

Algorithm Study

04.28.2024 김홍주

#2110. 공유기 설치/Gold 4

Q. 공유기 설치

도현이의 집 N 개가 수직선 위에 있다. 각각의 집의 좌표는 x_1, \dots, x_N 이고, 집 여러개가 같은 좌표를 가지는 일은 없다.

집에 공유기 C 개를 설치하려고 한다. 최대한 많은 곳에서 와이파이를 사용하려고 하기 때문에, 한 집에는 공유기를 하나만 설치할 수 있고, 가장 인접한 두 공유기 사이의 거리를 가능한 크게 하여 설치하려고 한다.

C 개의 공유기를 N 개의 집에 적당히 설치해서, **가장 인접한 두 공유기 사이의 거리를 최대**로 하는 프로그램을 작성하시오.

<입력 조건 >

- 집의 개수 N ($2 \leq N \leq 200,000$)와 공유기 개수 C ($2 \leq C \leq N$)
- N 개의 집의 좌표 x_i ($0 \leq x_i \leq 1,000,000,000$)

<출력 조건>

- 가장 인접한 **두 공유기 사이의 최대 거리** 출력

[예제1.]

5 3

1

2

8

4

9

3

[힌트]

공유기를 1, 4, 8 또는 1, 4, 9에 설치하면
가장 인접한 두 공유기 사이의 거리는 3이고,
이 거리보다 크게 공유기를 3개 설치할 수 없다.

Hint

0. 문제 유형 : 이분탐색, 매개변수 탐색

문제 조건

(1) N개 집 중에 “C개의 이상의 공유기 설치” 하기

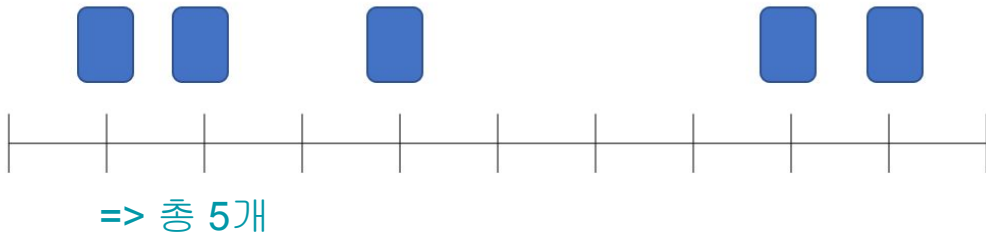
(2) 출력으로 설치된 공유기들 중 “최소 인접 간격” 중 최대값 반환

최소 인접 거리 범위 : 유한

=> “C 개 이상의 공유기” 설치가 가능한 “최소 인접거리 dist 거리”들 중 최대값

- 최소 인접 거리 **up** -> 설치 가능 공유기 c 개수 **감소**
- 최소 인접 거리 **down** -> 설치 가능 공유기 c 개수 **증가**

EX) 최소 인접 거리 : 1



```
5 3
1
2
8
4
9
```

Hint

0. 문제 유형 : 이분탐색, 매개변수 탐색

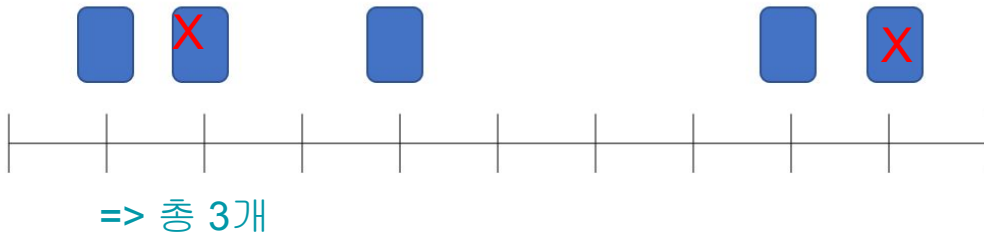
문제 조건

- (1) N개 집 중에 “C개의 이상의 공유기 설치” 하기
- (2) 출력으로 설치된 공유기들 중 “최소 인접 간격” 중 최대값 반환

=> “C 개 이상의 공유기” 설치가 가능한 “최소 인접거리 dist 거리”들 중 최대값

- 최소 인접 거리 **up** -> 설치 가능 공유기 c 개수 **감소**
- 최소 인접 거리 **down** -> 설치 가능 공유기 c 개수 **증가**

