

复合操作：加还是乘

- 假设算法有两步，每一步的时间复杂度为 $O(A)$ ， $O(B)$

计算时间复杂度时什么时候该将两步的时间复杂度相加，什么时候该相乘

```
for (int a : arrA) {  
    print(a);  
}  
  
for (int b : arrB) {  
    print(b);  
}
```

$O(A + B)$

```
for (in a : arrA) {  
    for (int b : arrB) {  
        print(a + "," + b);  
    }  
}
```

$O(A * B)$

- 结论：
1. 先做A，然后做完A后，再做B，应该将这两件事的时间复杂度相加
 2. 每一次做A的时候都需要将B全部做一遍，应该将这两件事的时间复杂度相乘

例题

```
void foo(int[] array) {  
    int sum = 0;  
    int product = 1;  
    for (int i = 0; i < array.length; i++) {  
        sum += array[i];  
    }  
  
    for (int i = 0; i < array.length; i++) {  
        product *= array[i]  
    }  
    System.out.println(sum + ", " + product);  
}
```