이진탐색, 이분탐색(binary search)

binary search란?

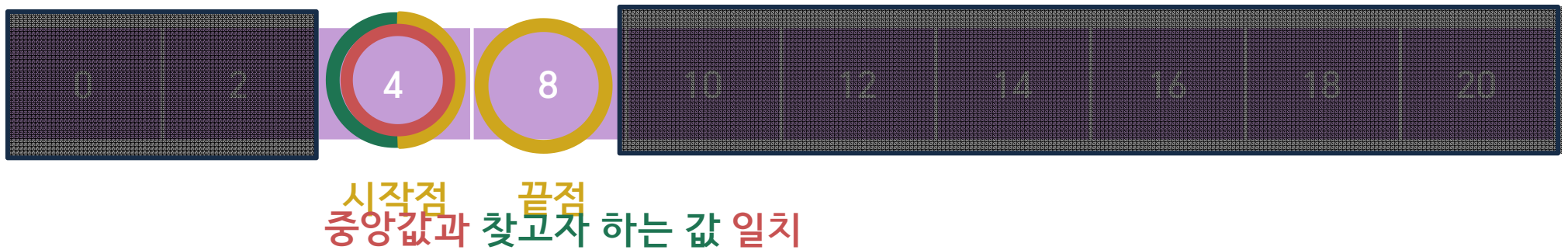
찾고자하는 값을 중앙값을 기준으로 찾는 알고리즘

binary search를 하려면 데이터가 순차정렬이 되어있어야한다.

binary search란?

찾고자하는 값을 중앙값을 기준으로 찾는 알고리즘

binary search를 하려면 데이터가 순차정렬이 되어있어야한다.



전체 데이터 수 : 10개 탐색 횟수 : 3회

이진탐색을 1번 할때마다 확인해야하는 원소의 갯수가 약 절반씩 줄어들어 연산 횟수는 $\log_2 N$에 비례함.

시간 복잡도 : O(logN)

원소 m의 위치 찾기

```python
import sys

n, m = map(int, sys.stdin.readline().split(" "))
data = list(map(int, sys.stdin.readline().split(" ")))

def binary_search(array, target):
    start, end = 0, n-1

    while start <= end:
        mid = (start + end) // 2

        if array[mid] == target :
            return mid
        elif array[mid] < target:
            start = mid + 1
        else:  # array[mid] > m :
            end = mid - 1

    return None

result = binary_search(data, m)
if result == None:
    print("탐색 실패")
else:
    print(result + 1)
```

입력

탐색범위 설정
(시작점 : 0, 끝점:데이터의 마지막 인덱스)

찾은경우 중간점 인덱스 반환

중간값보다 target이 작은 경우 왼쪽 탐색

중간값보다 target이 큰 경우 오른쪽 탐색

결과 출력

입력 예제

n, m	10	7								
data	1	3	5	7	9	11	13	15	17	19
	0	1	2	3	4	5	6	7	8	9

과정

탐색 1회 : mid = (0 +9) // 2 = 4
array[4] = 9, 9 > 7 이므로 end = 3

탐색 2회 : mid = (0 + 2) // 2 = 1
array[1] = 3, 3 < 7 이므로 start = 2

탐색 3회 : mid = (2 + 3) // 2 = 2
array[2] = 5, 5<7 이므로 start = 3

탐색 4회 : mid = 3 (target과 일치)

결과

4

백준 2805번

나무 자르기 성공 다국어 출처

☆ 한국어 ▾

실버 III

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	53768	15013	10003	25.033%

문제

상근이는 나무 M미터가 필요하다. 근처에 나무를 구입할 곳이 모두 망해버렸기 때문에, 정부에 벌목 허가를 요청했다. 정부는 상근이네 집 근처의 나무 한 줄에 대한 벌목 허가를 내주었고, 상근이는 새로 구입한 목재절단기를 이용해서 나무를 구할것이다.

