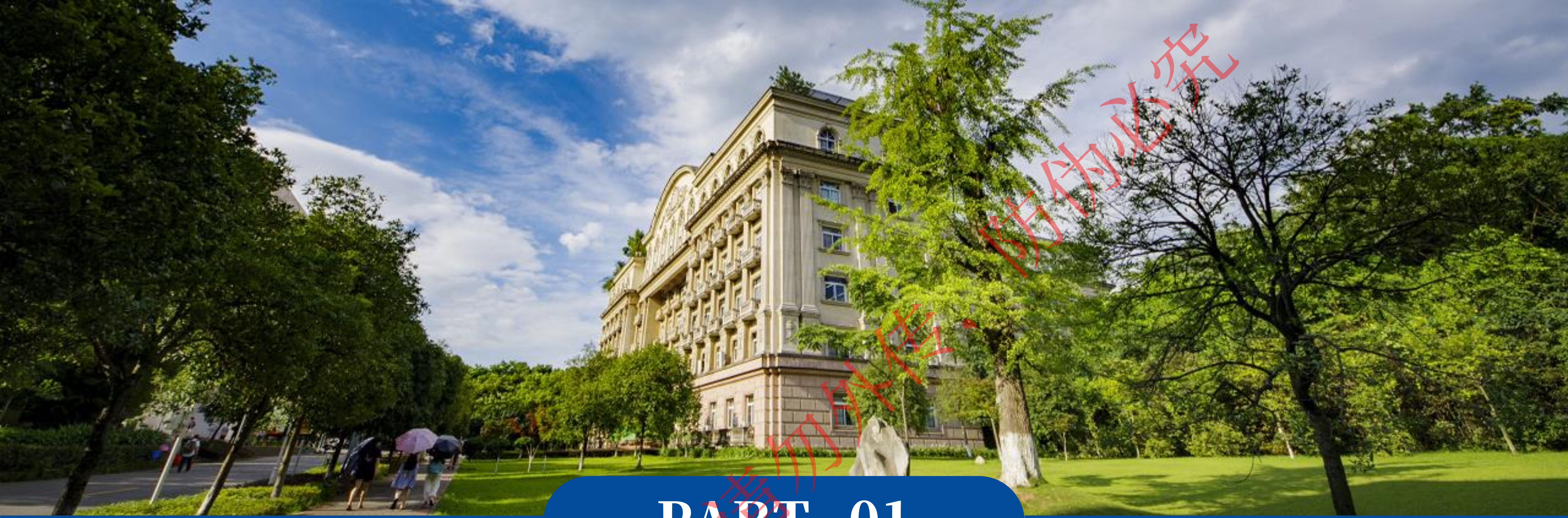
芯片测试

张里博
lbzhang@swu.edu.cn

目录



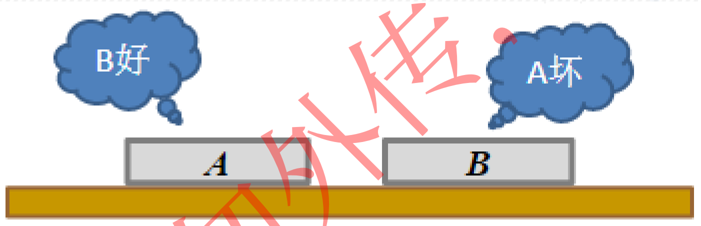
PART 01

问题介绍

例2.1 芯片测试

■**已知：**有n片芯片，好芯片至少比坏芯片多1片

■**问题：**设计算法，使用尽可能少的测试次数从中挑出1片好芯片.

