, -1));  
        }  
    }  
    sort(v.begin(), v.end());  
}
```

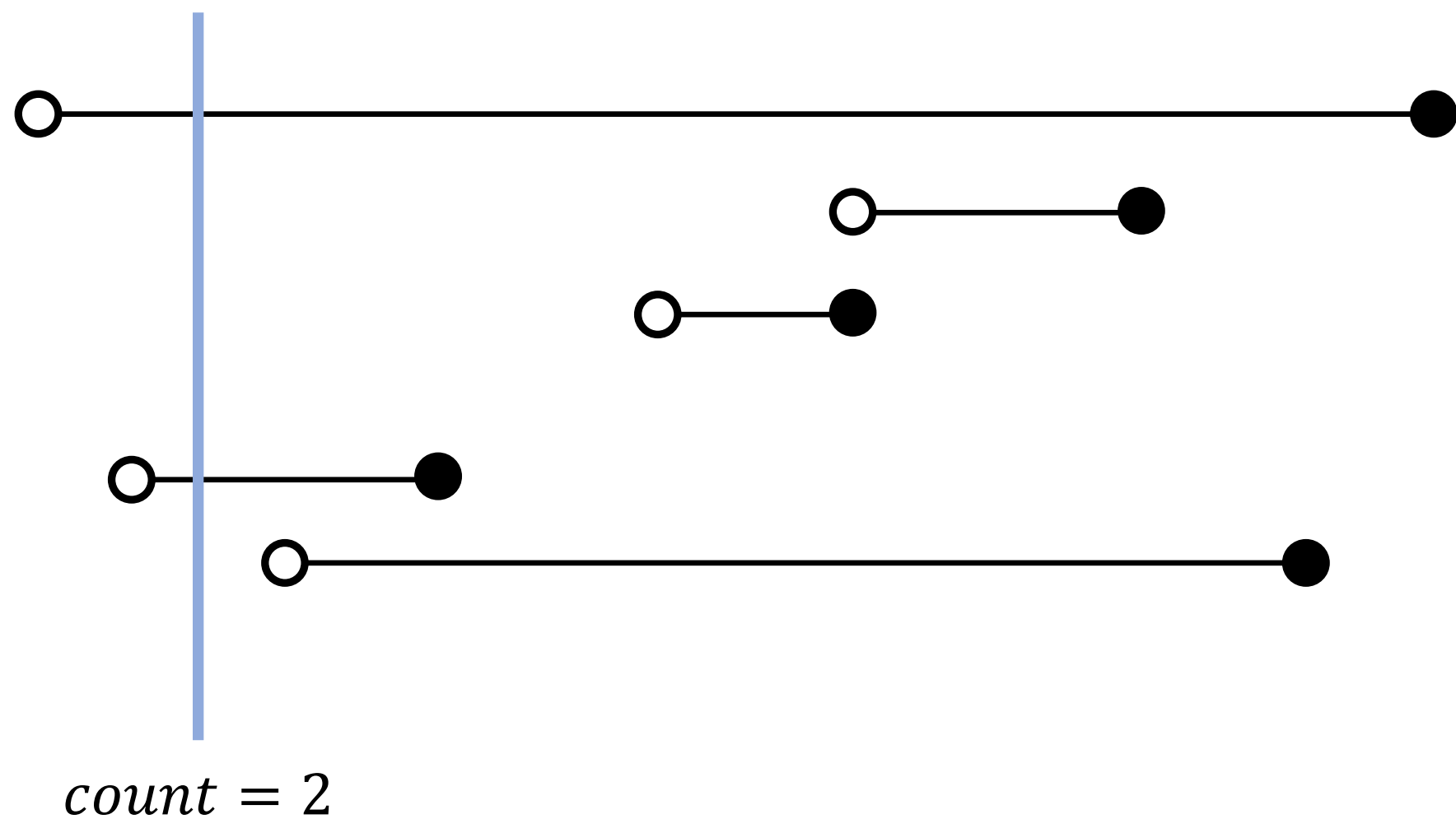
UOS Coder (Div.1)



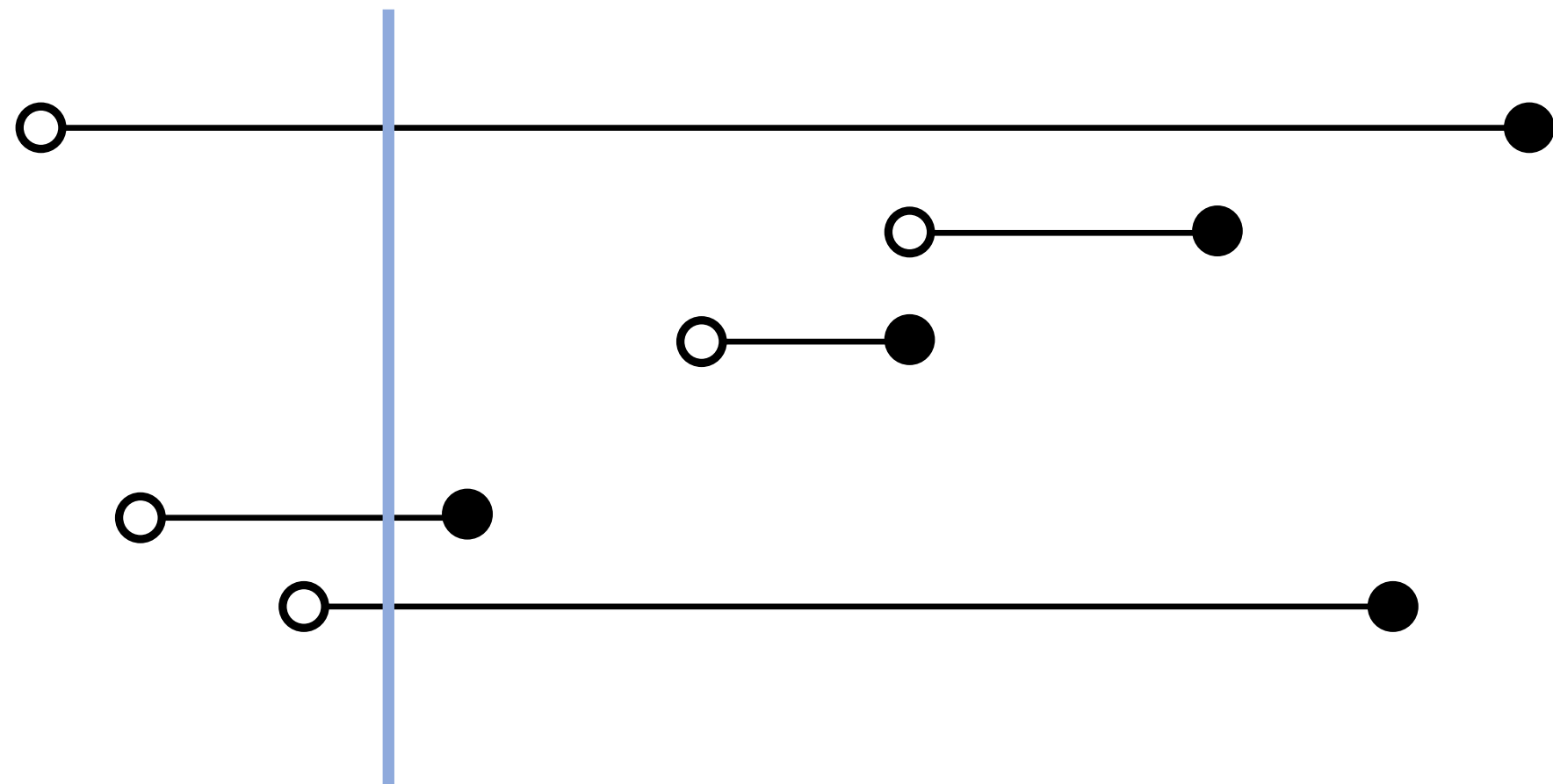
UOS Coder (Div.1)



UOS Coder (Div.1)

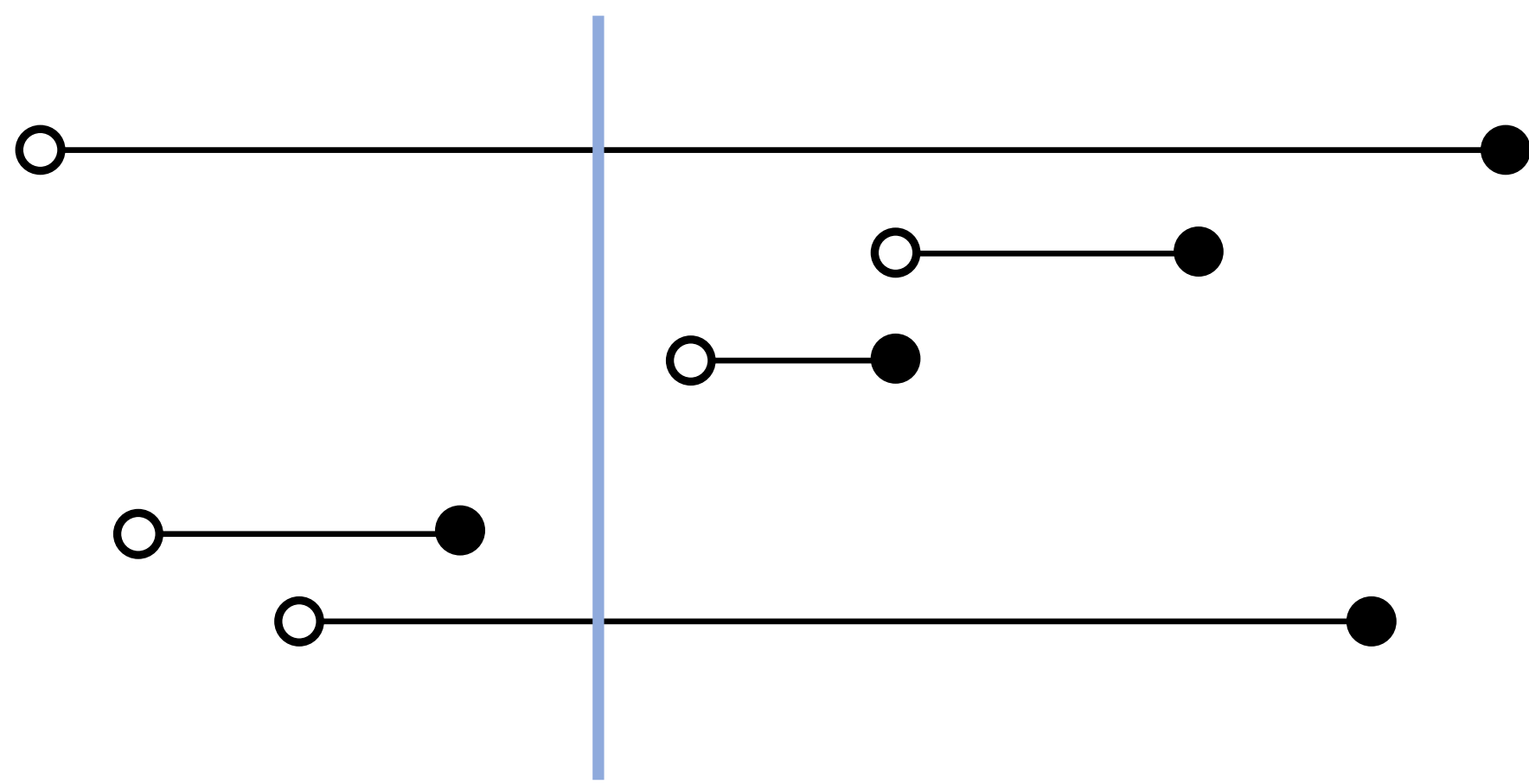


UOS Coder (Div.1)



count = 3

UOS Coder (Div.1)



