

算法的由来

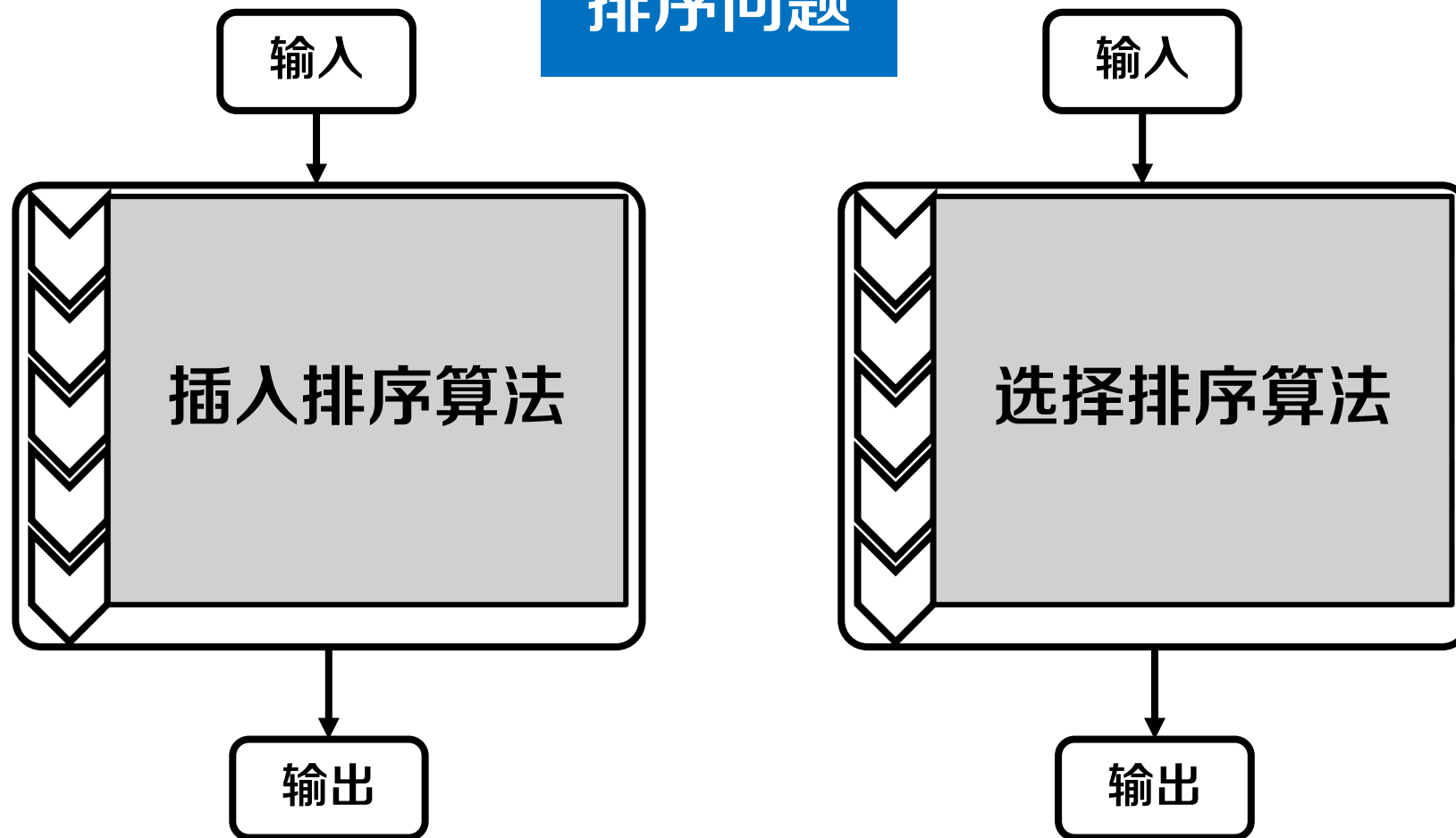
算法的定义

算法的性质

算法的表示

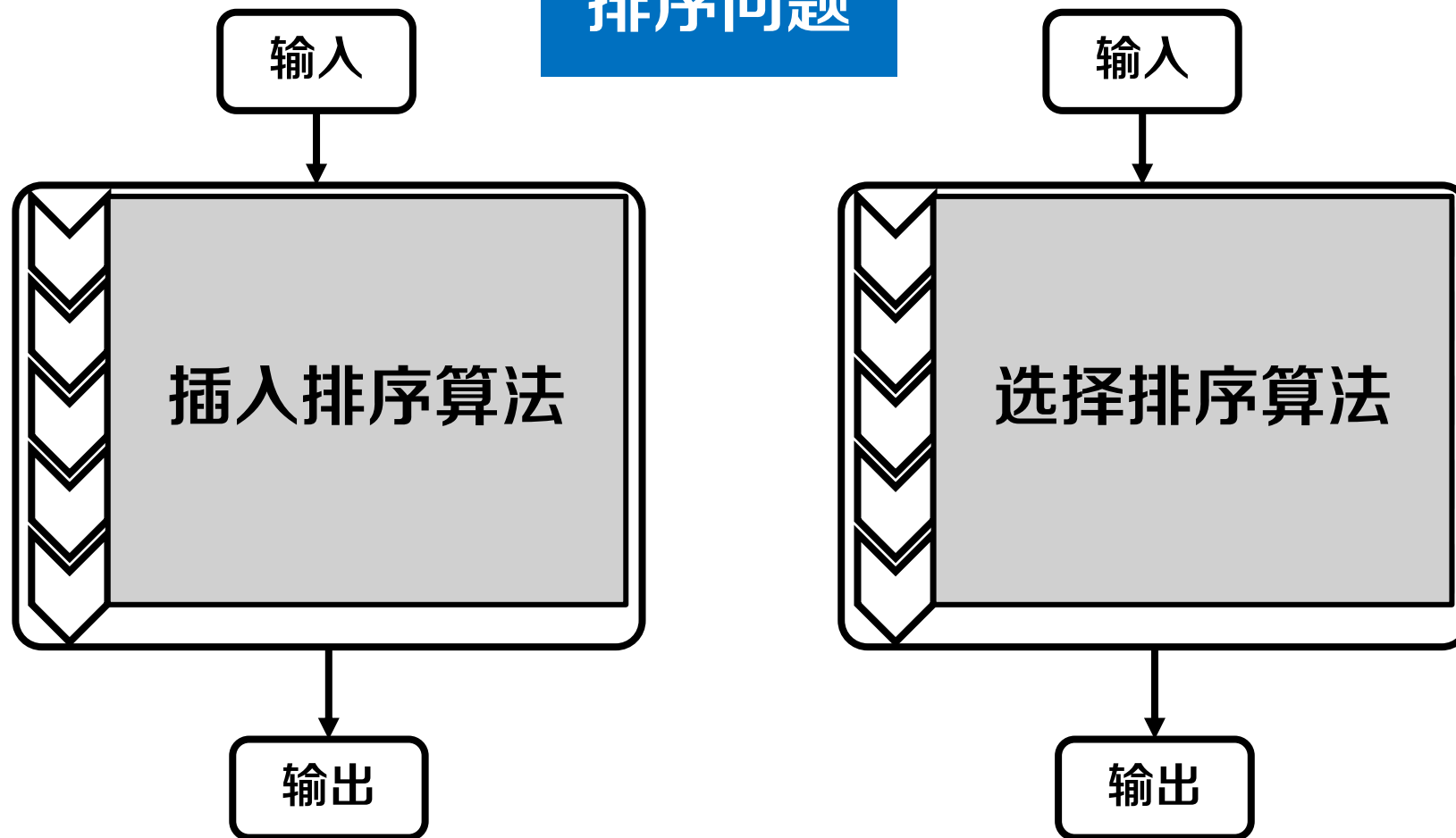
算法的分析

排序问题



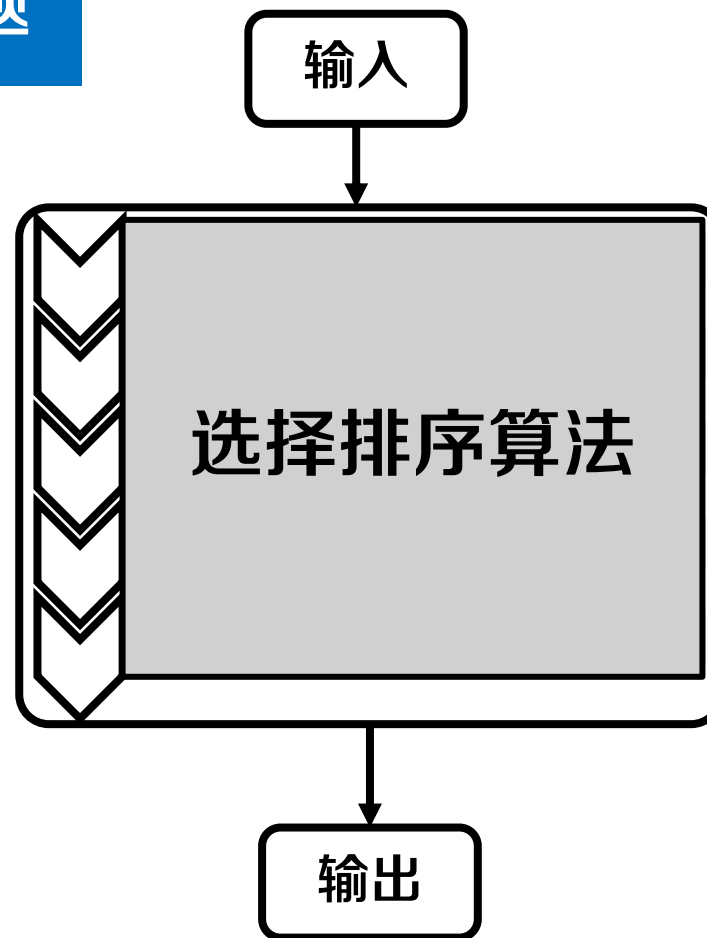
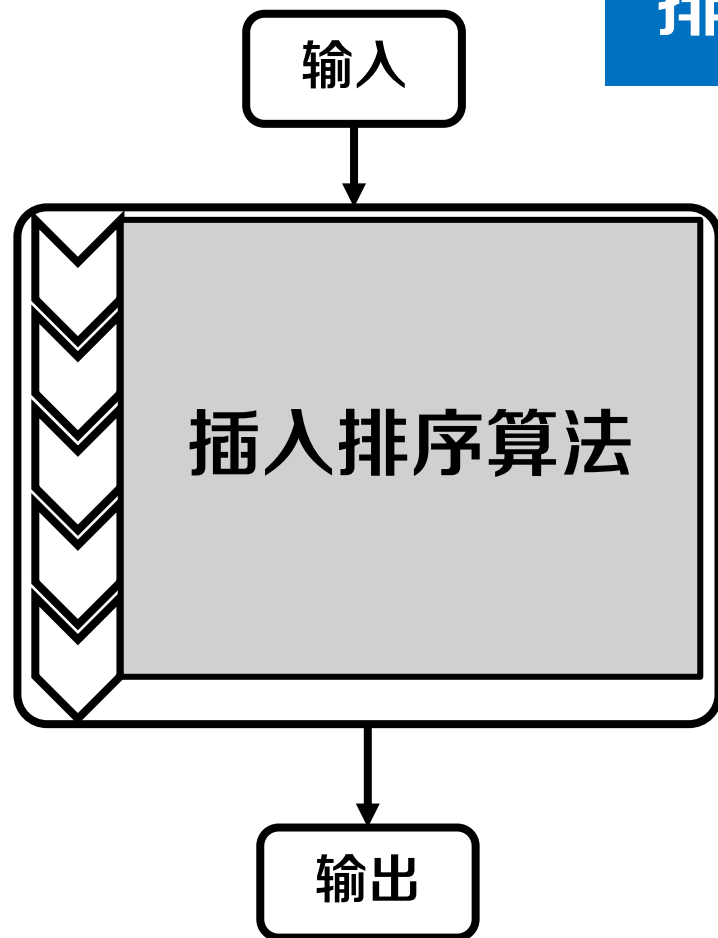
问题：如何比较不同算法性能？

