

时间复杂度

$T(n) = O(f(n))$ 执行次数与 $f(n)$ 成正比

//例一:

```
for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
        c[i][j] = 0;
        for(int k=1;k<=n;k++)
            c[i][j] += a[i][k] * b[k][j];
```

/*

基本操作: 相乘

时间复杂度: $O(n^3)$

*/

//例二:

```
void select_sort(int* a,const int& n){
    for(int i=0;i<n-1;i++){
        j=i;
        for(k=i+1;k<n;++k)
            if(a[k]<a[j]) j=k;
        if(j!=i) swap(a[j],a[i]);
    }
}
```

//select_sort

/*

基本操作: 比较元素的操作

操作次数: $(n-1)n/2$

时间复杂度: $O(n^2)$

*/

//以上两个算法的时间复杂度与输入数据的特点没关系, 仅与问题规模有关。

//例三:

```
void bubble_sort(int* a, int n){
    for(int i=n-1,change=true;i>1&&change;--i){
        change = false;
        for(j=0;j<i;j++){
            if(a[j]>a[j+1]){
                swap(a[j],a[j+1]);
                change=true;
            }
        }
    }
}
```

/*

基本操作: 赋值操作: swap函数本质是赋值过程

时间复杂度: $O(n^2)$ (最坏的情况), 执行次数与 n^2 成正比。

算法的空间复杂度

$$S(n) = O(g(n))$$

算法储存量：

1. 输入数据的空间
2. 程序本身的空间
3. 辅助变量的空间

以上两个复杂度的概念是一个“数量级”的一个概念

学习要点：

1. 熟悉各类名词、术语、掌握基本概念
2. 理解算法五要素的确切含义
有穷性、确定性、可行性、输入、输出
3. 估算时间复杂度
4. 课后题：
两个函数， n^2 比较 $50n\log_2(n)$ 的差别