

이분 탐색

최백준 choi@startlink.io

이분 탐색으로 정답 찾기

2

Binary Search

- 정답을 구할 수 없지만
- X가 가능한지 아닌지 알아내는 것은 가능한 문제들이 있다

이분 탐색으로 정답 찾기

3

Binary Search

- 정답을 구하는 문제
 - A에서 B까지 가는 가장 빠른 시간을 구하는 것
- 가능한지 살펴보는 문제
 - A에서 B까지 X라는 시간으로 이동할 수 있나?

이분 탐색으로 정답 찾기

Binary Search

- 정답을 구하는 문제는 가능여부를 판별하는 문제로 바꿀 수 있다.
- A에서 B까지 가는 가장 빠른 시간이 M인 경우에
 - M보다 빠른 시간은 모두 불가능
 - M보다 큰 시간은 모두 가능

이분 탐색으로 정답 찾기

Binary Search

- 가능한지 아닌지를 알아보는 문제도 정답을 구하는 문제로 바꿀 수 있다
- A에서 B까지 1라는 시간으로 이동할 수 있나?
- A에서 B까지 2라는 시간으로 이동할 수 있나?
- A에서 B까지 3라는 시간으로 이동할 수 있나?
- A에서 B까지 4라는 시간으로 이동할 수 있나?
- A에서 B까지 5라는 시간으로 이동할 수 있나?
- A에서 B까지 6라는 시간으로 이동할 수 있나?
- A에서 B까지 7라는 시간으로 이동할 수 있나?
- ...

이분 탐색으로 정답 찾기

Binary Search

- 가능한지 아닌지를 알아보는 문제도 정답을 구하는 문제로 바꿀 수 있다
- A에서 B까지 1라는 시간으로 이동할 수 있나? 아니오
- A에서 B까지 2라는 시간으로 이동할 수 있나? 아니오
- A에서 B까지 3라는 시간으로 이동할 수 있나? 아니오
- A에서 B까지 4라는 시간으로 이동할 수 있나? 아니오
- **A에서 B까지 5라는 시간으로 이동할 수 있나? 예**
- A에서 B까지 6라는 시간으로 이동할 수 있나? 예
- A에서 B까지 7라는 시간으로 이동할 수 있나? 예
- A에서 B로 이동하는 가장 빠른 시간은 5이다.

이분 탐색으로 정답 찾기

7

Binary Search

- 어떤 기준 X 를 가지고 Yes/No로 나누어지는 것만 정답을 찾을 수 있다.

이분 탐색으로 정답 찾기

Binary Search

- 이분 탐색을 이용하는 방법이기 때문에 다음을 모두 정해야 한다.
 1. 가능한 정답의 최솟값 (left)
 2. 가능한 정답의 최댓값 (right)
 3. 정답을 하나 결정했을 때, 이것이 문제의 조건에 맞는지 검사하는 방법 (go 함수)
 4. 조건에 맞는 경우 정답을 더 크게 해야 하는지 작게 해야 하는지 결정

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- 1부터 N까지 수를 이어서 쓰면 아래와같은 새로운 수를 만들 수 있다. ($1 \leq N \leq 100,000,000$)
- 1234567891011121314151617...
- 이 때, K번째 수가 무엇인지 찾는 문제 ($1 \leq K \leq 1,000,000,000$)

수 이어 쓰기 2

10

<https://www.acmicpc.net/problem/1790>

- 실제로 수를 만드는 것은 시간이 너무 오래 걸려서 불가능하다.

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- N번째 수의 길이는 자리수 별로 길이를 계산하는 방식으로 알 수 있다.
- $N = 120$
- $1 \sim 9 \rightarrow (9 - 1 + 1) \times 1$
- $10 \sim 99 \rightarrow (99 - 10 + 1) \times 2$
- $100 \sim 120 \rightarrow (120 - 100 + 1) \times 3$

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- 이 점을 이용해서 이분 탐색으로 N 을 결정하고
- 그 때마다 수의 길이를 재보고
- K 보다 작거나 같은지 비교해본다.

