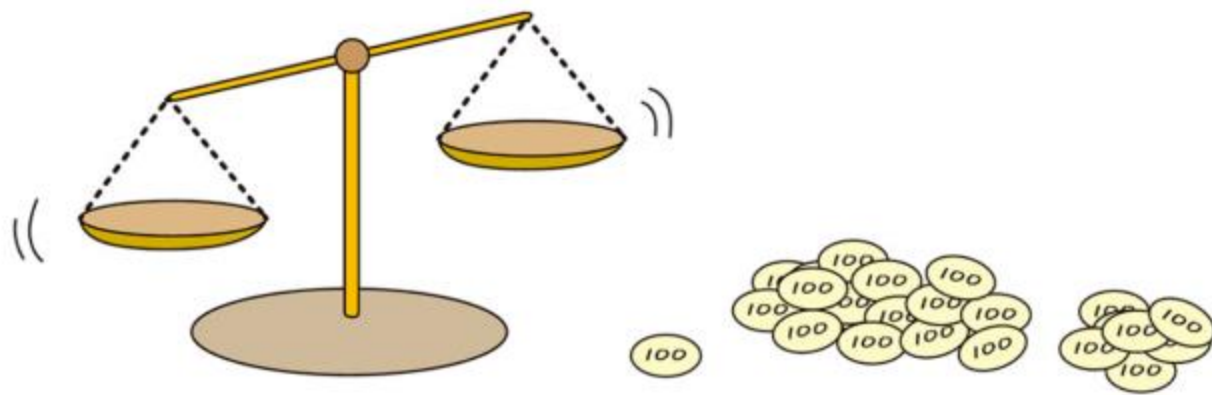
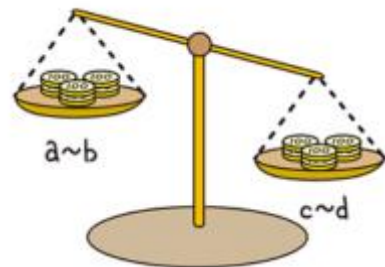


모두의 알고리즘 with 파이썬

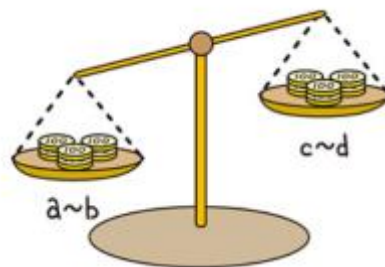
문제 17 가짜 동전 찾기 알고리즘

겉보기에는 똑같은 동전이 n 개 있습니다. 이 중에서 한 개는 싸고 가벼운 재료로 만들어진 '가짜 동전'입니다. 좌우 무게를 비교할 수 있는 양팔 저울을 이용해서 다른 동전보다 가벼운 가짜 동전을 찾아내는 알고리즘을 만들어 보세요.

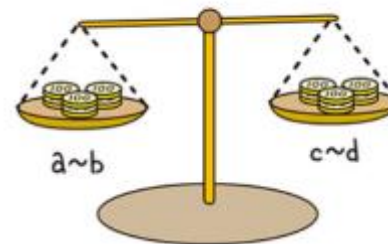




return -1
a~b에 가짜 동전이 있음



return 1
c~d에 가짜 동전이 있음



return 0
가짜 동전이 없음

```
def weigh(a, b, c, d):  
    fake = 29 # 가짜 동전의 위치(알고리즘은 weigh( ) 함수를 이용하여 이 값을 맞춰야 함)  
    if a <= fake and fake <= b:  
        return -1  
    if c <= fake and fake <= d:  
        return 1  
    return 0
```

방법 ①: 하나씩 비교하기

```
# weigh( ) 함수(저울질)를 이용하여
# left에서 right까지에 놓인 가짜 동전의 위치를 찾아냄
def find_fakecoin(left, right):
    for i in range(left + 1, right + 1): # left + 1부터 right까지 반복
        # 가장 왼쪽 동전과 나머지 동전을 차례로 비교
        result = weigh(left, left, i, i)
        if result == -1: # left 동전이 가벼움(left 동전이 가짜)
            return left
        elif result == 1: # i 동전이 가벼움(i 동전이 가짜)
            return i
        # 두 동전의 무게가 같으면 다음 동전으로

    # 모든 동전의 무게가 같으면 가짜 동전이 없는 예외 경우
    return -1

n = 100 # 전체 동전 개수
print(find_fakecoin(0, n - 1))
```

방법 ②: 반씩 그룹으로 나누어 비교하기

```
def find_fakecoin(left, right):  
    # 종료 조건: 가짜 동전이 있을 범위 안에 동전이 한 개뿐이면 그 동전이 가짜 동전임  
    if left == right:  
        return left  
  
    # left에서 right까지에 놓인 동전을 두 그룹(g1_left~g1_right, g2_left~g2_right)으로 나눔  
    # 동전 수가 홀수면 두 그룹으로 나누고 한 개가 남음  
    half = (right - left + 1) // 2  
    g1_left = left  
    g1_right = left + half - 1  
    g2_left = left + half  
    g2_right = g2_left + half - 1  
  
    # 나뉜 두 그룹을 weigh( ) 함수를 이용하여 저울질함  
    result = weigh(g1_left, g1_right, g2_left, g2_right)  
    if result == -1: # 그룹 1이 가벼움(가짜 동전이 이 그룹에 있음)  
        # 그룹 1 범위를 재귀 호출로 다시 조사  
        return find_fakecoin(g1_left, g1_right)  
    elif result == 1: # 그룹 2가 가벼움(가짜 동전이 이 그룹에 있음)  
        # 그룹 2 범위를 재귀 호출로 다시 조사  
        return find_fakecoin(g2_left, g2_right)  
    else: # 두 그룹의 무게가 같으면(나뉜 두 그룹 안에 가짜 동전이 없다면)  
        return right # 두 그룹으로 나뉘지 않고 남은 나머지 한 개의 동전이 가짜 동전임  
  
n = 100    # 전체 동전 개수  
print(find_fakecoin(0, n - 1))
```

계산복잡도

방법 ①: 하나씩 비교하기

N-1번으로 $O(n)$

방법 ②: 반씩 그룹으로 나누어 비교하기

절반씩 제외하며 $O(\log n)$