EX) 최소 인접 거리 : 2



```
5 3
1
2
8
4
9
```

Hint

0. 문제 유형 : 이분탐색, 매개변수 탐색

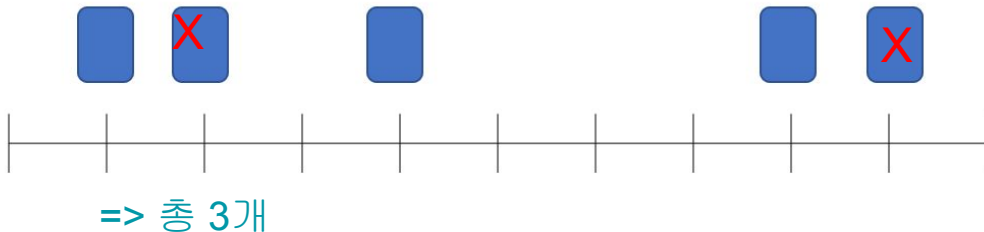
문제 조건

- (1) N개 집 중에 “C개의 이상의 공유기 설치” 하기
- (2) 출력으로 설치된 공유기들 중 “최소 인접 간격” 중 최대값 반환

=> “C 개 이상의 공유기” 설치가 가능한 “최소 인접거리 dist 거리”들 중 최대값

- 최소 인접 거리 **up** -> 설치 가능 공유기 c 개수 **감소**
- 최소 인접 거리 **down** -> 설치 가능 공유기 c 개수 **증가**

EX) 최소 인접 거리 : 3



```
5 3
1
2
8
4
9
```

Hint

0. 문제 유형 : 이분탐색, 매개변수 탐색

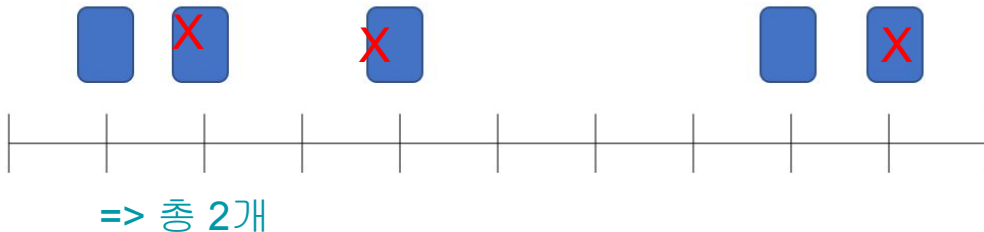
문제 조건

- (1) N개 집 중에 “C개의 이상의 공유기 설치” 하기
- (2) 출력으로 설치된 공유기들 중 “최소 인접 간격” 중 최대값 반환

=> “C 개 이상의 공유기” 설치가 가능한 “최소 인접거리 dist 거리”들 중 최대값

- 최소 인접 거리 **up** -> 설치 가능 공유기 c 개수 **감소**
- 최소 인접 거리 **down** -> 설치 가능 공유기 c 개수 **증가**

EX) 최소 인접 거리 : 4



5 3
1
2
8
4
9

Hint

0. 문제 유형 : 이분탐색, 매개변수 탐색

문제 조건

- (1) N개 집 중에 “C개의 이상의 공유기 설치” 하기
- (2) 출력으로 설치된 공유기들 중 “최소 인접 간격” 중 최대값 반환

=> “C 개 이상의 공유기” 설치가 가능한 “최소 인접거리 dist 거리”들 중 최대값

설치 여부 판단 => 그 중 최소 인접거리가 가장 큰 값 반환

- 최소 인접 거리 up -> 설치 가능 공유기 c 개수 감소
- 최소 인접 거리 down -> 설치 가능 공유기 c 개수 증가

[예제]

최소 인접 거리 : 1, 2, 3, 4, 5, 6, 7, 8

C 개 이상 설치 가능 여부: T, T, T, F, F, F, F, F



```
5 3
1
2
8
4
9
```

Hint

2. FLOW

- Goal: “공유기 C개 이상” 설치 가능한 “최소 인접 거리” 중 최대값 구하기
=> C개 이상의 공유기가 가능한 “최소 인접 거리” 후보군 찾기 -> 후보군들 중 최대값

매개 변수 탐색

(1) 매개 변수: 최소 인접 거리 dist

- 탐색 범위 초기화: [start : end]

(2) 결정 함수 정의

fn(dist) : dist 이상의 간격으로 C개 이상의 공유기 설치 가능 여부 T/F

(3) 매개 변수 탐색

[1] 모든 집 N 을 순회하면서 최소 인접 간격 이상으로 설치 가능한 공유기 개수 계산=> O(N)

[2] 최소 인접 거리 dist 탐색 # 이진 탐색 O(long)

해당 최소 인접 거리에서 공유기 개수가 C 개 이상 -> 후보군 + 탐색 범위를 인접거리(mid)보다 큰 값으로 이동 [start -> mid+1]

„미만 -> 탐색 범위를 적은 값(left)으로 이동 : [end -> mid -1]

[예제]