排序问题



分析算法的运行时间

排序问题



算法分析的原则

算法分析的工具

算法分析的实例

分析算法的运行时间

算法分析的原则



- 机器的运算速度影响算法的运行时间

机器	运算速度	运行算法	运行时间
天河三号	百亿亿次/秒	插入排序	无法公平比较
个人电脑	十亿次/秒	选择排序	



算法分析的原则

- 机器的运算速度影响算法的运行时间

机器	运算速度	运行算法	运行时间
天河三号	百亿亿次/秒	插入排序	无法公平比较
个人电脑	十亿次/秒	选择排序	



分析算法的运行时间应独立于机器

- 归纳基本操作

- 如：运算、赋值、比较

+	-	×	÷
:=	>	<	=

算法分析的原则



- 归纳基本操作

- 如：运算、赋值、比较

+	-	×	÷
:=	>	<	=

- 统一机器性能

- 假设基本操作代价均为1



算法分析的原则



- 归纳基本操作

- 如：运算、赋值、比较

+	-	×	÷
:=	>	<	=

- 统一机器性能

- 假设基本操作代价均为1



统一机器性能后，算法运行时间依赖于问题输入规模与实例

- 相同输入规模，实例影响运行

输入: 数组 $A[a_1, a_2, \dots, a_n]$

输出: 升序数组 $A'[a'_1, a'_2, \dots, a'_n]$, 满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$

for $j \leftarrow 2$ to n do

$key \leftarrow A[j]$

$i \leftarrow j - 1$

 while $i > 0$ and $A[i] > key$ do

$A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

 end

$A[i + 1] \leftarrow key$

end

循环次数未知

插入排序算法伪代码

- 相同输入规模，实例影响运行
 - 插入排序最好情况：数组升序
 - 比较次数： $1 + 1 + 1 + \cdots + 1 = n - 1$

$$\underbrace{\hspace{10em}}_{n-1}$$

4	8	13	14	17	18	21	22	24	28	32	37	40	40	47	48
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- 相同输入规模，实例影响运行

- 插入排序最好情况：数组升序

- 比较次数： $1 + 1 + 1 + \cdots + 1 = n - 1$

$$\underbrace{\hspace{10em}}_{n-1}$$

4	8	13	14	17	18	21	22	24	28	32	37	40	40	47	48
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- 插入排序最坏情况：数组降序

- 比较次数： $1 + 2 + 3 + \cdots + (n - 1) = \frac{n(n-1)}{2}$

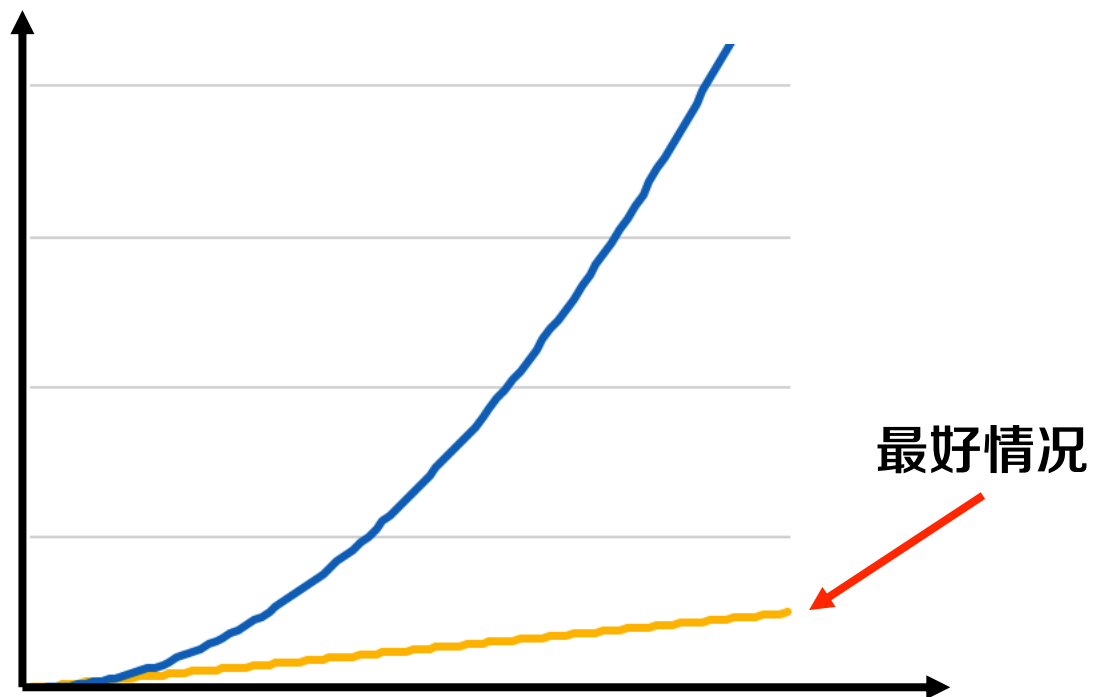
48	47	40	40	37	32	28	24	22	21	18	17	14	13	8	4
----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---

