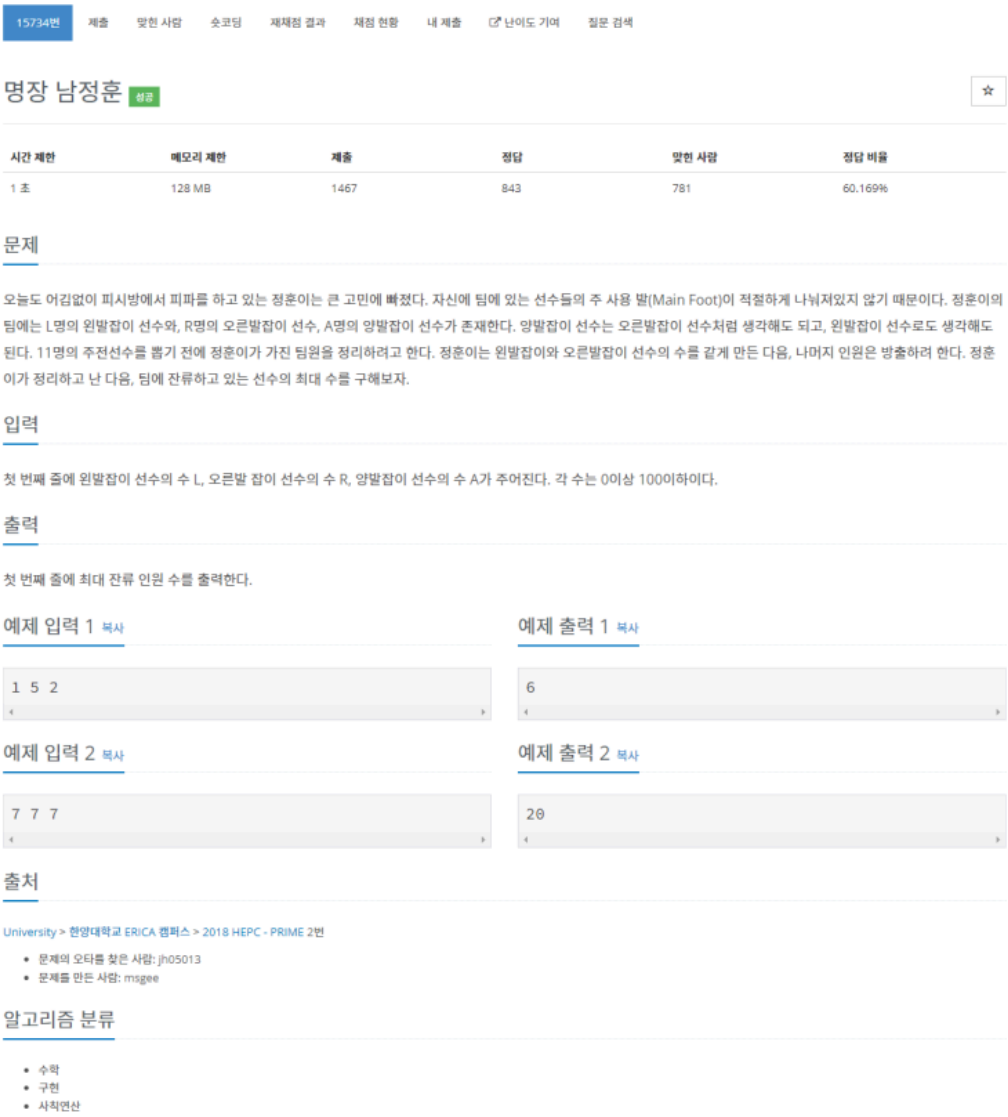
다양한 문제 풀이 방식을 보여주기에 좋은 문제 같아 가져와봤다.
https://www.acmicpc.net/problem/15734