count = 2

UOS Coder (Div.1)

오름차순으로 count를 갱신하면서

count값의 최댓값을 탐색

(라인 스위핑)

```
int cnt = 0;
int cnt_max = 0;
for (point i : v) {
    cnt += i.second;
    if (cnt > cnt_max)
        cnt_max = cnt;
}
ans += 10 * cnt_max;
}

cout << ans;
```

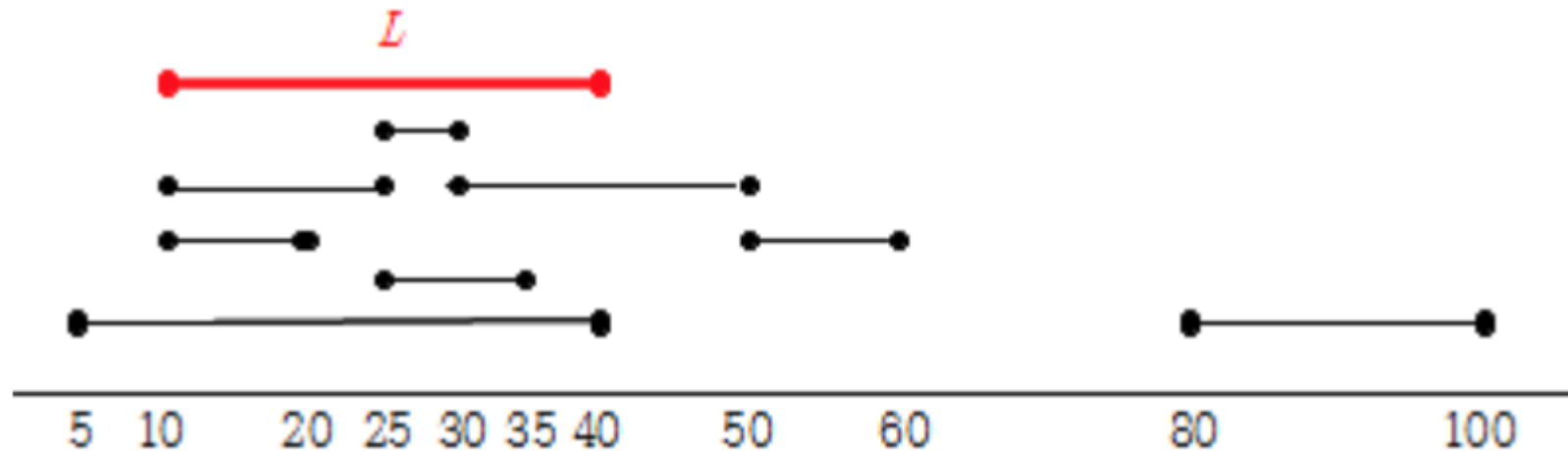