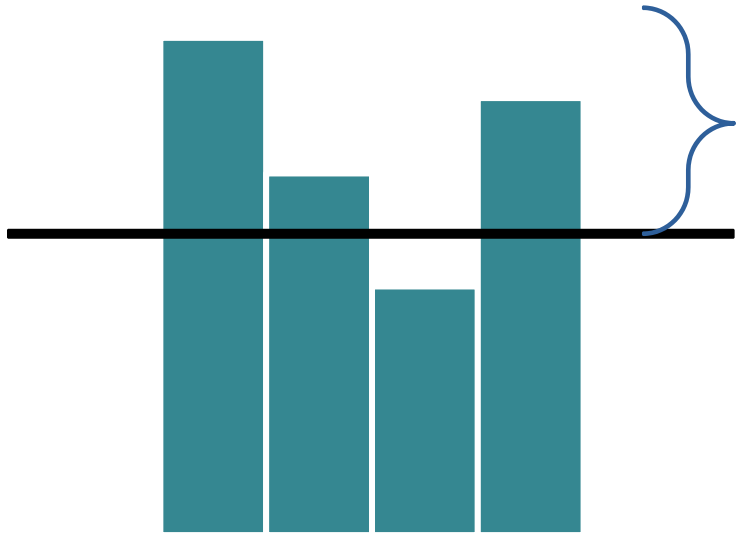
목재절단기는 다음과 같이 동작한다. 먼저, 상근이는 절단기에 높이 H를 지정해야 한다. 높이를 지정하면 톱날이 땅으로부터 H미터 위로 올라간다. 그 다음, 한 줄에 연속해있는 나무를 모두 절단해버린다. 따라서, 높이가 H보다 큰 나무는 H 위의 부분이 잘릴 것이고, 낮은 나무는 잘리지 않을 것이다. 예를 들어, 한 줄에 연속해있는 나무의 높이가 20, 15, 10, 17이라고 하자. 상근이가 높이를 15로 지정했다면, 나무를 자른 뒤의 높이는 15, 15, 10, 15가 될 것이고, 상근이는 길이가 5인 나무와 2인 나무를 들고 집에 갈 것이다. (총 7미터를 집에 들고 간다) 절단기에 설정할 수 있는 높이는 양의 정수 또는 0이다.

상근이는 환경에 매우 관심이 많기 때문에, 나무를 필요한 만큼만 집으로 가져가려고 한다. 이때, 적어도 M미터의 나무를 집에 가져가기 위해서 절단기에 설정할 수 있는 높이의 최댓값을 구하는 프로그램을 작성하시오.

백준 2805번

입력 예제

n, m	4	7		
tree_array	20	15	10	17



탐색데이터의 범위 : 1~가장 큰 나무의 길이

*tree_array가 탐색 범위가 아님

- 자른 나무의 총합(sum)이 상근이가 필요한 M 보다 크거나 같을 경우 :
  나무를 최대한 작게 잘라야 하기 때문에 시작점을 mid+1로 재탐색

- 자른 나무의 총합(sum)이 상근이가 필요한 M 보다 작을 경우 :
  자른값이 M보다 작은경우는 너무 크게 잘랐을 경우 이므로 더 작게 자르기 위해 끝점을 m-1 로 재탐색

백준 2805번

```python
import sys

n, m = map(int, sys.stdin.readline().split())
tree_array = list(map(int, sys.stdin.readline().split()))

def solution(target, tree_array):
    start, end = 1, max(tree_array)
    cut = 0
    while start <= end:
        mid = (start + end) // 2
        sum = 0

        for tree in tree_array:
            if tree >= mid:
                sum += tree - mid

        if sum >= target:
            cut = mid
            start = mid + 1
        else:
            end = mid - 1

    return cut

print(solution(m, tree_array))
```

입력

탐색범위 설정(시작점 : 1, 끝점:가장 긴 나무길이)

