



王道计算机考研
www.cskaoyan.com

算法的评价

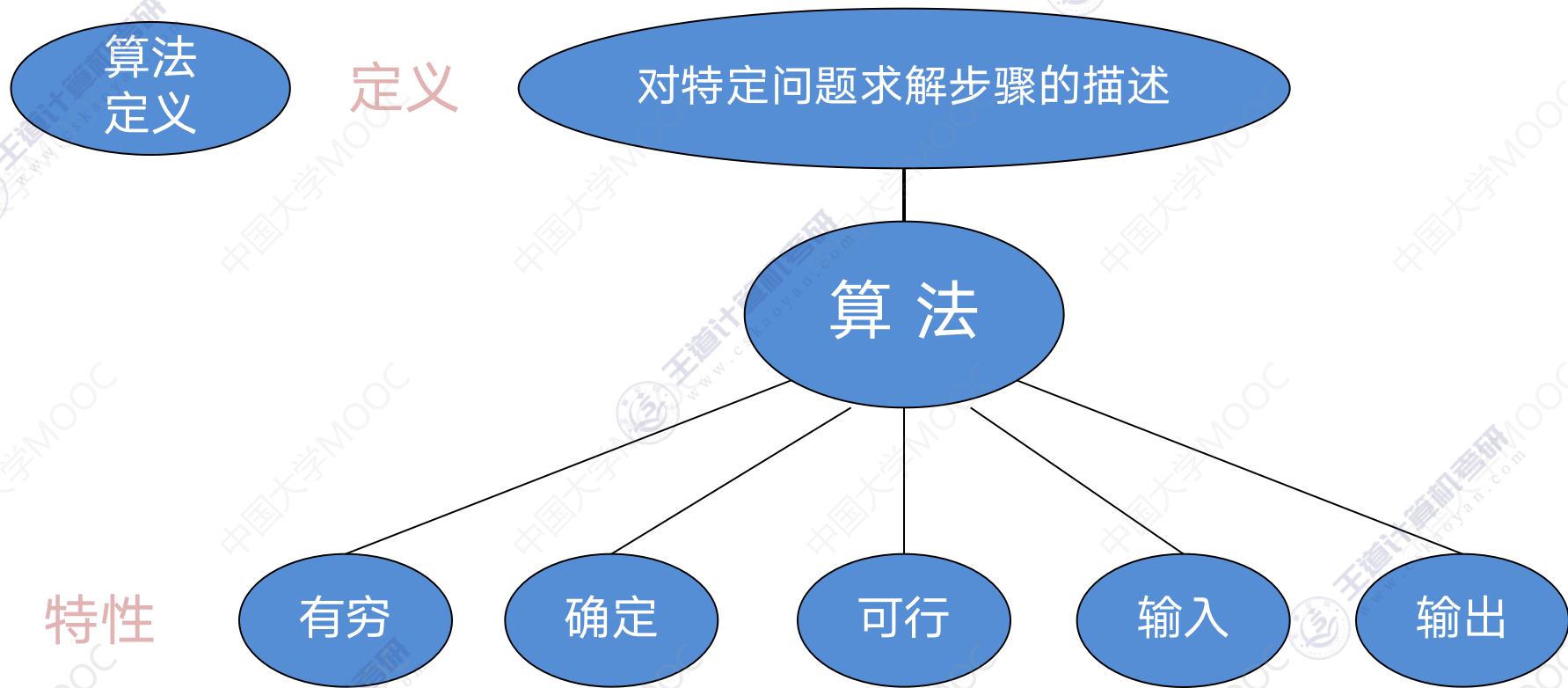
(时间复杂度与空间复杂度)