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- $N = 20$, $K = 23$ 인 경우
- 1부터 20까지 이어 붙인 수의 길이: 31
- K 가 길이보다 작기 때문에
- N 이 20보다 작아져도 등장한다는 것을 알 수 있다

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- $N = 20$, $K = 23$ 인 경우
- 현재 가능한 정답의 범위: 1~20
- 1부터 10까지 이어 붙인 수의 길이: 11
- K 가 길이보다 크기 때문에
- N 이 10보다 작아지면 절대 K 번째 수가 없다는 것을 알 수 있다
- 하지만, 10보다 커지면 등장할 수도 있다
- 따라서, 오른쪽을 택한다.

수 이어 쓰기 2

15

<https://www.acmicpc.net/problem/1790>

- $N = 20$, $K = 23$ 인 경우
- 현재 가능한 정답의 범위: 11~20
- 1부터 15까지 이어 붙인 수의 길이: 21
- K 가 길이보다 크기 때문에
- N 이 15보다 작아지면 절대 K 번째 수가 없다는 것을 알 수 있다
- 하지만, 15보다 커지면 등장할 수도 있다
- 따라서, 오른쪽을 택한다.

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- $N = 20$, $K = 23$ 인 경우
- 현재 가능한 정답의 범위: 16~20
- 1부터 18까지 이어 붙인 수의 길이: 27
- K 가 길이보다 작기 때문에
- N 이 18보다 커지면 항상 K 번째 수가 등장한다.
- 하지만, K 번째 수가 등장하는 순간 N 은 작아져야 하기 때문에
- 왼쪽을 택한다.

수 이어 쓰기 2

17

<https://www.acmicpc.net/problem/1790>

- $N = 20$, $K = 23$ 인 경우
- 현재 가능한 정답의 범위: 16~17
- 1부터 16까지 이어 붙인 수의 길이: 23
- K 가 이어 붙인 수의 길이와 같기 때문에, K 번째 수는 6이라는 사실을 알 수 있다

수 이어 쓰기 2

18

<https://www.acmicpc.net/problem/1790>

- left = 1
- right = N

수 이어 쓰기 2

<https://www.acmicpc.net/problem/1790>

- 소스: <http://codeplus.codes/165d76d38e824cd1871b1c6fdbd90109>

랜선 자르기

20

<https://www.acmicpc.net/problem/1654>

- 랜선 K개가 있다
- 랜선을 같은 길이로 잘라서 N개 이상을 만들어야 한다
- 300cm를 140cm로 자르면, 140cm 2개, 남는 20cm는 버린다
- 자른 것을 붙일 수 없다

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 예를 들어, 랜선이 4개가 있고, 11개를 만들어야 하는 경우
- 802, 743, 457, 539 인 경우
- 200으로 자르면 11개를 만들 수 있다
- $802/200 = 4$
- $743/200 = 3$
- $457/200 = 2$
- $539/200 = 2$
- $4+3+2+2 = 11$

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 200으로 자르면 11개를 만들 수 있다는 것은
- 199으로 자르면 11개 이상을
- 198로 자르면 11개 이상을
- ...
- 1로 자르면 11개 이상을
- 만들 수 있다는 것을 의미한다.
- 따라서, 200이하의 값으로는 N개 이상이 나오는지 구해볼 필요가 없다.

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 만약, 길이 X 로 잘랐을 때, N 개 이상을 만들 수 있으면, X 를 크게 만들어 봐야 한다
- N 개 이상을 만들 수 없다면 X 를 작게 만들어야 한다

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 예를 들어, 랜선이 4개가 있고, 11개를 만들어야 하는 경우
- 802, 743, 457, 539 인 경우
- 201로 자르는 경우
- $802/201 = 3$
- $743/201 = 3$
- $457/201 = 2$
- $539/201 = 2$
- $3+3+2+2 = 10$

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 예를 들어, 랜선이 4개가 있고, 11개를 만들어야 하는 경우
- 802, 743, 457, 539 인 경우
- 199로 자르는 경우
- $802/199 = 4$
- $743/199 = 3$
- $457/199 = 2$
- $539/199 = 2$
- $4+3+2+2 = 11$

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 예를 들어, 랜선이 4개가 있고, 11개를 만들어야 하는 경우
- 802, 743, 457, 539 인 경우
- 100으로 자르는 경우
- $802/199 = 8$
- $743/199 = 7$
- $457/199 = 4$
- $539/199 = 5$
- $8+7+4+5 = 24$

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- left = 1
 - right = 랜선 길이의 최댓값
1. X라는 길이로 랜선을 만들어본다.
 2. N개 이상 만들 수 있으면, X를 크게 바꾼다.
 3. 만들 수 없으면 X를 작게 바꾼다.