算法分析的原则



输入情况	情况说明
最好情况	不常出现，不具普遍性

运行时间

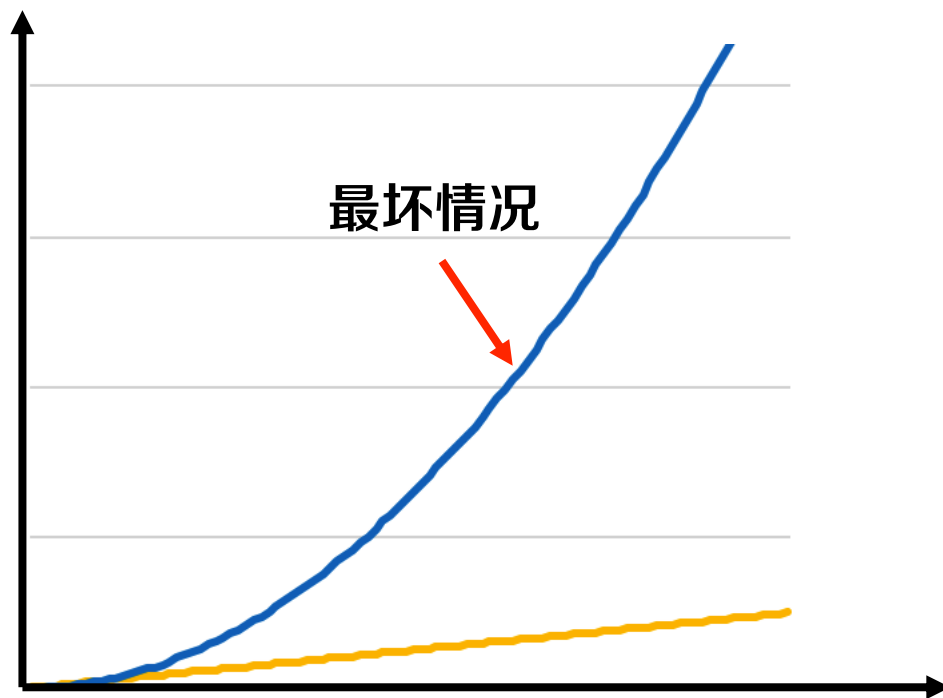


算法分析的原则



输入情况	情况说明
最好情况	不常出现，不具普遍性
最坏情况	确定上界，更具一般性

运行时间

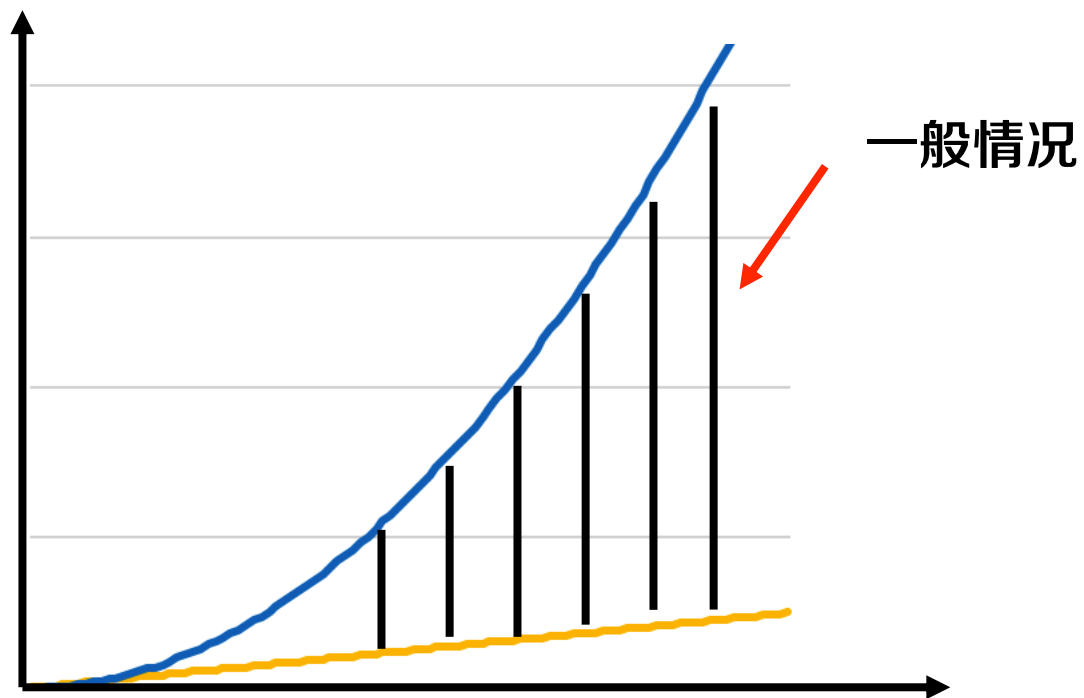


算法分析的原则



输入情况	情况说明
最好情况	不常出现，不具普遍性
最坏情况	确定上界，更具一般性
一般情况	情况复杂，分析难度大

运行时间

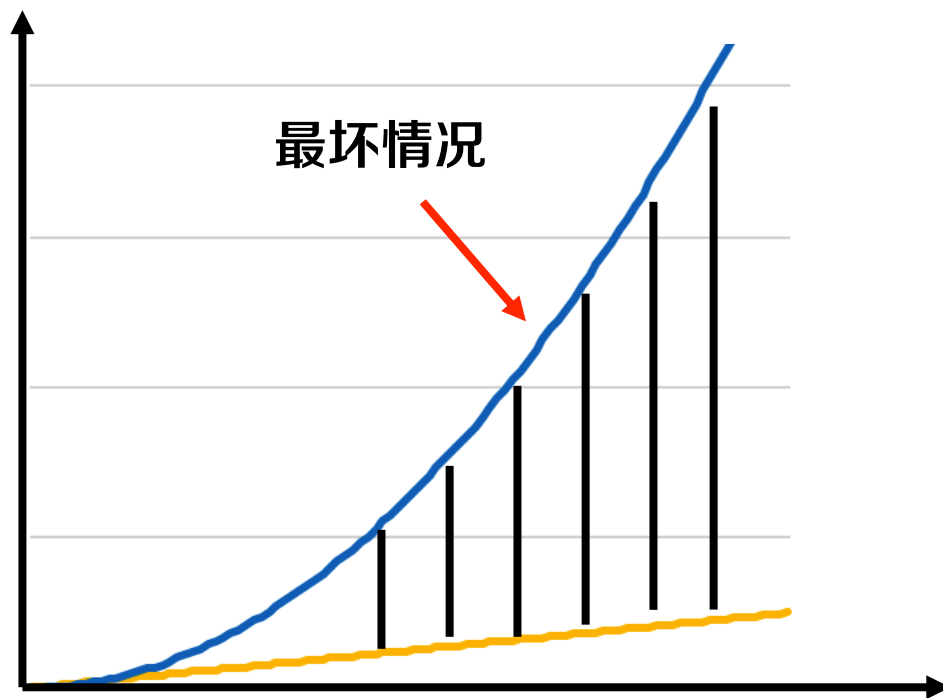


算法分析的原则



输入情况	情况说明
最好情况	不常出现，不具普遍性
最坏情况	确定上界，更具一般性
一般情况	情况复杂，分析难度大

运行时间



常用最坏情况分析算法运行时间

算法分析的原则



- 统一机器性能



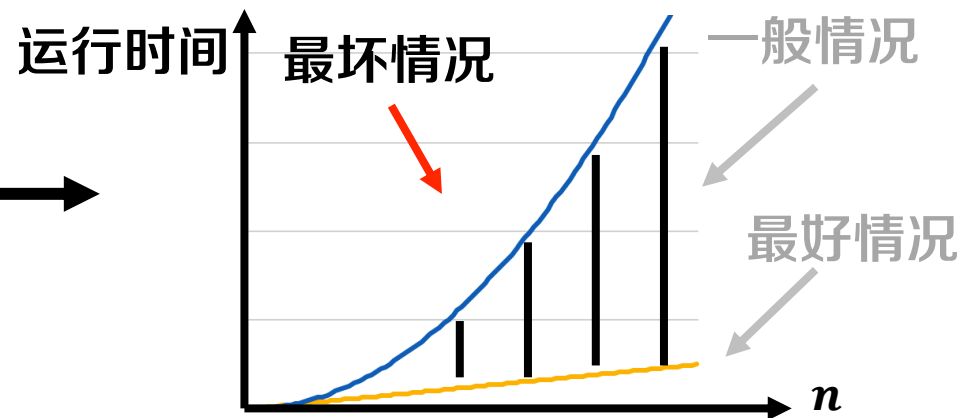
+	-	×	÷
:=	>	<	=



- 分析最坏情况

4	...	40	47	48
---	-----	----	----	----

48	...	13	8	4
----	-----	----	---	---



算法运行时间仅依赖于问题输入规模 n ，表示为 $T(n)$

- 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do
   $key \leftarrow A[j]$ 
   $i \leftarrow j - 1$ 
  while  $i > 0$  and  $A[i] > key$  do
     $A[i + 1] \leftarrow A[i]$ 
     $i \leftarrow i - 1$ 
  end
   $A[i + 1] \leftarrow key$ 
end
```

- 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
   $cur\_min \leftarrow A[i]$ 
   $cur\_min\_pos \leftarrow i$ 
  for  $j \leftarrow i + 1$  to  $n$  do
    if  $A[j] < cur\_min\_pos$  then
       $cur\_min \leftarrow A[j]$ 
       $cur\_min\_pos \leftarrow j$ 
    end
  end
  交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

- 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

- 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
     $cur\_min \leftarrow A[i]$ 
     $cur\_min\_pos \leftarrow i$ 
    for  $j \leftarrow i + 1$  to  $n$  do
        if  $A[j] < cur\_min\_pos$  then
             $cur\_min \leftarrow A[j]$ 
             $cur\_min\_pos \leftarrow j$ 
        end
    end
    交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

计算每行伪代码操作次数

● 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do .....  $n$ 次  
     $key \leftarrow A[j]$  .....  $n - 1$ 次  
     $i \leftarrow j - 1$  .....  $n - 1$ 次  
    while  $i > 0$  and  $A[i] > key$  do  
         $A[i + 1] \leftarrow A[i]$   
         $i \leftarrow i - 1$   
    end  
     $A[i + 1] \leftarrow key$   
end
```

● 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do  
     $cur\_min \leftarrow A[i]$   
     $cur\_min\_pos \leftarrow i$   
    for  $j \leftarrow i + 1$  to  $n$  do  
        if  $A[j] < cur\_min\_pos$  then  
             $cur\_min \leftarrow A[j]$   
             $cur\_min\_pos \leftarrow j$   
        end  
    end  
    交换  $A[i]$  和  $A[cur\_min\_pos]$   
end
```

计算每行伪代码操作次数

● 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do .....  $n$ 次
|    $key \leftarrow A[j]$  .....  $n - 1$ 次
|    $i \leftarrow j - 1$  .....  $n - 1$ 次
|   while  $i > 0$  and  $A[i] > key$  do...  $\sum_{k=2}^n k$ 次
|   |    $A[i + 1] \leftarrow A[i]$  .....  $\sum_{k=2}^n k - 1$ 次
|   |    $i \leftarrow i - 1$  .....  $\sum_{k=2}^n k - 1$ 次
|   end
|    $A[i + 1] \leftarrow key$ 
end
```

● 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
|    $cur\_min \leftarrow A[i]$ 
|    $cur\_min\_pos \leftarrow i$ 
|   for  $j \leftarrow i + 1$  to  $n$  do
|   |   if  $A[j] < cur\_min\_pos$  then
|   |   |    $cur\_min \leftarrow A[j]$ 
|   |   |    $cur\_min\_pos \leftarrow j$ 
|   |   end
|   end
|   交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

计算每行伪代码操作次数

● 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do .....  $n$ 次
|    $key \leftarrow A[j]$  .....  $n - 1$ 次
|    $i \leftarrow j - 1$  .....  $n - 1$ 次
|   while  $i > 0$  and  $A[i] > key$  do...  $\sum_{k=2}^n k$ 次
|   |    $A[i + 1] \leftarrow A[i]$  .....  $\sum_{k=2}^n k - 1$ 次
|   |    $i \leftarrow i - 1$  .....  $\sum_{k=2}^n k - 1$ 次
|   end
|    $A[i + 1] \leftarrow key$  .....  $n - 1$ 次
end
```

● 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
|    $cur\_min \leftarrow A[i]$ 
|    $cur\_min\_pos \leftarrow i$ 
|   for  $j \leftarrow i + 1$  to  $n$  do
|   |   if  $A[j] < cur\_min\_pos$  then
|   |   |    $cur\_min \leftarrow A[j]$ 
|   |   |    $cur\_min\_pos \leftarrow j$ 
|   |   end
|   end
|   交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

计算每行伪代码操作次数

● 插入排序最坏情况

```
for j ← 2 to n do .....  $n$ 次
|   key ← A[j] .....  $n - 1$ 次
|   i ← j - 1 .....  $n - 1$ 次
|   while i > 0 and A[i] > key do...  $\sum_{k=2}^n k$ 次
|   |   A[i + 1] ← A[i] .....  $\sum_{k=2}^n k - 1$ 次
|   |   i ← i - 1 .....  $\sum_{k=2}^n k - 1$ 次
|   end
|   A[i + 1] ← key .....  $n - 1$ 次
end
```

● 选择排序最坏情况

