

基本操作的时间复杂度 - 丢弃次要项

- $O(n^2 + n) \rightarrow O(n^2)$
- $O(\log n + n) \rightarrow O(n)$
- $O(5 * 2^n + 1000n^{100}) \rightarrow O(2^n)$

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