최소 인접 거리 : 1, 2, 3, 4, 5, 6, 7, 8

C 개 이상 설치 가능 여부: T, T, T, F, F, F, F, F

이분탐색 VS 매개변수 탐색

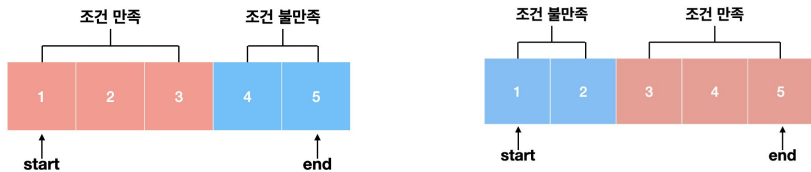
이분 탐색

- 중간값 Mid 과 Target값의 대소 비교로 탐색 범위를 절반씩 줄여가면서 탐색

매개변수 탐색

- 최적해 문제를 결정 문제로 변환

- 유한한 탐색 범위내에서 <조건>을 만족하는 매개변수의 최대/최소값 구하기 위해 사용하는 탐색 방법
 - 이분탐색의 Mid 값이 바로 정답은 아니지만 ,점차 정답에 가까워 가는 방식으로 문제 해결(정답여부 판단 가능)
- [적용 문제]
1. 결정 문제
 2. 어떤 시점까지는 조건을 만족하지만 , 그 시점 이후로 만족하지 않는 경우에서 최대값 구하기 (Upper Bound)
 3. ,, 조건 불만족하다 , 그 시점 이후로 만족하는 경우 최소값 구하기(Lower Bound)



[동작 과정]

1. **매개 변수 설정 & 이분 탐색 범위 초기화**
: 문제에서 최종적으로 구하는 최소/최대값의 변수 = 매개변수로 설정
2. **결정 함수 구현 & 결과 배열 연속 확인**
결정 함수 $fn(\#param)$: 매개변수를 입력으로 <조건>에 따라 T/F 반환
3. **최대값구할때, 결정 함수 결과가 $[t,t,t,...t,f,f]$ 로 $t \rightarrow f$ 로 바뀌는 부분 찾기**
 - 이분 탐색 : $mid = Target \rightarrow$ 탐색 종료
 - 매개변수 탐색 : $fn(mid) = T \rightarrow$ 정답 후보군 등재 + 탐색

Solution

```
import sys
input = sys.stdin.readline

#1. 입력 변수 + 오름차순 정렬
N, C = map(int, input().split())
arr = sorted([int(input()) for _ in range(N)])

#2. "최소 거리 dist"의 탐색 범위 초기화 # 이전 탐색
start = 1 ; end = arr[-1] - arr[0]
answer = 0 # 최소 거리 중 최대 거리

#3. 최소 인접 거리 dist 최대값 찾기
#설치 공유기는 dist이상의 간격으로 설치
while start <= end :
    mid = (start+end)//2
    cnt = 1 # arr[0]은 설치 시작점

    #[1] 최소 인접 거리가 mid 이상으로 "최대 설치 가능 공유기 개수 계산"
    prev = arr[0] # 이전에 설치한 공유기 위치
    for i in range(1,N) :
        if arr[i] - prev >= mid : # 공유기 등록
            prev = arr[i]
            cnt+=1

    #[2] 최소거리 dist 이분 탐색
    # 조건 : 설치한 공유기 개수 cnt 가 c 이상
    # True 경우 , 최소거리 Answer 업데이트, 탐색 범위 (mid+1 : end) 이동
    if cnt >= C :
        answer = max(answer, mid)
        start = mid+1
    # False 경우: 탐색 범위 Lower bound로 이동(start : mid)
    else :
        end = mid - 1

print(answer)
```

1. 오름차순 정렬

2. “최소 인접 거리”의 이분 탐색 범위 초기화 (start, end)

3. 최소인접거리의 최대값 찾기

<조건>: 최대한 설치 가능 공유기개수가 C 이상일 것

- 최소 인접 거리 가정 : mid

[1] 설치 거리가 mid 이상으로 설치 가능한 공유기 개수 계산

- 첫번째 원소 : 무조건 설치
- 현 집 위치와 이전 설치 공유기 간의 거리가 mid 이상이면 설치 가능
=> 다음 집과 비교

[2] <조건> 충족여부에 따른 이분 탐색 범위 업데이트

- 설치 공유기 개수가 C개 이상
-> 최소 인접 거리 (mid)는 정답 후보군 & 탐색 범위 [mid +1: end]
- 아니면,
-> 탐색 범위 [start : mid]

다른 Solution

공유기 개수 **C** 를 고려한 탐색 범위 초기화

```
1 from sys import stdin
2 n, c = map(int, input().split())
3 x = [int(y) for y in stdin.read().splitlines()]
4 x.sort()
5 def f(d):
6     count, val = 1, x[0]+d
7     for h in x:
8         if h>=val:
9             count, val = count+1, h+d
10    return count
11 B, E = 1, (x[-1]-x[0])//(c-1)+1
12 while E>B+1:
13     M = (B+E)//2
14     B, E = (M, E) if f(M)>=c else (B, M)
15 print(B)
```

[복사](#)[다운로드](#)

제출 번호	아이디	문제	문제 제목	결과	메모리	시간	언어	코드 길이	제출한 시간
14781601	fler9617	2110	공유기 설치	맞았습니다!!	54264 KB	104 ms	Python 3	350 B	5년 전

Assignment

[백준#1654.랜선자르기] 실버2

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

#이분탐색 , #매개변수탐색

참고자료 : <https://wikidocs.net/206315>

랜선 자르기



2 실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	267532	65153	43981	21.932%

문제