王道论坛

中国大学慕课



算法定义





时间复杂度



时间 复杂度

时间复杂度指算法中所有语句的频度（执行次数）之和。

记为：

$$T(n)=O(f(n))$$

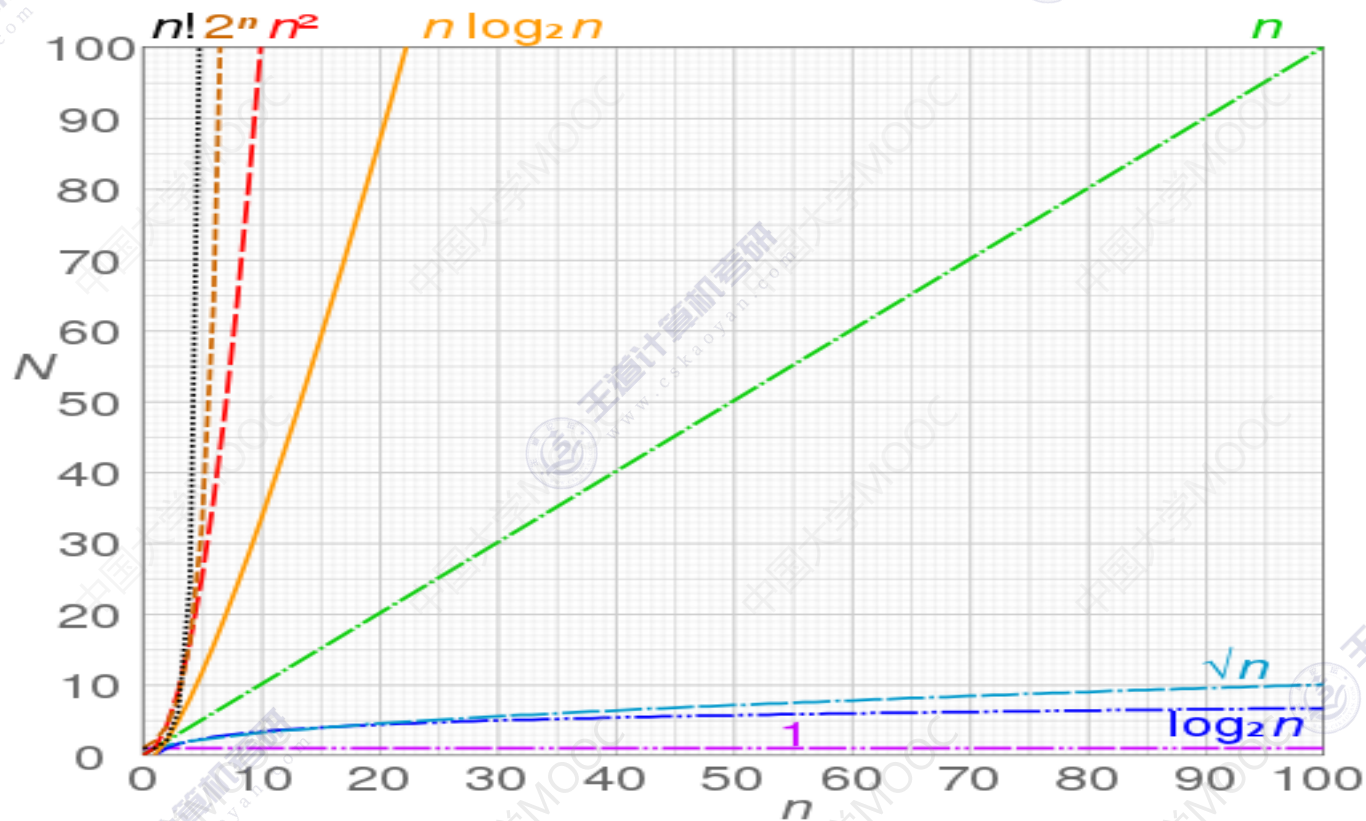
其中， n 是问题的规模； $f(n)$ 是问题规模 n 的某个函数。

表示随着问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同。

常见的时间复杂度：

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$$

最高阶数越小，说明算法的时间性能越好。





时间 复杂度

示例程序1:

```
int sum=0;  
sum=n*(n+1)/2;  
printf("%d",sum);
```

```
//执行一次  
//执行一次  
//执行一次
```

算法的执行次数等于3。

时间复杂度为 $T(n)=O(1)$

表示不会随n的增长而增长。



时间
复杂度

示例程序2:

```
int x=2;  
while (x<n/2)  
    x=2*x;
```

【2011年计算机联考真题】

执行频率最高的语句为“ $x=2*x$ ”。

设该语句共执行了 t 次，则 $2^{t+1} < n/2$ ，故 $t = \log_2(n/2) - 1 = \log_2 n - 2$

时间复杂度 $T(n) = O(\log_2 n)$



时间 复杂度

示例程序3:

```
int sum=0,i=1;
while (i<n){
    sum=sum+i;
    i++;
}
printf ("%d",sum);
```

执行频率最高的语句是while循环体中的代码。
一共执行n次。

时间复杂度 $T(n)=O(n)$



时间 复杂度

示例程序4:

```
int i, x=2;
for (i=0; i<n; i++)
{
    x=0;
    while (x<n/2)
        x=2*x;
}
```

执行频率最高的语句为“ $x=2*x$ ”。

设该语句内层循环执行了 $\log_2 n$ 次，外层执行了 n 次，因此总计执行次数为 $n\log_2 n$ 次。



时间 复杂度

示例程序5:

```
int i, j;  
for (i=0; i<n; i++) {  
    for (j=0; j<m; j++)  
        sum=sum+1;  
}
```

对于外层循环，相当于内部时间复杂度为 $O(m)$ 的语句再循环 n 次。

所以时间复杂度 $T(n)=O(m \times n)$

如果 $m=n$ ，则时间复杂度 $T(n)=O(n^2)$

➤时间复杂度的乘法规则



时间 复杂度

示例程序6:

```
int sum1=0, sum2=0, i, j;  
for (i=0; i<n; i++)  
    sum1=sum1+i;  
for (j=0; j<m; j++)  
    sum2=sum2+j;  
printf("%d, %d", sum1, sum2);
```

两个循环没有嵌套，串行执行。

所以时间复杂度 $T(n)=O(n)+O(m)$

取最大的，即时间复杂度 $T(n)=\max(O(n)+O(m))$

➤时间复杂度的加法规则



思考题：如果一个算法的执行次数为 $3n^3+5n$ ，那么该算法的时间复杂度为多少？



空间复杂度



空间 复杂度

空间复杂度 $S(n)$ 指算法运行过程中所使用的辅助空间的大小。
记为：

$$S(n)=O(f(n))$$

- 除了需要存储算法本身的指令、常数、变量和输入数据外，还需要存储对数据操作的存储单元。
- 若输入数据所占空间只取决于问题本身，和算法无关，这样只需分析该算法在实现时所需的辅助单元即可。
- 算法原地工作是指算法所需的辅助空间是常量，即 $O(1)$ 。