```
for i ← 1 to n - 1 do .....  $n$ 次
|   cur_min ← A[i] .....  $n - 1$ 次
|   cur_min_pos ← i .....  $n - 1$ 次
|   for j ← i + 1 to n do .....  $\sum_{k=0}^{n-2} (n - k)$ 次
|   |   if A[j] < cur_min_pos then .....  $\sum_{k=1}^{n-1} (n - k)$ 次
|   |   |   cur_min ← A[j] .....  $\sum_{k=1}^{n-1} (n - k)$ 次
|   |   |   cur_min_pos ← j .....  $\sum_{k=1}^{n-1} (n - k)$ 次
|   |   end
|   end
|   交换 A[i] 和 A[cur_min_pos] .....  $\sum_{k=1}^{n-1} (n - k)$ 次
end .....  $n - 1$ 次
```

计算每行伪代码操作次数

● 插入排序最坏情况

```
for j ← 2 to n do ..... n次
| key ← A[j] ..... n - 1次
| i ← j - 1 ..... n - 1次
| while i > 0 and A[i] > key do...  $\sum_{k=2}^n k$ 次
| | A[i + 1] ← A[i] .....  $\sum_{k=2}^n k - 1$ 次
| | i ← i - 1 .....  $\sum_{k=2}^n k - 1$ 次
| end
| A[i + 1] ← key ..... n - 1次
end
```

求和

$$T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4$$

● 选择排序最坏情况

```
for i ← 1 to n - 1 do ..... n次
| cur_min ← A[i] ..... n - 1次
| cur_min_pos ← i ..... n - 1次
| for j ← i + 1 to n do .....  $\sum_{k=0}^{n-2} (n - k)$ 次
| | if A[j] < cur_min_pos then .....  $\sum_{k=1}^{n-1} (n - k)$ 次
| | | cur_min ← A[j] .....  $\sum_{k=1}^{n-1} (n - k)$ 次
| | | cur_min_pos ← j .....  $\sum_{k=1}^{n-1} (n - k)$ 次
| | end
| end
| 交换 A[i] 和 A[cur_min_pos] ..... n - 1次
end
```

求和

$$T(n) = 2n^2 + 3n - 4$$

● 插入排序最坏情况