BOJ #1689 겹치는 선분

<https://www.acmicpc.net/problem/1689>

“선분이 최대로 겹쳐있는 부분의 겹친 선분의 개수”

BOJ #13334 첼로

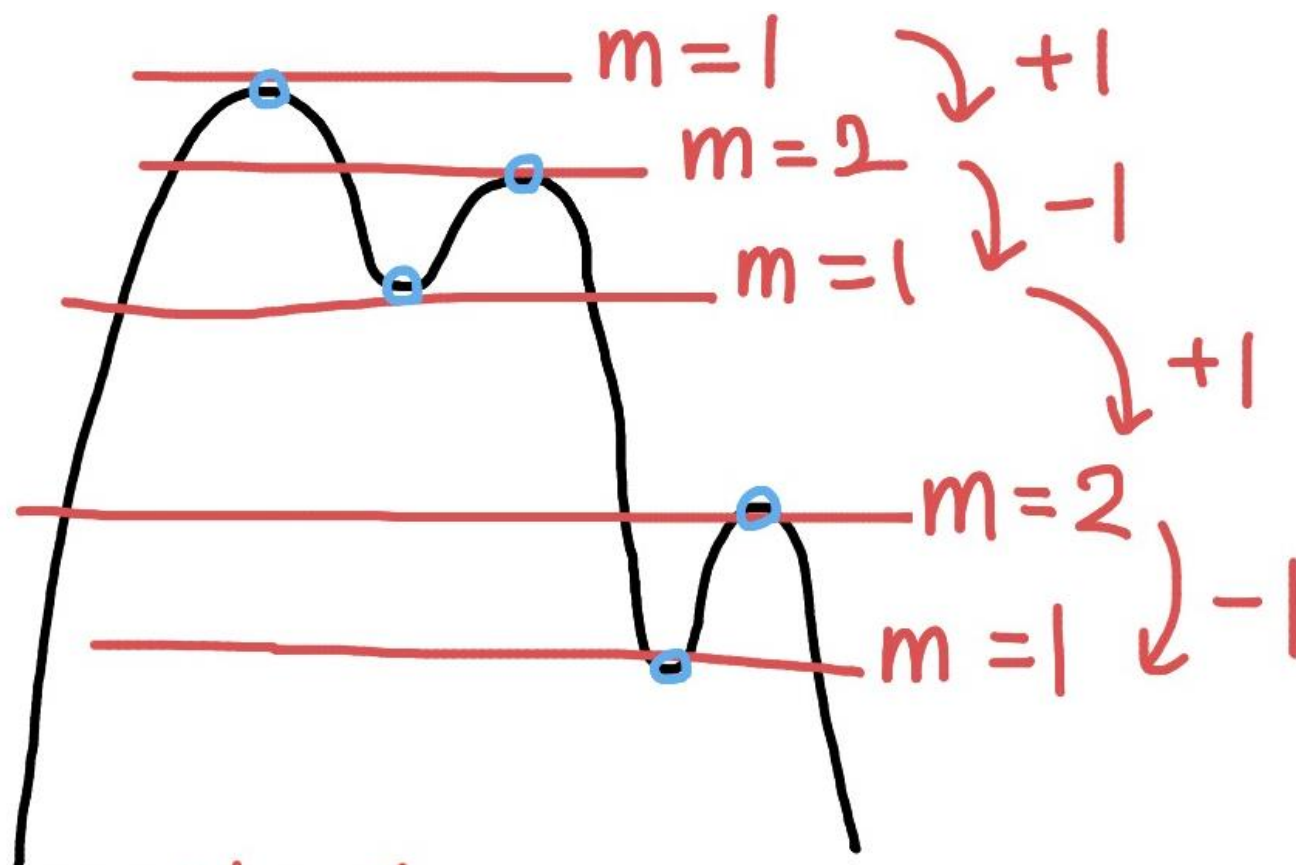
<https://www.acmicpc.net/problem/13334>



BOJ # 16788 [JOI 2019 예선 4번] 일본침몰

<https://www.acmicpc.net/problem/16788>

“수면이 차오를 때 최대 섬의 개수”



$m = \text{섬의 수}$
 $\circ = \text{극대, 극소}$

Codeforces Round #600 (Div. 2) D. Harmonious Graph

<http://codeforces.com/contest/1253/problem/D>

“If from a node l we can reach a node r through edges ($l < r$), then we should be able to reach nodes $(l+1), (l+2), \dots, (r-1)$ too.”

“Print the minimum number of edges we have to add to the graph to make it harmonious.”

팰린드롬 게임 (Div.2)

팰린드롬(palindrome) 수열이란 앞에서부터 읽을 때와 뒤에서부터 읽을 때 똑같은 수열을 말한다.

예를 들어 $\{3\}$, $\{2, 2\}$, $\{3, 3, 3\}$, $\{3, 2, 2, 3\}$, $\{1, 2, 2, 1\}$, $\{3, 2, 1, 2, 3\}$ 은 팰린드롬 수열이고 $\{2, 3\}$, $\{1, 2, 3\}$, $\{3, 2, 1, 3\}$ 은 팰린드롬 수열이 아니다.