15734번: 명장 남정훈
15734번 제출 맞힌 사람 숏코딩 재채점 결과 채점 현황 질문 검색 명장 …
www.acmicpc.net



15734번 제출 맞힌 사람 숏코딩 재채점 결과 채점 현황 내 제출 난이도 기여 질문 검색

명장 남정훈 성공

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	1467	843	781	60.169%

문제

오늘도 어김없이 피시방에서 피파를 하고 있는 정훈이는 큰 고민에 빠졌다. 자신에 팀에 있는 선수들의 주 사용 발(Main Foot)이 적절하게 나눠져있지 않기 때문이다. 정훈이의 팀에는 L명의 왼발잡이 선수와, R명의 오른발잡이 선수, A명의 양발잡이 선수가 존재한다. 양발잡이 선수는 오른발잡이 선수처럼 생각해도 되고, 왼발잡이 선수로도 생각해도 된다. 11명의 주전선수를 뽑기 전에 정훈이가 가진 팀원을 정리하려고 한다. 정훈이는 왼발잡이와 오른발잡이 선수의 수를 같게 만든 다음, 나머지 인원은 방출하려 한다. 정훈이가 정리하고 난 다음, 팀에 잔류하고 있는 선수의 최대 수를 구해보자.

입력

첫 번째 줄에 왼발잡이 선수의 수 L, 오른발 잡이 선수의 수 R, 양발잡이 선수의 수 A가 주어진다. 각 수는 0이상 100이하이다.

출력

첫 번째 줄에 최대 잔류 인원 수를 출력한다.