```
for j ← 2 to n do ..... n次
| key ← A[j] ..... n - 1次
| i ← j - 1 ..... n - 1次
| while i > 0 and A[i] > key do...  $\sum_{k=2}^n k$ 次
| | A[i + 1] ← A[i] .....  $\sum_{k=2}^n k - 1$ 次
| | i ← i - 1 .....  $\sum_{k=2}^n k - 1$ 次
| end
| A[i + 1] ← key ..... n - 1次
end
```

求和

$$T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4$$

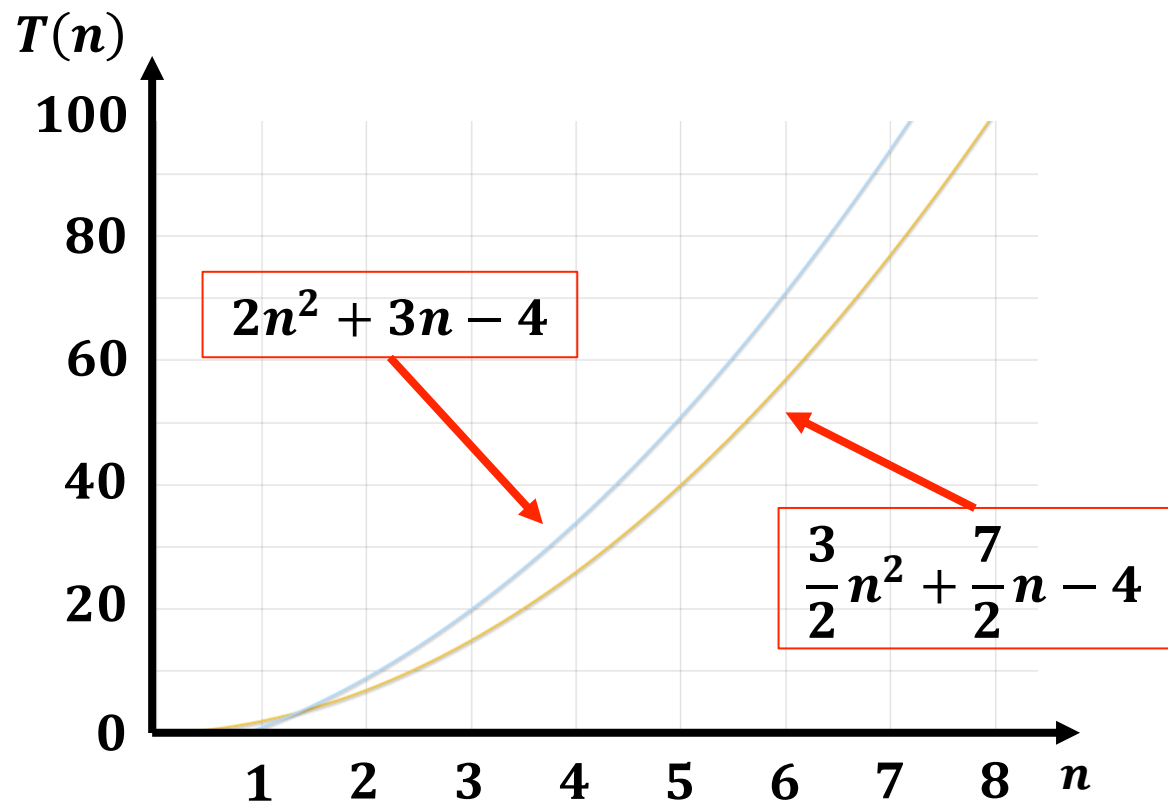
● 选择排序最坏情况

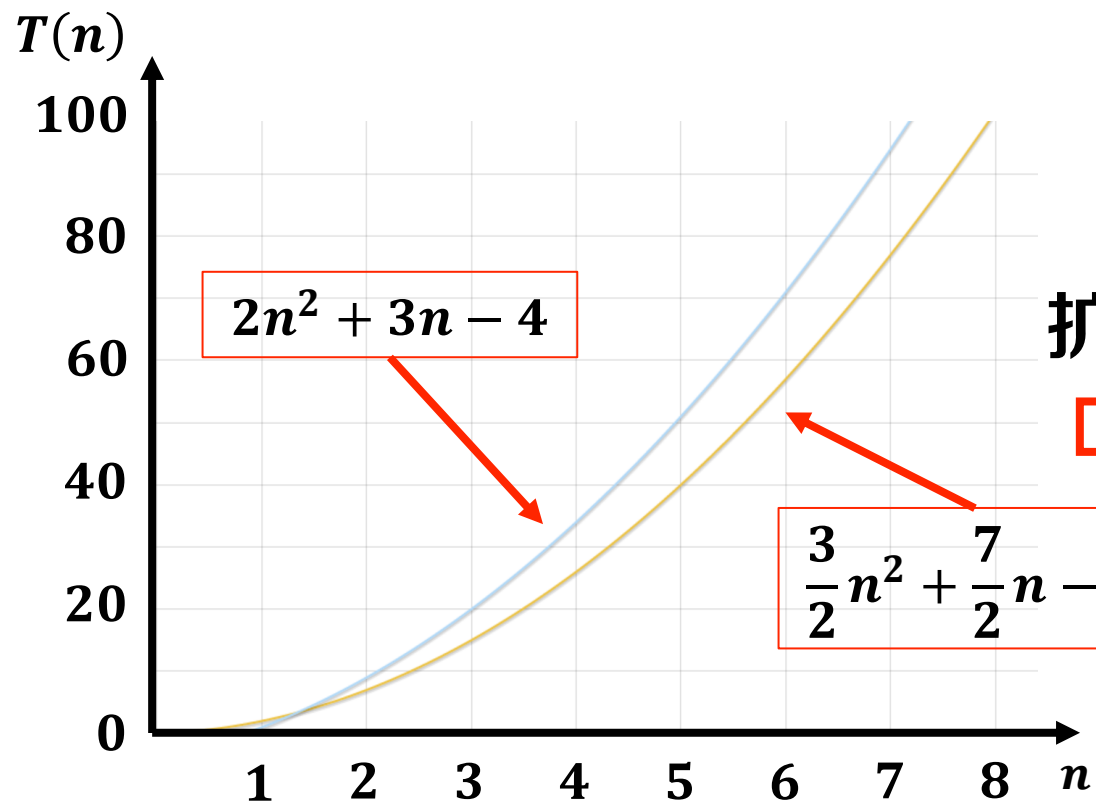
```
for i ← 1 to n - 1 do ..... n次
| cur_min ← A[i] ..... n - 1次
| cur_min_pos ← i ..... n - 1次
| for j ← i + 1 to n do .....  $\sum_{k=0}^{n-2} (n - k)$ 次
| | if A[j] < cur_min_pos then .....  $\sum_{k=1}^{n-1} (n - k)$ 次
| | | cur_min ← A[j] .....  $\sum_{k=1}^{n-1} (n - k)$ 次
| | | cur_min_pos ← j .....  $\sum_{k=1}^{n-1} (n - k)$ 次
| | end
| end
| 交换 A[i] 和 A[cur_min_pos] ..... n - 1次
end
```

求和

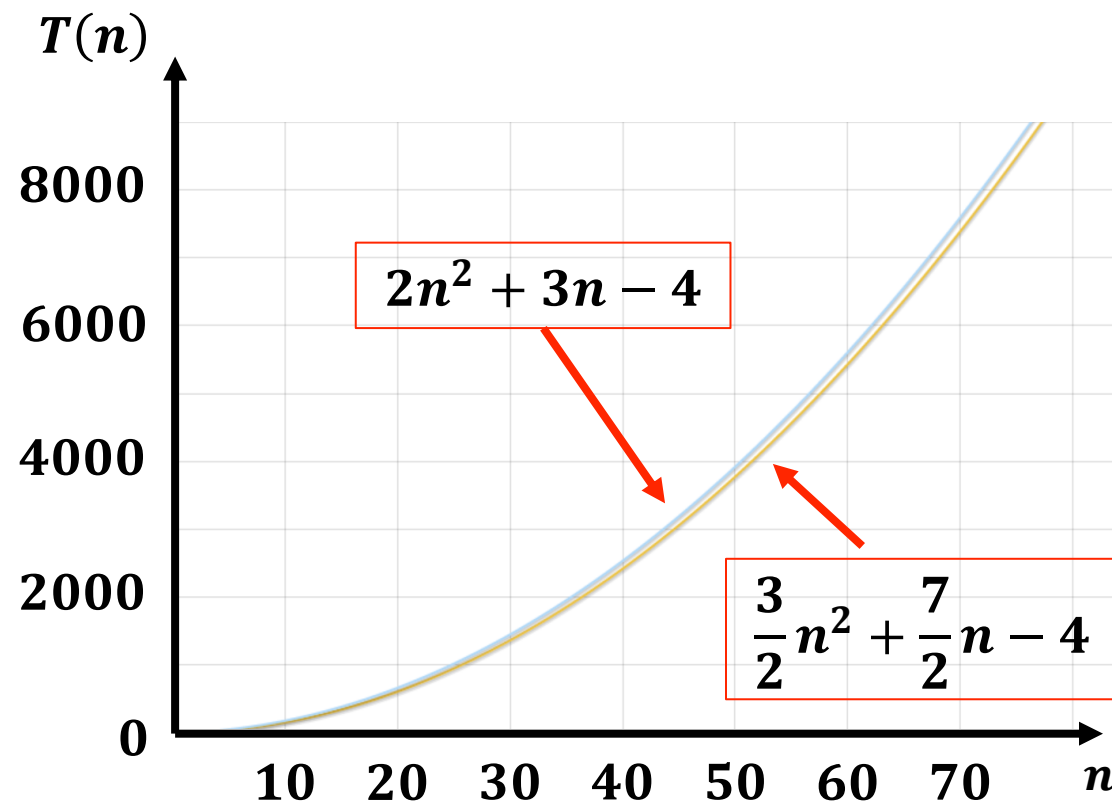
$$T(n) = 2n^2 + 3n - 4$$

问题：能否简洁地衡量算法运行时间？

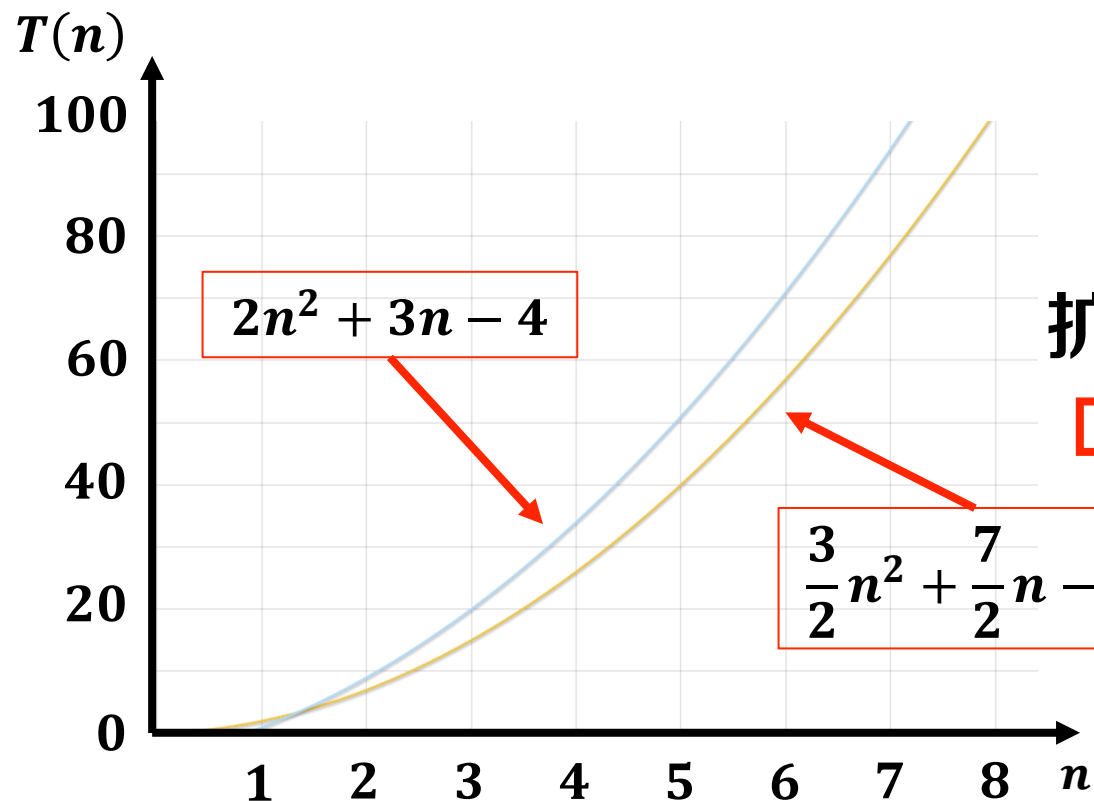




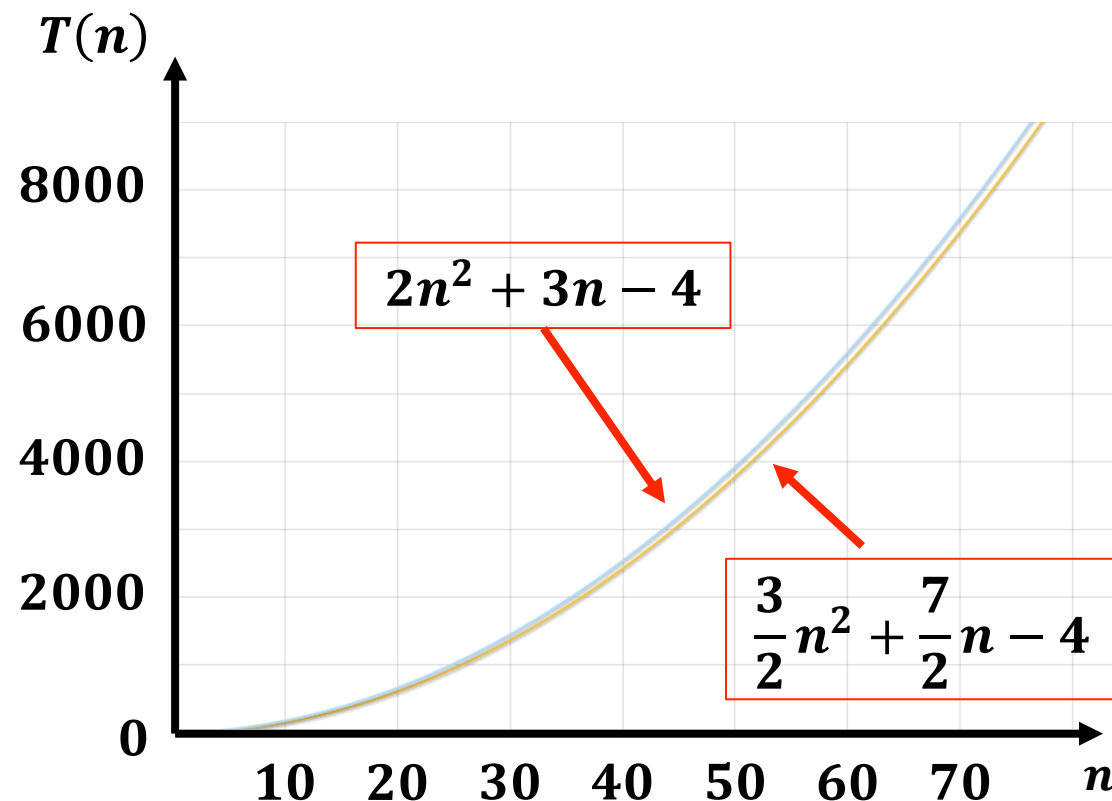
扩展规模



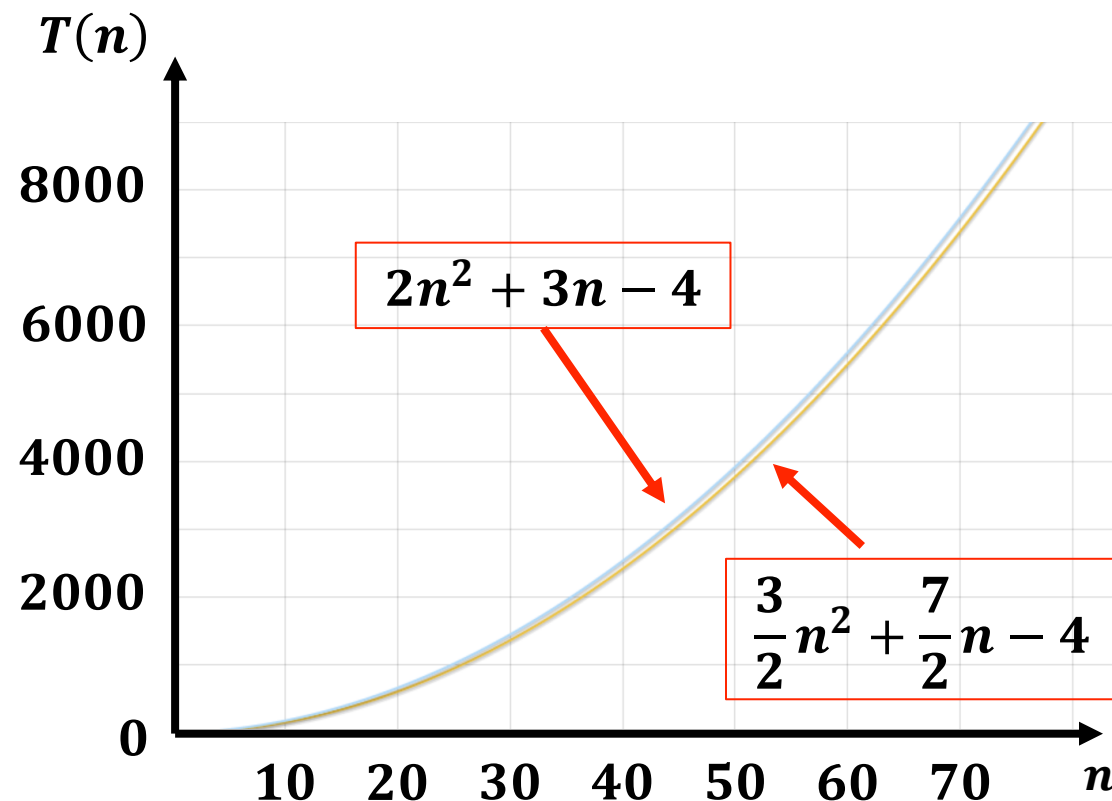
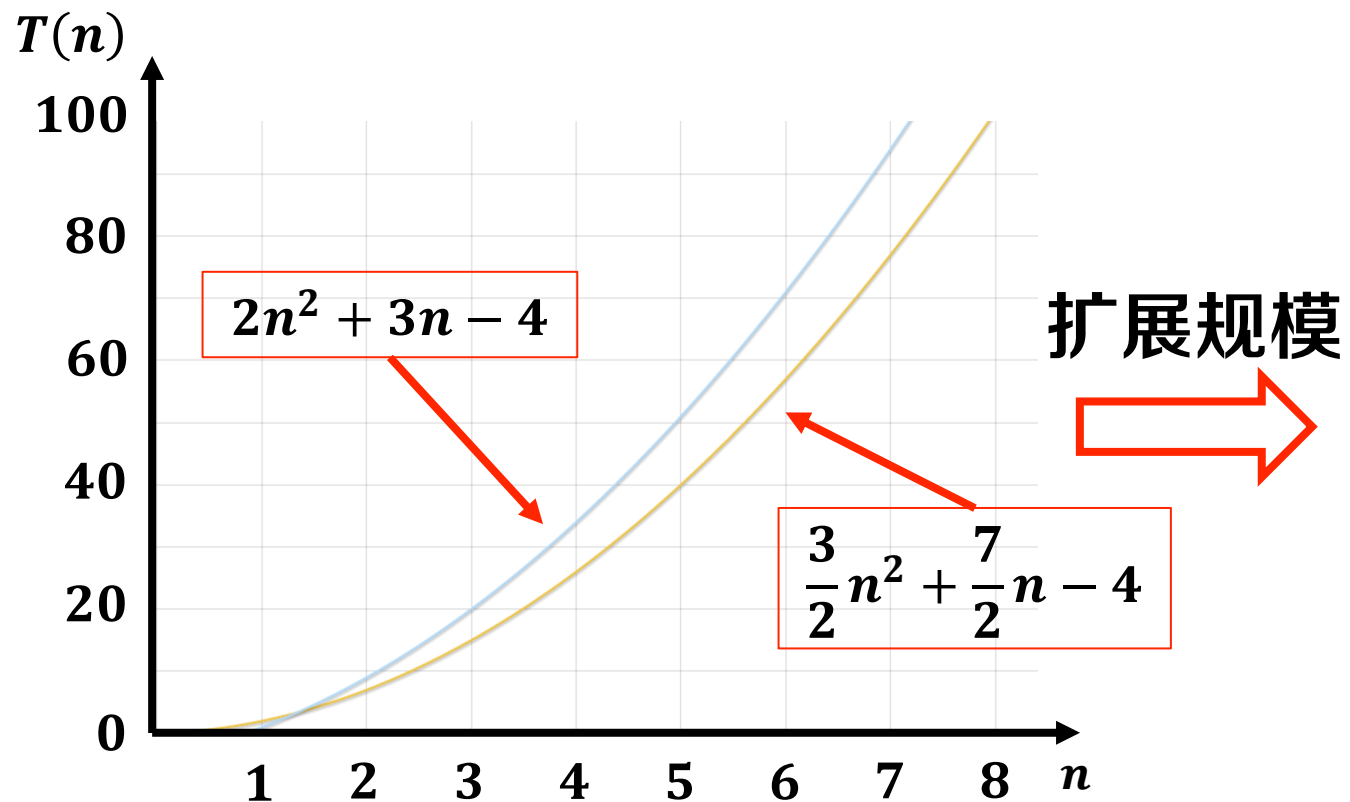
- 在 n 充分大时，两者相差不大



扩展规模



- 在 n 充分大时，两者相差不大
- 原因？



- 在 n 充分大时，两者相差不大
- 原因：两函数的最高阶项相同

- 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

$$T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4$$

- 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
     $cur\_min \leftarrow A[i]$ 
     $cur\_min\_pos \leftarrow i$ 
    for  $j \leftarrow i + 1$  to  $n$  do
        if  $A[j] < cur\_min\_pos$  then
             $cur\_min \leftarrow A[j]$ 
             $cur\_min\_pos \leftarrow j$ 
        end
    end
    交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

$$T(n) = 2n^2 + 3n - 4$$

渐近分析：忽略 $T(n)$ 的系数与低阶项，仅关注高阶项，用记号 Θ 表示

- 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do
     $key \leftarrow A[j]$ 
     $i \leftarrow j - 1$ 
    while  $i > 0$  and  $A[i] > key$  do
         $A[i + 1] \leftarrow A[i]$ 
         $i \leftarrow i - 1$ 
    end
     $A[i + 1] \leftarrow key$ 
end
```