팰린드롬 게임은 주어진 수열을 여러 개의 팰린드롬 부분수열로 나누는 게임이다.

팰린드롬 게임의 점수의 획득은 다음과 같다.

Case 1) 길이가 3이상이면서 중심으로 갈수록 증가하는 팰린드롬 수열 : 1점

Case 2) 길이가 3이상이면서 중심으로 갈수록 감소하는 팰린드롬 수열 : 2점

Case 3) 길이가 3미만이거나 Case 1) 또는 Case 2)가 아닌 팰린드롬 수열 : 3점

팰린드롬 게임 (Div.2)

“수열 A 의 팰린드롬 수열의 점수를 출력한다.

만약 팰린드롬 수열이 아니면 D 를 출력한다.”

팰린드롬 게임 (Div.2)

1. 팰린드롬 수열인가
 2. 중심까지 증가하는가
 3. 중심까지 감소하는가
- (풀이 생략)

실행 결과

Test case #1: AC [0.011s, 824.00 KB] (1/1)
 Test case #2: AC [0.007s, 824.00 KB] (1/1)
 Test case #3: AC [0.009s, 724.00 KB] (1/1)
 Test case #4: AC [0.008s, 780.00 KB] (1/1)
 Test case #5: AC [0.011s, 820.00 KB] (1/1)
 Test case #6: AC [0.009s, 764.00 KB] (1/1)
 Test case #7: AC [0.017s, 808.00 KB] (1/1)
 Test case #8: AC [0.022s, 768.00 KB] (1/1)
 Test case #9: AC [0.007s, 744.00 KB] (1/1)
 Test case #10: AC [0.007s, 740.00 KB] (1/1)
 Test case #11: AC [0.008s, 744.00 KB] (1/1)
 Test case #12: AC [0.009s, 816.00 KB] (1/1)
 Test case #13: AC [0.008s, 812.00 KB] (1/1)
 Test case #14: AC [0.009s, 876.00 KB] (1/1)
 Test case #15: AC [0.007s, 784.00 KB] (1/1)
 > Test case #16: WA [0.010s, 784.00 KB] (0/1)
 Test case #17: AC [0.007s, 820.00 KB] (1/1)
 Test case #18: AC [0.008s, 836.00 KB] (1/1)
 Test case #19: AC [0.008s, 768.00 KB] (1/1)
 Test case #20: AC [0.008s, 820.00 KB] (1/1)
 Test case #21: AC [0.010s, 832.00 KB] (1/1)
 Test case #22: AC [0.009s, 832.00 KB] (1/1)
 Test case #23: AC [0.012s, 768.00 KB] (1/1)
 Test case #24: AC [0.011s, 728.00 KB] (1/1)
 Test case #25: AC [0.017s, 768.00 KB] (1/1)
 Test case #26: AC [0.034s, 768.00 KB] (1/1)
 Test case #27: AC [0.009s, 820.00 KB] (1/1)

Resources: 0.291s, 876.00 KB

Final score: 26/27 (0.0/10 points)



Test case # 16

10 11 8 11 10

팰린드롬 게임 (Div.1)

“주어진 수열로 팰린드롬 게임을 했을 때
얻을 수 있는 가장 적은 점수”

{ 3 , 2 , 1 , 2 , 3 , 2 , 6 , 2 , 3 }

팰린드롬 게임 (Div.1)

- i) $\{3\}, \{2\}, \{1\}, \{2, 3, 2\}, \{6\}, \{2\}, \{3\}$ 로 나누었을 때 : 19점
- ii) $\{3\}, \{2, 1, 2\}, \{3, 2, 6, 2, 3\}$ 로 나누었을 때 : 8점
- iii) $\{3, 2, 1, 2, 3\}, \{2, 6, 2\}, \{3\}$ 로 나누었을 때 : 6점

팰린드롬 게임 (Div.1)

BOJ #10942 팰린드롬?