예제 입력 1 복사

```
1 5 2
```

예제 출력 1 복사

```
6
```

예제 입력 2 복사

```
7 7 7
```

예제 출력 2 복사

```
20
```

출처

University > 한양대학교 ERICA 캠퍼스 > 2018 HEPC - PRIME 2번

- 문제의 오타를 찾은 사람: jh05013
- 문제를 만든 사람: msgee

알고리즘 분류

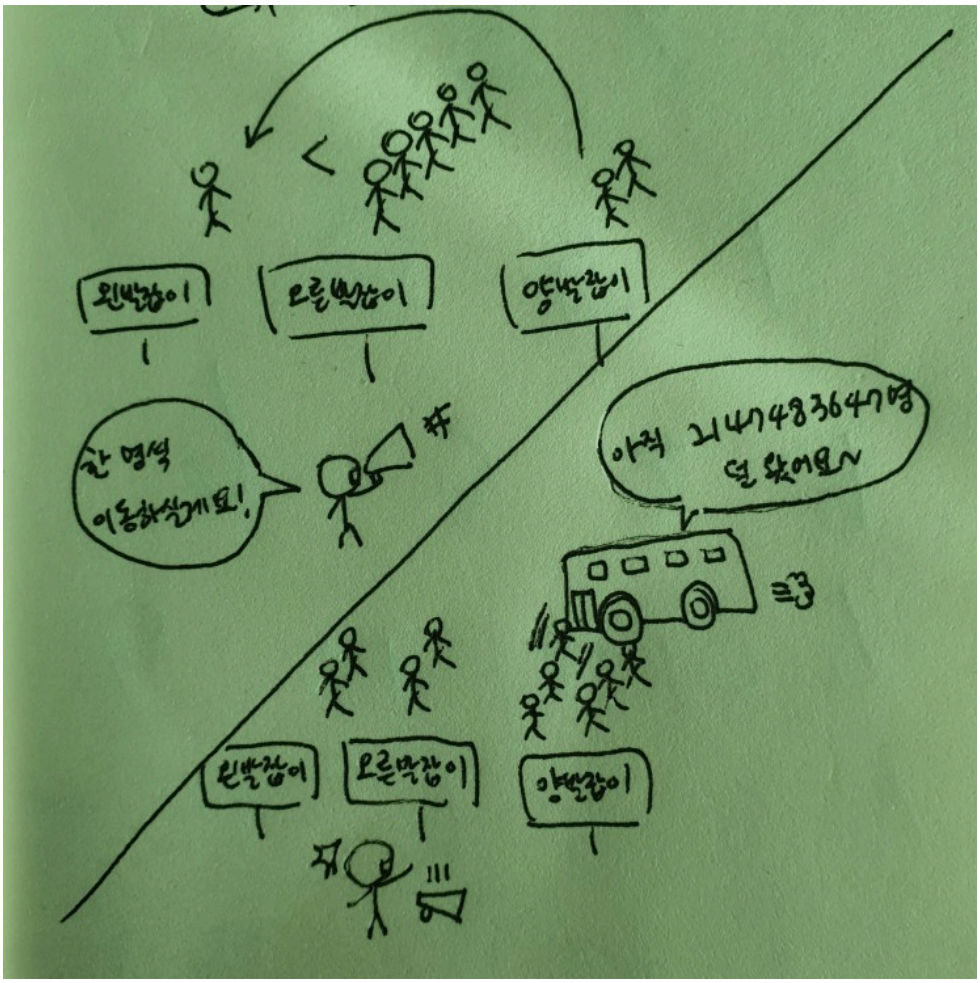
- 수학
- 구현
- 사칙연산

양발잡이 선수를 왼발잡이 선수와 오든발잡이 선수에 최대한 동등하게 분배한 뒤, 왼발잡이 선수와 오든발잡이 선수 중 적은 쪽 선수에 2를 곱한 값이 팀에 잔듀하고 있는 선수의 최대 수가 될 것이다.
채점 현황에 공개된 소스 코드 두 개와 내 코드의 풀이 방법을 소개해 보려고 한다.

1. 양받잡이 선수를 한 명씩 왼받잡이, 오른받잡이 선수로 바꾸기

```
# 제출 번호 48580887번 소스 코드, 아이디 smiley123, 언어 Python 3
a, b, c = map(int,input().split())
while c != 0:
    if a > b:
        c -= 1
        b += 1
    else:
        c -= 1
        a += 1
print(min(a,b)*2)
```

양발잡이 선수를 한 명씩 왼발잡이 선수가 많을 때는 오든발잡이, 그 외의 경우에는 왼발잡이 선수로 바꾸는 풀이법이다.
선수들의 수 L, R, A는 100 이하의 정수이므로 while문은 길어야 A=100일 때 300번 반복된다.
이 문제를 푸는 데는 충분한 횟수지만, 만약 A의 최댓값이 커진다면 어떻게 될까?



직접 그려 본 문제 풀이 방식 설명 그림 1

while문의 반복 횟수는 A의 최댓값에 비례하여 증가한다.
따라서 양발잡이 선수의 수가 일정 값 이상으로 커지면 문제 풀이에 너무 많은 시간이 걸리게 된다.
이쯤에서 다른 문제 풀이 방법도 살펴보자.

