# 题目

你正在玩一个单人游戏，面前放置着大小分别为 a​​​​​​、b 和 c​​​​​​ 的 三堆 石子。

每回合你都要从两个 不同的非空堆 中取出一颗石子，并在得分上加 1 分。当存在 两个或更多 的空堆时，游戏停止。

给你三个整数 a 、b 和 c ，返回可以得到的 最大分数 。

示例 1：

输入：a = 2, b = 4, c = 6

输出：6

解释：石子起始状态是 (2, 4, 6) ，最优的一组操作是：

- 从第一和第三堆取，石子状态现在是 (1, 4, 5)

- 从第一和第三堆取，石子状态现在是 (0, 4, 4)

- 从第二和第三堆取，石子状态现在是 (0, 3, 3)

- 从第二和第三堆取，石子状态现在是 (0, 2, 2)

- 从第二和第三堆取，石子状态现在是 (0, 1, 1)

- 从第二和第三堆取，石子状态现在是 (0, 0, 0)

总分：6 分 。

示例 2：

输入：a = 4, b = 4, c = 6

输出：7

解释：石子起始状态是 (4, 4, 6) ，最优的一组操作是：

- 从第一和第二堆取，石子状态现在是 (3, 3, 6)

- 从第一和第三堆取，石子状态现在是 (2, 3, 5)

- 从第一和第三堆取，石子状态现在是 (1, 3, 4)

- 从第一和第三堆取，石子状态现在是 (0, 3, 3)

- 从第二和第三堆取，石子状态现在是 (0, 2, 2)

- 从第二和第三堆取，石子状态现在是 (0, 1, 1)

- 从第二和第三堆取，石子状态现在是 (0, 0, 0)

总分：7 分 。

示例 3：

输入：a = 1, b = 8, c = 8

输出：8

解释：最优的一组操作是连续从第二和第三堆取 8 回合，直到将它们取空。

注意，由于第二和第三堆已经空了，游戏结束，不能继续从第一堆中取石子。

提示：

1 <= a, b, c <= 105

# 分析

要解决“三堆石子取最大分数”的问题，核心思路是利用贪心策略结合数学规律：通过分析取石子的最优操作，得出“最大分数由两堆较小石子的总和与最大堆石子的关系决定”的结论，避免模拟每一步操作（降低时间复杂度）。

解题思路

1、理解游戏规则的核心约束：

每回合需从两个不同的非空堆各取1颗石子，得1分；游戏停止条件是“存在两个或更多空堆”。因此，最终分数的上限由两方面决定：

- 两堆较小石子的总和（记为sum\_small）：若最大堆石子数（记为max\_pile）≥ sum\_small，则取完两堆较小石子后，仅剩最大堆非空，游戏停止，此时分数等于sum\_small。

- 三堆石子的总和的一半（记为total / 2）：若最大堆石子数 < sum\_small，则三堆石子可尽可能取到“仅剩0堆或1堆非空”，此时分数等于总石子数的一半（因每回合消耗2颗石子，总分数≤总石子数//2）。

2、数学规律推导：

设三堆石子排序后为 x ≤ y ≤ z（x最小，z最大），则：

- 若 z ≥ x + y：最多取x + y回合（取完x和y后，只剩z非空，游戏停止）。

- 若 z < x + y：最多取(x + y + z) // 2回合（总石子数为x+y+z，每回合消耗2颗，最终可能剩0或1颗石子，分数为总石子数的整数除法结果）。

3、步骤简化：

无需模拟每一步操作，只需：

1）对三堆石子排序，区分出最小、中间、最大堆。

2）根据上述两种情况计算最大分数。

代码：

class Solution {

public:

int maximumScore(int a, int b, int c) {

// 1. 将三堆石子排序，得到 x ≤ y ≤ z（x最小，z最大）

int x = min({a, b, c}); // 最小堆

int z = max({a, b, c}); // 最大堆

int y = a + b + c - x - z; // 中间堆（总和减去最小和最大）

// 2. 根据两种情况计算最大分数

if (z >= x + y) {

// 情况1：最大堆 ≥ 两堆较小之和，分数为两堆较小之和

return x + y;

} else {

// 情况2：最大堆 < 两堆较小之和，分数为总石子数的一半

return (x + y + z) / 2;

}

}

};

代码解释

1、排序三堆石子：

通过min({a,b,c})和max({a,b,c})直接获取最小堆x和最大堆z，再通过“总石子数 - x - z”得到中间堆y（避免三次比较，简化代码）。

2、分数计算逻辑：

- 若z ≥ x + y：取完x和y后，仅剩z非空，游戏停止，分数为x + y（如示例3：a=1, b=8, c=8，排序后x=1, y=8, z=8，z=8 < 1+8=9，进入情况2，总石子数17，分数17//2=8，与示例一致）。

- 若z < x + y：总石子数x+y+z，每回合消耗2颗，分数为(x+y+z)//2（如示例2：a=4, b=4, c=6，排序后x=4, y=4, z=6，z=6 < 4+4=8，总石子数14，分数14//2=7，与示例一致）。

复杂度分析

- 时间复杂度：O(1)。排序3个数字的操作是常数时间（无需遍历，仅需固定次数的比较），后续计算均为算术运算，无循环依赖输入规模。

- 空间复杂度：O(1)。仅使用固定数量的变量（`x`、`y`、`z`），无额外动态空间开销。