랜선 자르기

<https://www.acmicpc.net/problem/1654>

- 소스: <http://codeplus.codes/4afa49d3a24f46b5a5fd91dad69c509f>

나무 자르기

<https://www.acmicpc.net/problem/2805>

- 상근이는 나무 M 미터가 필요하다
- 절단기에 높이 H 를 지정해야 한다
- 높이를 지정하면 톱날이 땅으로부터 H 미터 위로 올라간다
- 한 줄에 연속해있는 나무를 모두 절단해버린다
- 높이가 H 보다 큰 나무는 H 위의 부분이 잘릴 것이고, 낮은 나무는 잘리지 않을 것이다
- 적어도 M 미터의 나무를 집에 가져가기 위해서 절단기에 설정할 수 있는 높이의 최대값

나무 자르기

30

<https://www.acmicpc.net/problem/2805>

- 예를 들어 나무가 20, 15, 10, 17인 경우
- 절단기의 높이를 15로 하면
- 15, 15, 10, 15 가 되어 버리고
- 길이가 5인 나무와 2인 나무를 집에 들고 갈 수 있다

나무 자르기

<https://www.acmicpc.net/problem/2805>

- $left = 0$
 - $right$ = 나무 높이의 최댓값
1. X 라는 높이로 잘라본다. 나온 길이의 합 = C
 2. C 가 M 보다 크거나 같으면 X 를 작게
 3. 아니면 X 를 크게

나무 자르기

<https://www.acmicpc.net/problem/2805>

- 소스: <http://codeplus.codes/c7bac0b5830f414abba5149324479cad>

공유기 설치

<https://www.acmicpc.net/problem/2110>

- 집 N 개, 위치 = x_1, \dots, x_N
- 공유기 C 개를 설치
- 가장 인접한 두 공유기 사이의 거리를 가능한 크게
- C 개의 공유기를 N 개의 집에 적당히 설치해서, 가장 인접한 두 공유기 사이의 거리를 최대
- $2 \leq C \leq N \leq 200,000, 1 \leq x_i \leq 1,000,000,000$

공유기 설치

<https://www.acmicpc.net/problem/2110>

- 집 N 개, 위치 = x_1, \dots, x_N
- 공유기 C 개를 설치
- 가장 인접한 두 공유기 사이의 거리를 가능한 크게
- C 개의 공유기를 N 개의 집에 적당히 설치해서, **가장 인접한 두 공유기 사이의 거리를 최대**
- $2 \leq C \leq N \leq 200,000, 1 \leq x_i \leq 1,000,000,000$

공유기 설치

<https://www.acmicpc.net/problem/2110>

- 가장 인접한 두 공유기 사이의 거리를 k 로 결정
- 이 때, 공유기를 설치했을 때, C 개 이상 설치할 수 있으면 가능하다.
- 가능한 경우면 k 를 크게
- 불가능하면 작게한다.

공유기 설치

<https://www.acmicpc.net/problem/2110>

- 가장 인접한 두 공유기 사이의 거리를 k 로 결정했을 때, 공유기를 D 개 설치할 수 있고 $C \leq D$ 라면
- 공유기를 C 개 설치할 수 있는 것과 같다.
- 가장 인접한 두 공유기 사이의 거리가 k 가 되게 유지하면서 $D-C$ 개를 제거하면 되기 때문이다.
- 제거한 후에 가장 인접한 두 공유기 사이의 거리가 k 가 아닐 수도 있지만, 사실 이는 별 문제가 되지 않는다.
- 제거하는 모든 방법이 변한 경우라면 정답보다 큰 값이라는 것을 의미하기 때문이다.