2. 4가지 경우로 나누어 풀이

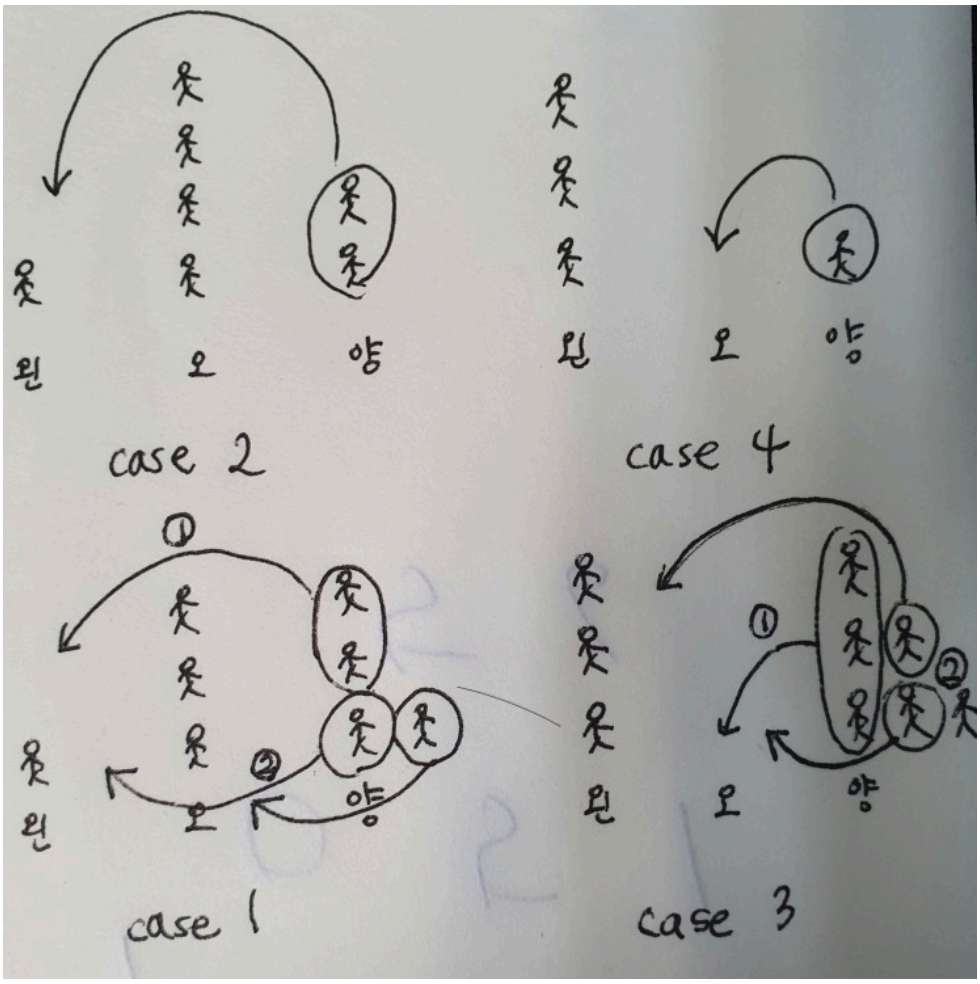
```python
# 제출 번호 49606498번 소스 코드, 아이디 panpan8, 언어 Python 3
import sys
input = sys.stdin.readline

L, R, A = map(int, input().split())
if L <= R:
    if R - L <= A:
        ans = (R + (A - (R - L)) // 2) * 2
    else:
        ans = (L + A) * 2
else:
    if L - R <= A:
        ans = (L + (A - (L - R)) // 2) * 2
    else:
        ans = (R + A) * 2
print(ans)
```

내가 푼 방식인데, 다음과 같이 네 가지 경우로 나누어 각각에 맞는 답을 출력하는 방식의 풀이법이다.

1. 오른발잡이 선수가 왼발잡이 선수보다 많거나 같고, 두 선수의 차보다 양발잡이 선수가 많거나 같을 때

2. 오든발잡이 선수가 왼발잡이 선수보다 많거나 같고, 두 선수의 차보다 양발잡이 선수가 적을 때
3. 오든발잡이 선수가 왼발잡이 선수보다 적고, 두 선수의 차보다 양발잡이 선수가 많거나 같을 때
4. 오든발잡이 선수가 왼발잡이 선수보다 적고, 두 선수의 차보다 양발잡이 선수가 적을 때

1번과 3번의 경우 우선 왼발잡이, 오든발잡이의 수 중 적은 쪽 발잡이 선수 집단으로 양발잡이 선수를 넣어 두 집단의 수를 같게 만들어주고 남은 나머지 양발잡이 선수를 반으로 나누어 넣는다.
2번과 4번의 경우 양발잡이 선수 모두를 부족한 발잡이 선수 집단에 넣는다.