$$T(n) = \Theta(n^2)$$

- 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
     $cur\_min \leftarrow A[i]$ 
     $cur\_min\_pos \leftarrow i$ 
    for  $j \leftarrow i + 1$  to  $n$  do
        if  $A[j] < cur\_min\_pos$  then
             $cur\_min \leftarrow A[j]$ 
             $cur\_min\_pos \leftarrow j$ 
        end
    end
    交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

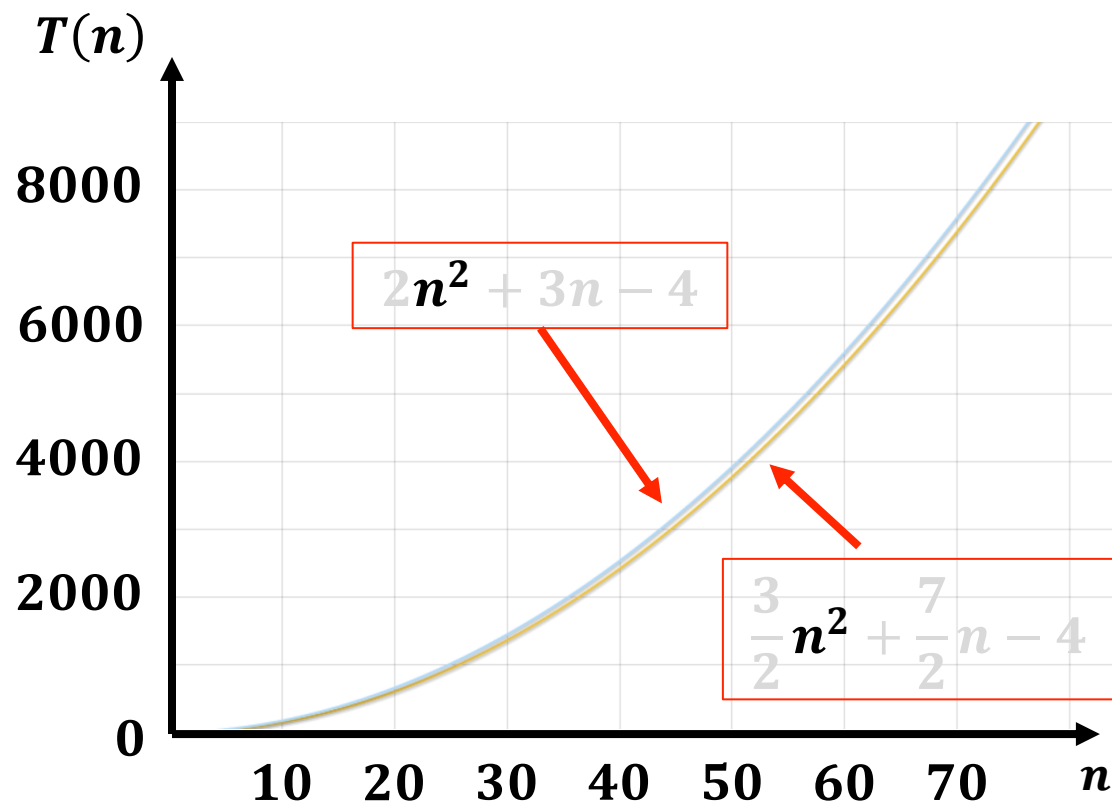
$$T(n) = \Theta(n^2)$$

渐近分析：忽略 $T(n)$ 的系数与低阶项，仅关注高阶项，用记号 Θ 表示

渐近分析



- 分析 n 充分大时函数的大小关系，并用渐近记号表示



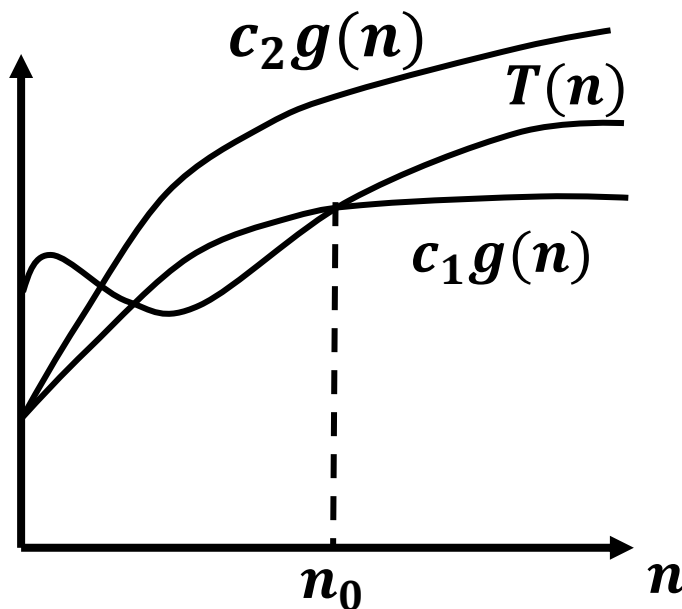
渐近记号	名称
$T(n) = \Theta(g(n))$	渐近紧确界
$T(n) = O(g(n))$	渐近上界
$T(n) = \Omega(g(n))$	渐近下界