공유기 설치

<https://www.acmicpc.net/problem/2110>

- 소스: <http://codeplus.codes/ccb587d5197a498daeafcf68fd0a13d3>

중량제한

<https://www.acmicpc.net/problem/1939>

- N개의 섬과 M개의 다리로 이루어진 나라가 있다.
- 각 다리에는 중량제한이 있고, 중량제한을 초과하면 다리가 무너지게 된다
- 한 번의 이동에서 옮길 수 있는 물품들의 중량의 최대값을 구하는 문제

중량제한

<https://www.acmicpc.net/problem/1939>

- 무엇을 결정해야 할까?

중량제한

40

<https://www.acmicpc.net/problem/1939>

- 최대 무게를 결정하고, 그 무게로 갈 수 있는지를 검사해본다
- 검사는 어떤 알고리즘을 이용하면 될까?

중량제한

<https://www.acmicpc.net/problem/1939>

- 최대 무게를 결정하고, 그 무게로 갈 수 있는지를 검사해본다
- 검사는 어떤 알고리즘을 이용하면 될까? DFS나 BFS를 이용하면 된다.

중량제한

42

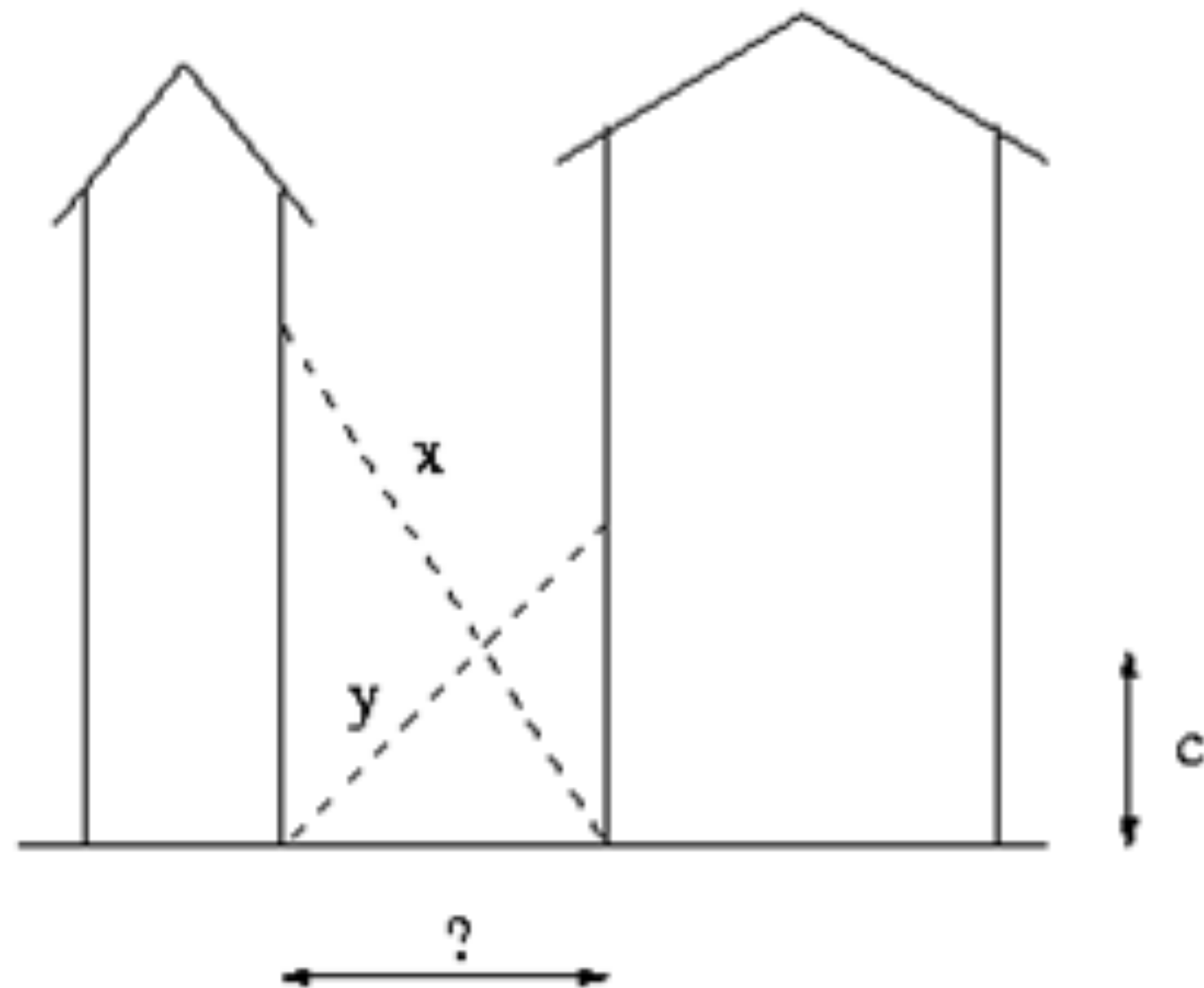
<https://www.acmicpc.net/problem/1939>

- 소스: <http://codeplus.codes/0befebde85444facab51a9eba0e8693c>

사다리

<https://www.acmicpc.net/problem/2022>

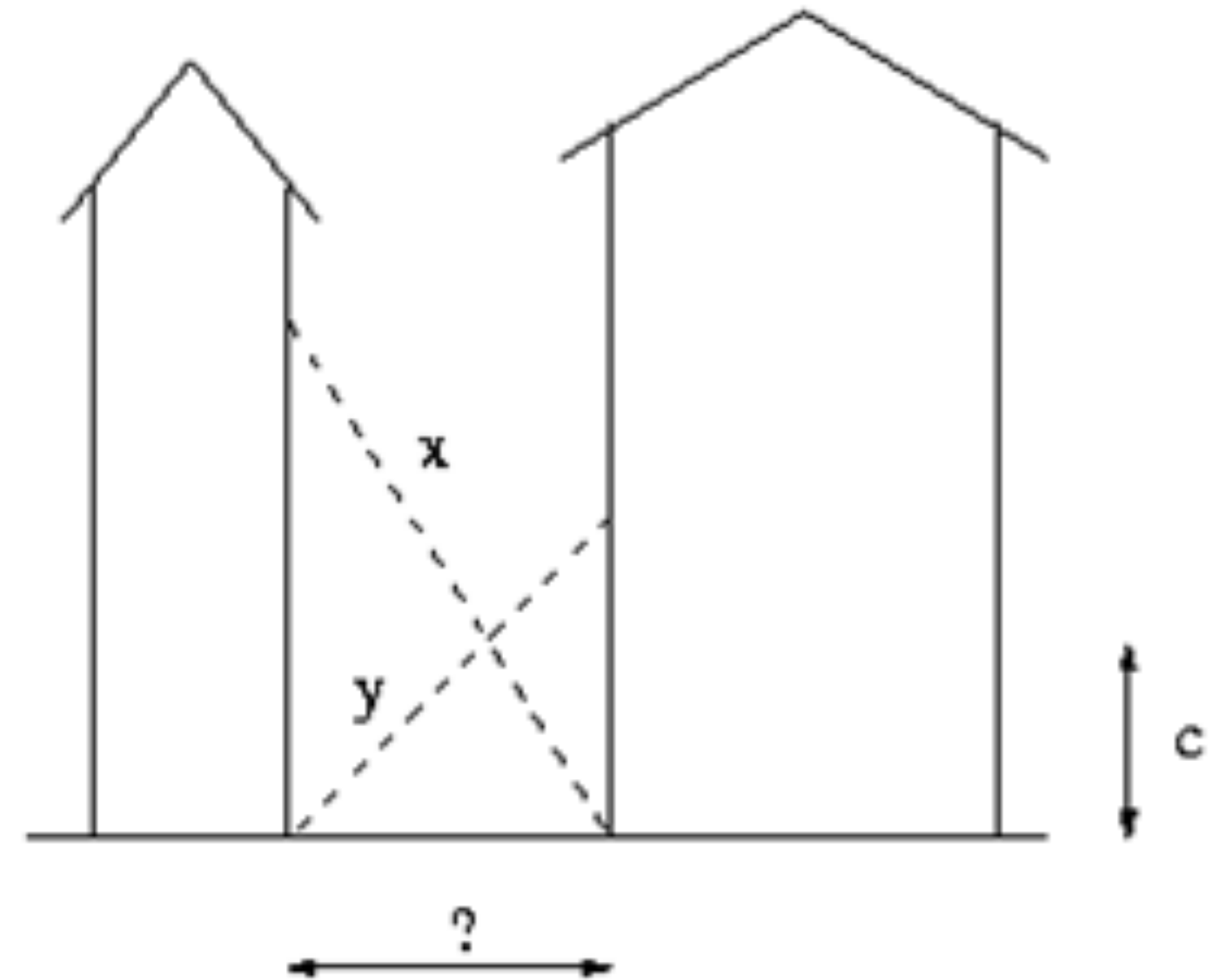
- x , y , c 가 주어졌을 때, 두 빌딩 사이의 떨어진 거리를 구하는 문제



