

有三种葡萄，每种分别有 a, b, c 颗。有三个人，第一个人只吃第 1, 2 种葡萄，第二个人只吃第 2, 3 种葡萄，第三个人只吃第 1, 3 种葡萄。
适当安排三个人使得吃完所有的葡萄，并且且三个人中吃的最多的那个人吃得尽量少

分析：

1. 只要平均分，每个人吃 $(a+b+c)/3$ 颗葡萄；
2. 吃的最多的人吃的最少，表示尽可能平均分， $(a+b+c)/3$ 向上取整，即 $(a+b+c+2)/3$ ；

大部分编程语言中，如果你想计算 M 除以 N ， M / N 会向下取整，你想向上取整的话，可以改成 $(M+(N-1)) / N$

3. 把葡萄的颗数 a, b, c 作为三条线段，它们的大小作为线段的长度,可以组成什么图形：

- 如果 $a+b>c$ 可以构成三角形，取每条边的中点就可以将这个三角形均分，则每个人可以吃到的数目为 $(a+b+c+2)/3$ ；
- 如果 $a+b \leq c$ ，可以考虑将 c 折成两段，构成一个四边形；
 - 若 $a+b$ 与 c 的差距不大时，均分仍然为 $(a+b+c+2)/3$
 - 如 $c > 2*(a+b)$ ，将 c 均分即可 $(c+1)/2$

```
long solution(long a, long b, long c) {
    std::vector<long> nums = {a,b,c};
    sort(nums.begin(), nums.end());
    long sum = a + b + c;

    // 构成三角形，完全平分
    if(nums[0]+nums[1] > nums[2]) {
        return (sum + 2) / 3;
    }

    // 不构成三角形，平分最长边
    if(2 * (nums[0] + nums[1]) < nums[2]) {
        return (nums[2] + 1) / 2;
    }

    return (sum+2)/3;
}
```