현재 높이(mid)에서 자를 경우 얼마만큼의 나무를 가져갈 수 있을지 계산

필요한 만큼 혹은 그 이상 잘라낸 경우-> 더 높이 자르도록 시작점 갱신

너무 적게 잘라낸 경우-> 더 낮게 자르도록 끝점 갱신

백준 1561번

입력 예제

n, m	22	5			
time_array	1	2	3	4	5

출력

4

놀이 공원 

 골드 II

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	5826	1277	893	21.846%

문제

N명의 아이들이 한 줄로 줄을 서서 놀이공원에서 1인승 놀이기구를 기다리고 있다. 이 놀이공원에는 총 M종류의 1인승 놀이기구가 있으며, 1번부터 M번까지 번호가 매겨져 있다.

모든 놀이기구는 각각 운행 시간이 정해져 있어서, 운행 시간이 지나면 탑승하고 있던 아이는 내리게 된다. 놀이 기구가 비어 있으면 현재 줄에서 가장 앞에 서 있는 아이가 빈 놀이기구에 탑승한다. 만일 여러 개의 놀이기구가 동시에 비어 있으면, 더 작은 번호가 적혀 있는 놀이기구를 먼저 탑승한다고 한다.

놀이기구가 모두 비어 있는 상태에서 신 โบ네 아이가 놀이기구에 탑승한다고 할 때, 줄의 마지막 아이가 타게 되는 놀이기구의 번호를 구하는 프로그램을 작성하시오.

백준 1561번

입력 예제

n, m	22	5			
time_array	1	2	3	4	5

놀이기구 번호

소요 시간	1	2	3	4	5	탑승인원(명)
0	o	o	o	o	o	5
1	o					6
2	o	o				8
3	o		o			10
4	o	o		o		13
5	o				o	15
6	o	o	o			18
7	o					19
8	o	o		o		22

백준 1561번

입력 예제

n, m	22	5			
time_array	1	2	3	4	5

탐색데이터의 범위 : 0 ~ n * 30

- 이분 탐색을 통해 아이들을 모두 태울 수 있는 시간찾기가 핵심
- 마지막 아이가 타기 전 시간까지 몇 명의 아이를 태울 수 있는지 탐색
- 위의 조건을 만족하는 아이들의 합과 마지막 아이가 타는 시간에 같이 앞의 놀이기구를 타는 아이들의 합을 구하며 제일 마지막에 탑승(N)한 아이 찾기
- 마지막 아이가 탈 때의 놀이기구의 번호를 출력

백준 1561번

```python
import sys

n, m = map(int, sys.stdin.readline().split())
time_array = list(map(int, sys.stdin.readline().split()))

def binary_search(array, target):
    start, end = 0, n * 30
    time = 0
    while start <= end:
        mid = (start + end) // 2
        sum = m
        for t in array:
            sum += mid // t

        if sum >= target:
            time = mid
            end = mid - 1
        else:
            start = mid + 1

    return time

last_time = binary_search(time_array, n)

sum_2 = m
for time in time_array:
    sum_2 += (last_time-1) // time

for i in range(m):
    if last_time % time_array[i] == 0:
        sum_2 += 1
    if sum_2 == n:
        print(i + 1)
        break
```

탐색범위 설정
(시작점 : 0,
끝점:n * 놀이기구 대기시간 최대 값)

* sum = m인 이유 : 0분에는 놀이기구의 수만큼 바로 탑승하기 때문

mid 기준으로 mid(분)까지 몇 명이 탔는지 계산

아이들의 합(sum)이 너무 적으면 시작점 갱신

아이들의 합(sum)이 너무 많으면 끝점 갱신

각 놀이기구별로 7분까지 몇명이 탔는지 계산하며 총 기구에 탑승한 아이들의 합을 구함

8분째에 각 놀이기구에 탈 수 있는 인원을 체크하여 인원 증가
* 8분에는 1번,2번,4번이 탑승 가능하다

아이들의 수(sum_2)가 n이 되는 순간, 해당 놀이기구 번호 반환

감사합니다