사다리

<https://www.acmicpc.net/problem/2022>

- $h1 = \sqrt{x^2 - d^2}$
- $h2 = \sqrt{y^2 - d^2}$

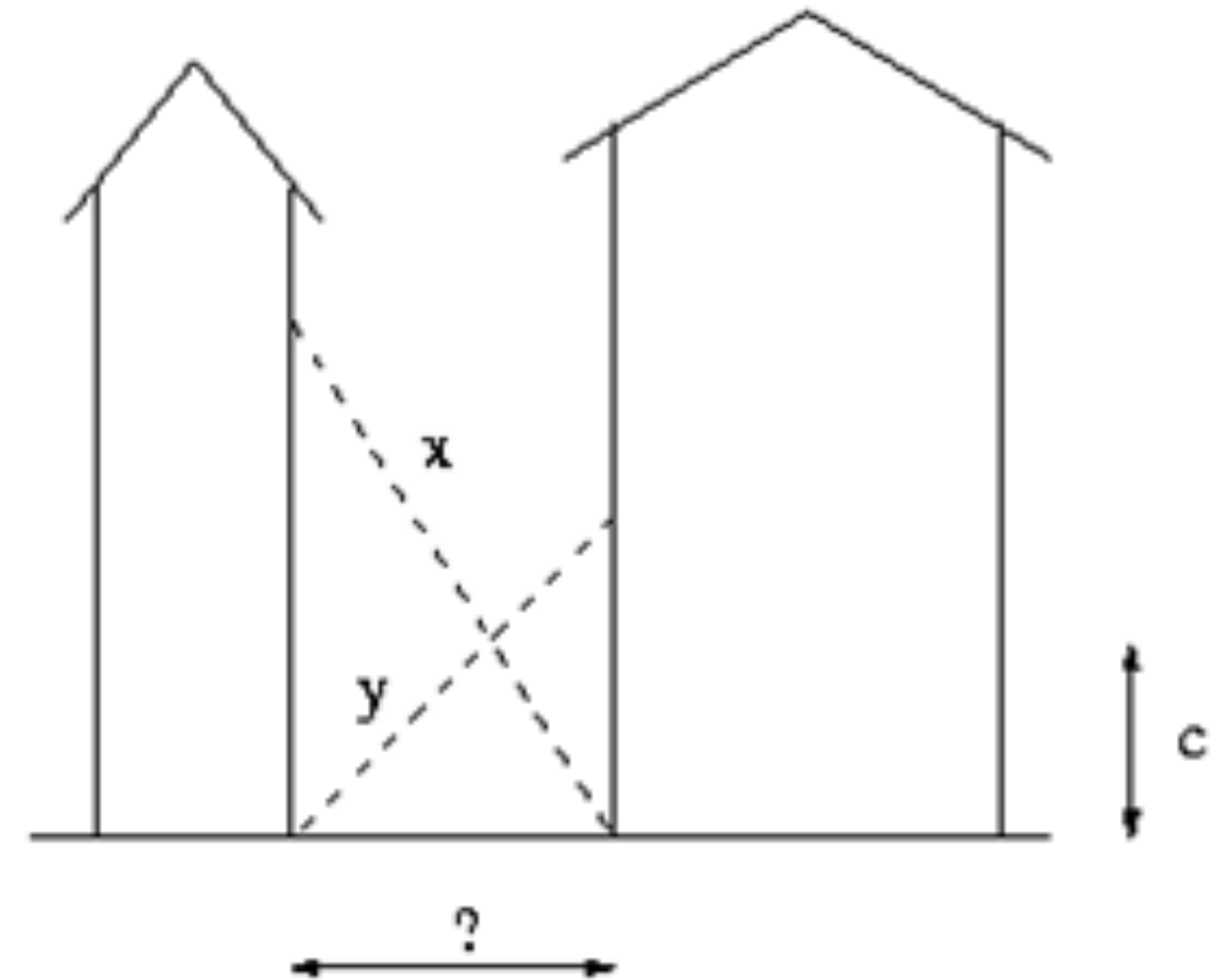


사다리

45

<https://www.acmicpc.net/problem/2022>

- 삼각형의 닮음을 이용하면
- $h_2/d = c/d_1$
- $h_1/d = c/d_2$
- $d = d_1 + d_2$
- $1/h_1 + 1/h_2 = 1/c$
- $c = h_1 \times h_2 / (h_1 + h_2)$



사다리

<https://www.acmicpc.net/problem/2022>

- 실수에서 이분 탐색을 수행하는 것이기 때문에, 변경해야 하는 부분이 있다.
- `while (left <= right)` 와 같은 표현의 사용이 불가능하다
- `left = mid + 1` 불가능 (`mid` 와 `mid + 1` 사이에 정답이 있을 수 있기 때문)
- `right = mid - 1` 불가능 (`mid - 1` 와 `mid` 사이에 정답이 있을 수 있기 때문)

사다리

<https://www.acmicpc.net/problem/2022>

- 실수에서 이분 탐색을 수행하는 것이기 때문에, 달라져야 하는 부분이 있다.
- `while (left <= right)` 와 같은 표현의 사용이 불가능하다
- `left = mid + 1` => `left = mid`
- `right = mid - 1` => `right = mid`

사다리

<https://www.acmicpc.net/problem/2022>

- `while (left <= right)` 와 같은 표현의 사용이 불가능하다
- 대신 아래와 같은 표현 중 하나를 사용한다
- `for (int k=0; k<100000; k++)`