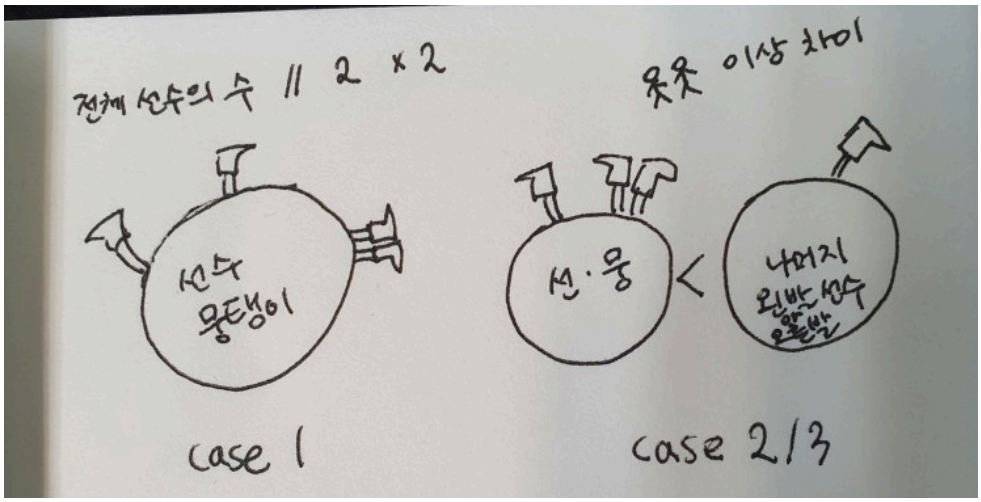
직접 그려 본 문제 풀이 방식 설명 그림 2

예외가 생기지 않도록 분류되어 있고, 선수의 수가 문제 풀이 시간에 영향을 주지 않는다.
한 번에 정답을 구할 수 있는 풀이법이지만, 다소 난해하게 느껴질 수도 있다.
좀 더 쉽고 직관적인 풀이 방법은 없을까?

3. 3가지 경우로 나누어 푼이

```
# 제출 번호 49228058번 소스 코드, 아이디 ssss9169, 언어 Python 3
L, R, A = map(int,input().split())

print(min(L+A,R+A,(L+R+A)//2)*2)
```

앞서 소개한 문제 풀이 방법 두 가지보다 코드도 짧고 이해하기 쉽다.
짧은 만큼 설명이 필요하다.
왜 왼발잡이 선수와 양발잡이 선수의 합, 오든발잡이 선수와 양발잡이 선수의 합, 그리고 전체 선수의 합을 2로 나눈 몫 중 최솟값에 2를 곱한 것이 정답일까?



직접 그려 본 문제 풀이 방식 설명 그림 3

쉽게 생각해 보자.
가능만 하다면 전체 선수의 수를 2로 나눈 몫에 2를 곱하는 방식이 제일 쉬운 방법이다.
이 방법이 통하지 않는 경우는 어떤 경우일까?

아래 두 가지 경우로 나뉜다.
1. 왼발잡이 선수가 오른발잡이 선수와 양발잡이 선수의 합보다 클 때
2. 오른발잡이 선수가 왼발잡이 선수와 양발잡이 선수의 합보다 클 때

이때 정답은 특정발잡이 선수와 양발잡이 선수의 합에 2를 곱한 값이 되므로 이것이 전체 선수를 2로 나눈 몫에 2를 곱한 값과 다를 수 있다.
한쪽발잡이 선수와 양발잡이 선수의 합과 반대쪽발잡이 선수의 합의 차이가 2 이상이 되면 무조건 달라지게 된다.
소스 코드에 조건이 드러나게 풀이한다면 이런 풀이도 가능하다.

```
# 제출 번호 49610873번 소스 코드, 아이디 panpan8, 언어 Python 3
import sys
input = sys.stdin.readline

L, R, A = map(int, input().split())
if L + A < R:
    print((L + A) * 2)
elif R + A < L:
    print((R + A) * 2)
else:
    print((L + R + A) // 2 * 2)
```

문제를 푸는 것보다 그 문제를 어떻게 풀었는지 설명하는 것이 더 어렵다.
풀이에 명확한 근거를 들고 설명할 수 있어야 한다.
세상에서 가장 어려운 문제를 풀 수 있는 사람은 되지 못하더라도, 세상에서 가장 따라 하기 쉬운 사람 중 가장 어려운 문제를 풀 수 있는 사람이 되고 싶다.