<https://www.acmicpc.net/problem/10942>

“S부터 E까지 팰린드롬?”

팰린드롬 게임 (Div.1)

BOJ #10942 팰린드롬?

$dp[i][j]$: i 부터 j 까지 팰린드롬인가? (True or False)

만약 수열 S 에서

$S[i] = S[j]$ 이고 $dp[i+1][j-1] = \text{True}$ 이면

$dp[i][j] = \text{True}$ 다.

팰린드롬 게임 (Div.1)

$dp[i][j]$: i 부터 j 까지 팰린드롬의 점수 (1 or 2 or 3 or 0)

만약 수열 S 에서 $S[i] = S[j]$ 일 때

i) $dp[i+1][j-1] = 1$ 이고 [증가] : $dp[i][j] = 1$

ii) $dp[i+1][j-1] = 2$ 이고 [감소] : $dp[i][j] = 2$

iii) 위 두가지 경우가 아니고 $dp[i+1][j-1] \leq 3$: $dp[i][j] = 3$

팰린드롬 게임 (Div.1)

$dp[i][j]$ 를 메모이제이션 $\rightarrow O(n^2)$

이제 i 부터 j 까지 팰린드롬 점수를 구하는데

$O(1)$ 이 걸린다.

```

// L = 1
for (int i = 0; i < N; i++) {
    cin >> P[i];
    fill(dp1[i], dp1[i] + N, 1e9);
    dp1[i][i] = 3;
}

// L = 2
for (int i = 0; i < N - 1; i++) {
    if (P[i] == P[i + 1])
        dp1[i][i + 1] = 3;
}

// L = 3, 4
for (int k = 2; k <= 3; k++) {
    for (int i = 0, j; i < N - k; i++) {
        j = i + k;
        if (P[i] == P[j] && dp1[i + 1][j - 1] == 3) {
            if (P[i] < P[i + 1]) dp1[i][j] = 1;
            else if (P[i] > P[i + 1]) dp1[i][j] = 2;
            else dp1[i][j] = 3;
        }
    }
}

```

```

// L > 4
for (int k = 4; k < N; k++) {
    for (int i = 0, j; i < N - k; i++) {
        j = i + k;
        if (P[i] == P[j]) {
            if (dp1[i + 1][j - 1] == 1 && P[i] < P[i + 1])
                dp1[i][j] = 1;
            else if (dp1[i + 1][j - 1] == 2 && P[i] > P[i + 1])
                dp1[i][j] = 2;
            else if (dp1[i + 1][j - 1] <= 3)
                dp1[i][j] = 3;
        }
    }
}

```

팰린드롬 게임 (Div.1)

이제 쉽게 풀 수 있다.

$dp2[i]$: 0부터 i 까지 팰린드롬 수열을 분할했을 때 최소 점수

$$dp2[i] = \min(dp2[j-1] + dp1[j][i])$$

-> $dp2[N-1]$ 까지 구하는데 $O(n^2)$

```
dp2[0] = 3;
for (int i = 1; i < N; i++) {
    dp2[i] = dp2[i - 1] + 3;
    for (int j = 0; j < i; j++) {
        if (P[i] == P[j]) {
            if (j == 0) dp2[i] = min(dp2[i], dp1[j][i]);
            else dp2[i] = min(dp2[i], dp2[j - 1] + dp1[j][i]);
        }
    }
}
cout << dp2[N - 1] << endl;
```

풀이가 비슷한 문제

BOJ #1509 팰린드롬 분할

<https://www.acmicpc.net/problem/1509>

BOJ #13330 유사 팰린드롬

<https://www.acmicpc.net/problem/13330>

그 외에도 제목에 ‘팰린드롬’이 들어간 문제가

백준에 정확히 50문제 있다.