

[Problem Analysis]

n개의 electircity poles 가 있다. ($2 \leq n \leq 100,000$)

n개 중에 k개의 poles를 선택하여, amplifier를 설치하려한다.

이 때 설치된 k개의 pole들에 있어, 두 Pole 사이의 최소 거리가 최대한 커지도록 해야한다. 이 때의 거리를 구하는 것이 목표.

C6 이신원 이원
빈

[Foaming Ideas]

- 현재 pole들의 위치를 저장해둔 배열을 **pole[i]** ($1 \leq i \leq n$)이라 할 때, 인접한 각 pole 사이의 가능한 거리의 범위를 다음과 같이 설정할 수 있다.
min(pole[i]-pole[i-1])를 low, (pole[n] - pole[1])를 high라고 하면 $\Rightarrow low \leq dist \leq high$. (range)
- 위 상황을 기반으로 k개의 pole을 배치하는 것에 있어서, 어떤 경우에 최대가 되는 최소 거리가 생기는지 모른다. 하지만 이를 위해 모든 경우를 확인하는 것은 inefficient 하므로, 우선 인접한 각 pole 사이의 가능한 거리들 중 중간값(**mid = low + high / 2**)을 최대의 minimum distance라고 가정한다. **low ≤ mid ≤ high**
- 이후 각 pole 사이의 거리가 적어도 mid값을 충족하면서 k개의 pole들을 배치할 수 있는지 확인한다(valid). 배치할 수 있다면 적어도 해당 mid값이 최대의 minimum distance가 되고, 해당 mid보다 더 큰 값으로 k개의 pole들을 배치할 수 있는 가능성이 있으므로 range를 mid보다 크게끔 좁혀 valid한지 확인한다. **mid + 1 ≤ new_mid ≤ high**. 배치할 수 없다면 최대의 최소 거리가 해당 mid값보다 작으므로, range를 mid보다 작게끔 좁혀 valid한지 확인한다. **low ≤ new_mid ≤ mid - 1**.
- 이것을 low가 high보다 크거나 같을 때까지 반복하면 결국 k개의 pole을 배치할 수 있는 최대의 최소 거리를 구할 수 있다.

[Solve]

- n과 k, 그리고 n개 pole들의 위치값을 입력 받아 1차원 배열 **pole[i]** ($1 \leq i \leq n$)를 선언한다.
- 배열 pole을 위치 기준으로 오름차순 정렬한다.
- 설치된 k개의 pole들 중, 인접한 pole들 사이의 최소거리의 최댓값을 구한다.
 - 사용할 값 초기화
low = min(pole[i]-pole[i-1]) / high = (pole[n] - pole[1])
 - **low ≤ high**를 만족하는 동안 아래 동작을 반복하며 low,high, mid 값의 범위를 줄여나간다.
 1. mid에는 low와 high 값의 중간 값을 넣어준다. (**mid = low + high / 2**) mid 값을 최대의 minimum distance 라고 가정한다.
 2. pole들의 가장 앞 지점(**poles[1]**)부터 mid 이상 거리에서 가장 가까운 지점을 찾는다. 찾은 지점에서 다시 mid 이상 거리에서 가장 가까운 지점을 찾아 나간다.
 - poles[1]부터 poles[n]까지 찾은 지점이 k 이상이라면, 해당 mid값으로 k개의 pole들을 배치할 수 있다는 것을 의미한다.
 - k개의 pole들을 배치할 수 있는 경우, **더 큰 mid 값이 있을 가능성이 있기에 low = mid+1로 설정하여 1번으로 돌아간다.** **mid + 1 ≤ new_mid ≤ high**.
 - k개의 pole들을 배치할 수 없는 경우, **만족하는 mid값을 다시 찾기 위해 high = mid-1로 설정하여 1번으로 돌아간다.** **low ≤ new_mid ≤ mid-1**
- 최종적으로 정해진 mid값이 곧 최대의 최소거리이다.

[Time complexity]

1) 주어진 **N개의 pole**들을 위치 기준으로 오름차순 정렬 $\Rightarrow O(N \log N)$

2) **range**에서 초기의 가장 큰 값 **high**를 **M**이라 할 때, mid를 기준으로 범위를 반씩 줄여나가므로 **O(logM)**

3) 2번의 과정에서 각 mid에 대해, 총 **N개의 pole**들에 있어 각 pole 사이의 거리를 적어도 mid값을 충족하면서 k개 만큼 배치할 수 있는지 체크 $\Rightarrow O(N)$

($2 \leq N \leq 100,000$)이고, $1 \leq x_i$ (각 pole의 위치) $\leq 1,000,000,000$ 이므로 최악의 경우 **N < M**이고, 따라서 Time Complexity는 **O(NlogM)**이다.