Θ 记号

定义：

- 对于给定的函数 $g(n)$ ， $\Theta(g(n))$ 表示以下函数的集合：

$$\Theta(g(n)) = \{T(n) : \exists c_1, c_2, n_0 > 0, \text{使得} \forall n \geq n_0, c_1 g(n) \leq T(n) \leq c_2 g(n)\}$$



$$T(n) = \Theta(g(n))$$

Θ 记号

定义：

- 对于给定的函数 $g(n)$ ， $\Theta(g(n))$ 表示以下函数的集合：

$$\Theta(g(n)) = \{T(n) : \exists c_1, c_2, n_0 > 0, \text{使得} \forall n \geq n_0, c_1 g(n) \leq T(n) \leq c_2 g(n)\}$$

• Θ 记号示例

- $T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4 = ?$

Θ 记号

定义：

- 对于给定的函数 $g(n)$ ， $\Theta(g(n))$ 表示以下函数的集合：

$$\Theta(g(n)) = \{T(n) : \exists c_1, c_2, n_0 > 0, \text{使得} \forall n \geq n_0, c_1 g(n) \leq T(n) \leq c_2 g(n)\}$$

• Θ 记号示例

- $T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4 = ?$
- 令 $n_0 = 2$ ，当 $n \geq n_0$ 时，有
- $\frac{3}{2}n^2 + \frac{7}{2}n - 4 \geq \frac{3}{2}n^2 \geq n^2$

Θ 记号

定义：

- 对于给定的函数 $g(n)$ ， $\Theta(g(n))$ 表示以下函数的集合：

$$\Theta(g(n)) = \{T(n) : \exists c_1, c_2, n_0 > 0, \text{使得} \forall n \geq n_0, c_1 g(n) \leq T(n) \leq c_2 g(n)\}$$

• Θ 记号示例

- $T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4 = ?$
- 令 $n_0 = 2$ ，当 $n \geq n_0$ 时，有
- $\frac{3}{2}n^2 + \frac{7}{2}n - 4 \geq \frac{3}{2}n^2 \geq n^2$
- $\frac{3}{2}n^2 + \frac{7}{2}n - 4 \leq \frac{3}{2}n^2 + \frac{7}{2}n^2 + n^2 = 6n^2$

Θ 记号

定义：

- 对于给定的函数 $g(n)$ ， $\Theta(g(n))$ 表示以下函数的集合：

$$\Theta(g(n)) = \{T(n) : \exists c_1, c_2, n_0 > 0, \text{使得} \forall n \geq n_0, c_1 g(n) \leq T(n) \leq c_2 g(n)\}$$

• Θ 记号示例

- $T(n) = \frac{3}{2}n^2 + \frac{7}{2}n - 4 = \Theta(n^2)$
- 令 $n_0 = 2$ ，当 $n \geq n_0$ 时，有
- $\frac{3}{2}n^2 + \frac{7}{2}n - 4 \geq \frac{3}{2}n^2 \geq n^2$
- $\frac{3}{2}n^2 + \frac{7}{2}n - 4 \leq \frac{3}{2}n^2 + \frac{7}{2}n^2 + n^2 = 6n^2$
- 故存在 $c_1 = 1, c_2 = 6, n_0 = 2$ ，使得 $\forall n \geq n_0, c_1 n^2 \leq T(n) \leq c_2 n^2$

- Θ 记号示例

- $\frac{3}{2}n^5 + \frac{7}{2}n - 10 =$

- Θ 记号示例

- $\frac{3}{2}n^5 + \frac{7}{2}n - 10 = \Theta(n^5)$

- Θ 记号示例

- $\frac{3}{2}n^5 + \frac{7}{2}n - 10 = \Theta(n^5)$

- $n^3 - n^2 + n =$

- Θ 记号示例

- $\frac{3}{2}n^5 + \frac{7}{2}n - 10 = \Theta(n^5)$

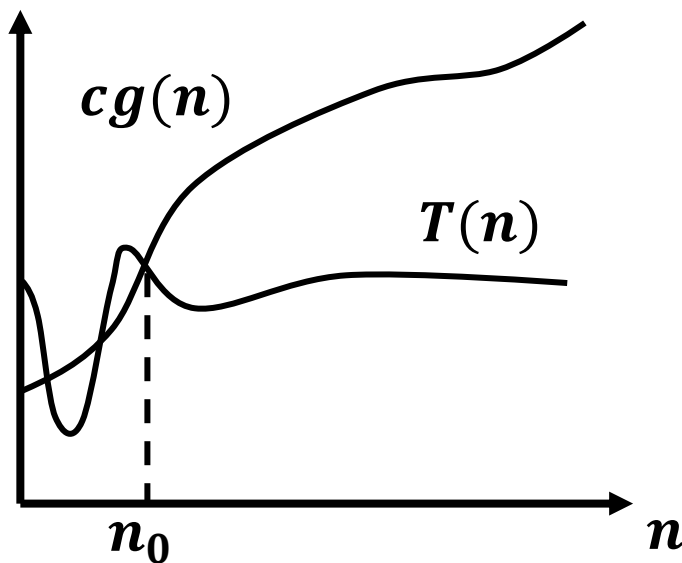
- $n^3 - n^2 + n = \Theta(n^3)$

O 记号

定义：

- 对于给定的函数 $g(n)$ ， $O(g(n))$ 表示以下函数的集合：

$$O(g(n)) = \{T(n) : \exists c, n_0 > 0, \text{使得} \forall n \geq n_0, 0 \leq T(n) \leq cg(n)\}$$

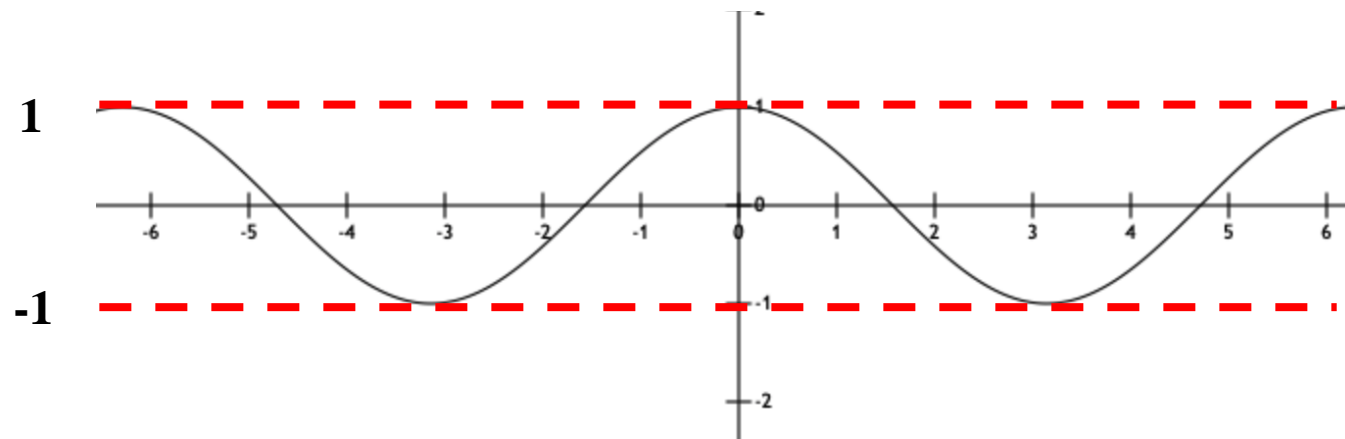


$$T(n) = O(g(n))$$

- O 记号示例
 - $\cos n$

- O 记号示例

- $\cos n \leq 1$



- O 记号示例
 - $\cos n = O(1)$

- O 记号示例
 - $\cos n = O(1)$
 - $\frac{n^2}{2} - 12n =$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n =$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} =$

对数换底公式

$$\log_x N = \frac{\log_y N}{\log_y x}$$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

对数换底公式

$$\log_x N = \frac{\log_y N}{\log_y x}$$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ （假设 n 是 2 的整数幂）

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \cdots + \frac{1}{n}$$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$\begin{aligned} &= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n} \\ &< \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \dots + \frac{1}{\frac{n}{2}} + \frac{1}{n} \end{aligned}$$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$\begin{aligned} &= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n} \\ &< \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \dots + \frac{1}{\frac{n}{2}} + \frac{1}{n} \end{aligned}$$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n}$$

$$< \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \dots + \frac{1}{\frac{n}{2}} + \frac{1}{n}$$

$$= \frac{1}{1} + 2 \cdot \left(\frac{1}{2}\right) + 4 \cdot \left(\frac{1}{4}\right) + 8 \cdot \left(\frac{1}{8}\right) + \dots + \frac{n}{2} \left(\frac{1}{\frac{n}{2}}\right) + \frac{1}{n}$$

- O 记号示例

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n}$$

$$< \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \dots + \frac{1}{\frac{n}{2}} + \frac{1}{n}$$

$$= \frac{1}{1} + 2 \cdot \left(\frac{1}{2}\right) + 4 \cdot \left(\frac{1}{4}\right) + 8 \cdot \left(\frac{1}{8}\right) + \dots + \frac{n}{2} \left(\frac{1}{\frac{n}{2}}\right) + \frac{1}{n}$$

$\log n$ 项

- **O 记号示例**

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n}$$

$$< \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \dots + \frac{1}{\frac{n}{2}} + \frac{1}{n}$$

$$= \frac{1}{1} + 2 \cdot \left(\frac{1}{2}\right) + 4 \cdot \left(\frac{1}{4}\right) + 8 \cdot \left(\frac{1}{8}\right) + \dots + \frac{n}{2} \left(\frac{1}{\frac{n}{2}}\right) + \frac{1}{n} = \frac{1}{n} + \sum_{j=0}^{\log n - 1} 1$$

- **O 记号示例**

- $\cos n = O(1)$

- $\frac{n^2}{2} - 12n = O(n^2)$

- $\log_7 n = \frac{\log_2 n}{\log_2 7} = O(\log_2 n) = O(\log n)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n}$$

$$< \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \dots + \frac{1}{\frac{n}{2}} + \frac{1}{n}$$

$$= \frac{1}{1} + 2 \cdot \left(\frac{1}{2}\right) + 4 \cdot \left(\frac{1}{4}\right) + 8 \cdot \left(\frac{1}{8}\right) + \dots + \frac{n}{2} \left(\frac{1}{\frac{n}{2}}\right) + \frac{1}{n} = \frac{1}{n} + \sum_{j=0}^{\log n - 1} 1$$