- `while (abs(right-left) > 1e-6)`

사다리

<https://www.acmicpc.net/problem/2022>

- 소스: <http://codeplus.codes/8a805fc5bbcc485db171e33c207f32a9>
- 소스: <http://codeplus.codes/f1555a61aea14749ba097543bfc75709>

삼분 탐색

삼분 탐색

Ternary Search

51

- 최소값 또는 최대값이 하나인 함수 (Unimodal function)에서 최소/최대값을 찾는 방법
- 이분 탐색과 비슷하지만, 삼등분을 한다.

삼분 탐색

Ternary Search

- 최대값을 찾는 경우 아래와 같이 동작한다.

```
while (true) {  
    if (abs(right-left) < 1e-9) return (left+right)/2;  
    double m1 = left + (right-left)/3;  
    double m2 = right - (right-left)/3;  
    if (f(m1) < f(m2)) {  
        left = m1;  
    } else {  
        right = m2;  
    }  
}
```

선분과 점

<https://www.acmicpc.net/problem/11664>

- 3차원 좌표 평면 위에 선분 하나와 점 하나가 있을 때
- 선분과 점 사이의 거리의 최소값을 구하는 문제

선분과 점

<https://www.acmicpc.net/problem/11664>

- 선분과 점 사이의 거리는 가까워졌다가 멀어진다.
- 즉, 삼분 탐색을 이용해 최소값을 구하는 문제이다.

선분과 점

<https://www.acmicpc.net/problem/11664>

- 소스: <http://codeplus.codes/f91362ecfceb4b9daebf3f3c2fbce47c>