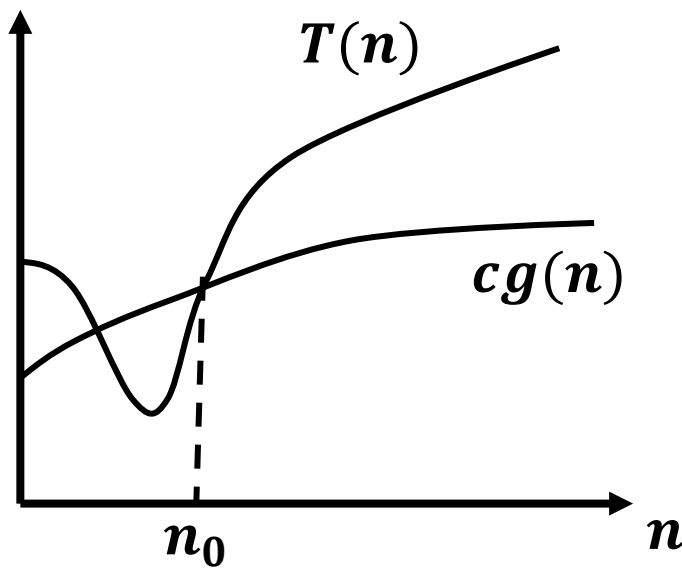
$$= \log n + \frac{1}{n} = O(\log n)$$

Ω 记号

定义：

- 对于给定的函数 $g(n)$ ， $\Omega(g(n))$ 表示以下函数的集合：

$$\Omega(g(n)) = \{T(n) : \exists c, n_0 > 0, \text{使得} \forall n \geq n_0, 0 \leq cg(n) \leq T(n)\}$$



$$T(n) = \Omega(g(n))$$

- Ω 记号示例
 - $n^3 - 2n =$

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- $n^2 + n = \Omega(n^2)$

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$
- $n^2 + n = \Omega(n^2)$
- $\sum_{i=1}^n \frac{1}{i}$ （假设 n 是2的整数幂）

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- $n^2 + n = \Omega(n^2)$

- $\sum_{i=1}^n \frac{1}{i}$ （假设 n 是 2 的整数幂）

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \cdots + \frac{1}{n}$$

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- $n^2 + n = \Omega(n^2)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$\begin{aligned} &= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \cdots + \frac{1}{n} \\ &> \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16} + \cdots + \frac{1}{n} \end{aligned}$$

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- $n^2 + n = \Omega(n^2)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n}$$

$$> \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16} + \dots + \frac{1}{n}$$

$$= \frac{1}{1} + \frac{1}{2} + 2 \cdot \left(\frac{1}{4}\right) + 4 \cdot \left(\frac{1}{8}\right) + 8 \cdot \left(\frac{1}{16}\right) + \dots + \frac{n}{2} \left(\frac{1}{n}\right)$$



$\log n$ 项

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- $n^2 + n = \Omega(n^2)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是2的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \cdots + \frac{1}{n}$$

$$> \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16} + \cdots + \frac{1}{n}$$

$$= \frac{1}{1} + \frac{1}{2} + 2 \cdot \left(\frac{1}{4}\right) + 4 \cdot \left(\frac{1}{8}\right) + 8 \cdot \left(\frac{1}{16}\right) + \cdots + \frac{n}{2} \left(\frac{1}{n}\right)$$

$$= 1 + \sum_{j=1}^{\log n} \frac{1}{2}$$

- Ω 记号示例

- $n^3 - 2n = \Omega(n^3)$

- $n^2 + n = \Omega(n^2)$

- $\sum_{i=1}^n \frac{1}{i}$ (假设 n 是 2 的整数幂)

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \dots + \frac{1}{n}$$

$$> \frac{1}{1} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16} + \dots + \frac{1}{n}$$

$$= \frac{1}{1} + \frac{1}{2} + 2 \cdot \left(\frac{1}{4}\right) + 4 \cdot \left(\frac{1}{8}\right) + 8 \cdot \left(\frac{1}{16}\right) + \dots + \frac{n}{2} \left(\frac{1}{n}\right)$$

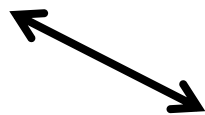
$$= 1 + \sum_{j=1}^{\log n} \frac{1}{2}$$

$$= 1 + \frac{1}{2} \log n$$

$$= \Omega(\log n)$$

- $T(n) = \Theta(g(n))$ 等价于: $T(n) = \Omega(g(n))$ 且 $T(n) = O(g(n))$

渐近记号	名称
Θ	渐近紧确界
O	渐近上界
Ω	渐近下界



输入情况	情况说明
最好情况	不常出现, 不具普遍性
最坏情况	确定上界, 更具一般性

算法运行时间称为算法的时间复杂度, 通常使用渐近记号 O 表示

- 插入排序最坏情况

```
for  $j \leftarrow 2$  to  $n$  do
   $key \leftarrow A[j]$ 
   $i \leftarrow j - 1$ 
  while  $i > 0$  and  $A[i] > key$  do
     $A[i + 1] \leftarrow A[i]$ 
     $i \leftarrow i - 1$ 
  end
   $A[i + 1] \leftarrow key$ 
end
```

$O(n)$ $O(n^2)$

- 选择排序最坏情况

```
for  $i \leftarrow 1$  to  $n - 1$  do
   $cur\_min \leftarrow A[i]$ 
   $cur\_min\_pos \leftarrow i$ 
  for  $j \leftarrow i + 1$  to  $n$  do
    if  $A[j] < cur\_min\_pos$  then
       $cur\_min \leftarrow A[j]$ 
       $cur\_min\_pos \leftarrow j$ 
    end
  end
  交换  $A[i]$  和  $A[cur\_min\_pos]$ 
end
```

$O(n)$ $O(n^2)$

- 算法分析的原则

+	-	×	÷
:=	>	<	=



统一机器性能

算法的分析小结

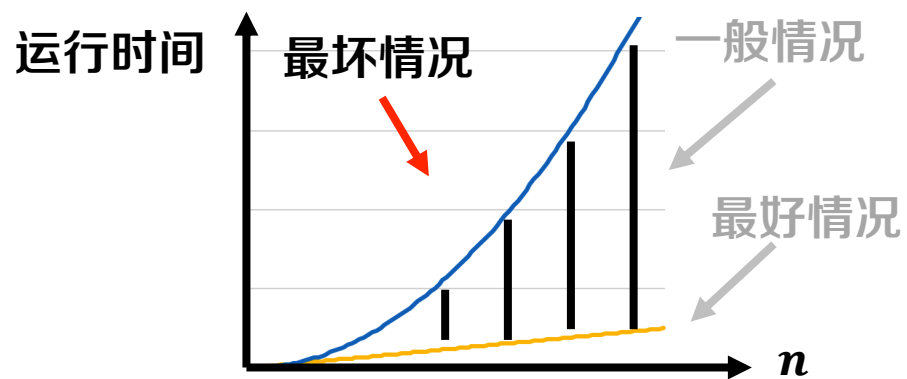


- 算法分析的原则

+	-	×	÷
:=	>	<	=



统一机器性能



分析最坏情况

算法的分析小结

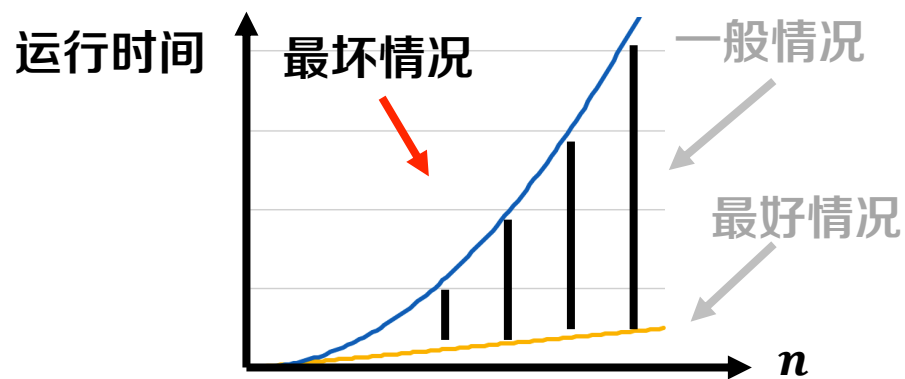


- 算法分析的原则

+	-	×	÷
:=	>	<	=

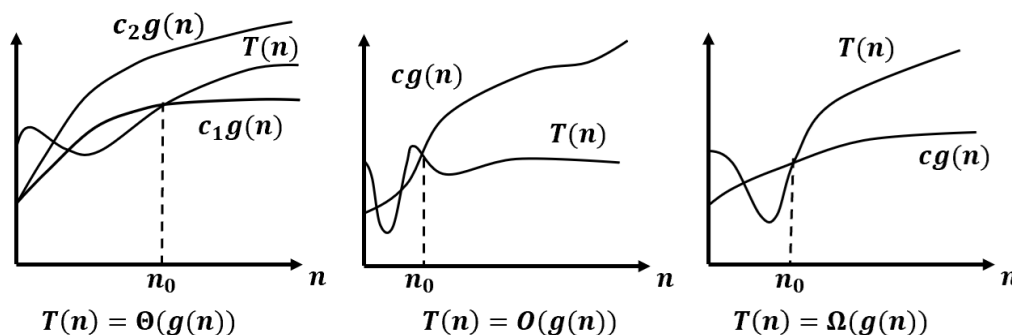


统一机器性能



分析最坏情况

- 算法分析的工具



采用渐近分析

算法的由来

算法的定义

算法的性质

算法的表示

算法的分析

算法设计与分析

分而治之篇

归并排序

递归式求解

最大子数组I

逆序对计数

快速排序

次序选择

动态规划篇

0-1背包

最大子数组II

最长公共子序列

最长公共子串

编辑距离

钢条切割

矩阵链乘法

贪心策略篇

部分背包

霍夫曼编码

任务调度

活动选择

最小生成树

单源最短路径

图算法篇

广度优先搜索

深度优先搜索

环路存在性判断

拓扑排序

强连通分量

全点对最短路径

二分图匹配