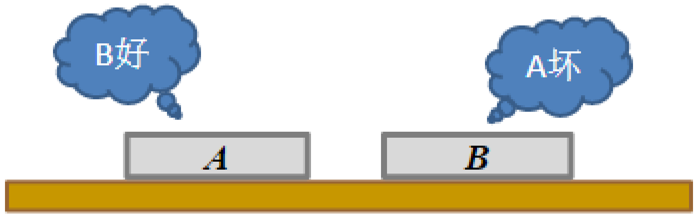


■将2片芯片置于测试台上，互相进行测试，测试结果为“好”或者“坏”。好芯片的报告一定是正确的，坏芯片的报告不确定（可能会出错）

■可以利用的两个条件：

■**1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确；**

■**蛮力算法：** 任取1片测试，如果是好芯片，测试结束；如果是坏芯片，抛弃，再从剩下芯片中任取1片测试，直到得到1片好芯片



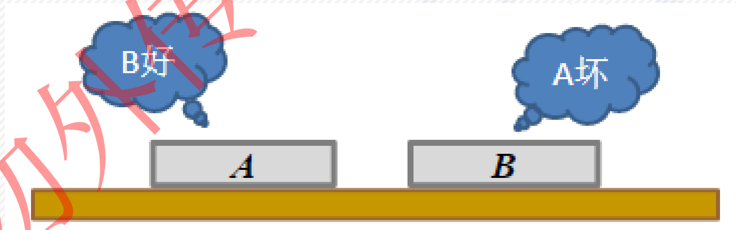
■**如何判断被测芯片是好芯片呢？**

■根据剩余所有芯片轮流、依次对被选芯片测试，根据剩余所有芯片的报告，判断被测芯片真假。

■**芯片报告满足什么条件时，被测芯片是好芯片？**

算法设计：如何根据芯片报告的情况判断被测芯片的真实情况？

在具体问题中，已知芯片真实组成情况下，分析报告的情况，据此进行算法规则设计



单次测试结果分析

	*A*是好的	*A*是坏的
*B*是好的	A、B都报告好	A报告B好或者坏，B报告A坏
*B*是坏的	A报告B坏，B报告A好或者坏	A报告B好或者坏，B报告A好或者坏

PART 02

芯片测试分治策略

■要求（需满足的条件）：

■1. 子问题与原始问题的性质完全一样；

■2. 子问题之间可彼此独立地求解；

■3. 递归或迭代停止时最小规模子问题可直接求解.

■可以利用的两个条件：

■1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确；

■最小规模子问题，要如何设计算法？ n=1，2，3

1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确

- 从**最小规模问题**分析开始：
- 若只有1片或2片芯片，则全为好芯片；
- 若有3片芯片，如何设计规则，从中挑出1块好芯片？

3片全是好芯片

2片好芯片，1片坏芯片

任意抽取2块芯片

2好

2好

1好1坏

如何根据芯片报告情况，判断被测芯片的真假？

分治策略

1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确

	A是好的	A是坏的
B是好的	A、B都报告好	A报告B好或者坏，B报告A坏
B是坏的	A报告B坏，B报告A好或者坏	A报告B好或者坏，B报告A好或者坏

n=3

任意抽取2块芯片

3片全是好芯片 → 2好 → 报告2好

2片好，1片坏芯片 → 2好；1好1坏

1好1坏 → 报告2坏；报告1好1坏

报告2好　报告2坏　报告1好1坏

教材P30

```
if  k ==3    then
        任取2片芯片测试;
        if  2片都报好  then
                任取1片被测芯片;
        else
                取没被测芯片;
        end
end
if  k ==2 or 1 then
        任取1片 ;
end
```

教材P30

这个地方对么？

芯片的真假可知么？

```
if  k ==3    then
        任取2片芯片测试;
        if  1好1坏  then
                取没测的芯片;
        else
                任取1片被测芯片;
        end
end
if  k ==2 or 1 then
        任取1片;
end
```

1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确

- 最小子问题可以求解：$n \leqslant 3$
- 1片或2片芯片，则全为好芯片；
- 3片芯片，一次测试即可得到好芯片；

$n>3$时如何设计算法？

目标：保持问题性质（好芯片比坏芯片多）的前提下，问题规模尽可能快速地减小

$n=3$时的筛选规则：报告全好，留一片；其他情况，全丢弃；

两两分组，按照这个筛选规则，$n>3$时会出现什么情况？

例2.1 芯片测试

	*A*是好的	*A*是坏的
*B*是好的	A、B都报告好	A报告B好或者坏，B报告A坏
*B*是坏的	A报告B坏，B报告A好或者坏	A报告B好或者坏，B报告A好或者坏

A 报告	*B* 报告	结论
*B*是好的	*A*是好的	*A*,*B* 都好或 *A*,*B* 都坏
*B*是好的	*A*是坏的	至少一片是坏的
*B*是坏的	*A*是好的	至少一片是坏的
*B*是坏的	*A*是坏的	至少一片是坏的

情况一，丢一片留一片；

其他情况，全丢弃；

剩余芯片进入下一轮筛选

子问题与原问题是否性质相同？

1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确

淘汰前芯片总数是偶数，子问题是否与原问题性质相同？

命题2.1 当n是偶数时，在上述规则下，经过一轮淘汰，剩下的好芯片比坏芯片至少多1片.

证 设A与B都是好芯片有 i 组，A与B一好一坏有 j 组，A与B都坏有 k 组。

$$2i + 2j + 2k = n;$$

初始好芯片数多于坏芯片 $2i+j > 2k+j$; $\Rightarrow i > k$

一轮淘汰后好芯片有i片，坏芯片至多 k（即≤ k） 片。因此，原问题的性质得到保持。

含弘光大 继往开来

至少淘汰一半芯片

1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确

淘汰前芯片总数是奇数，子问题是否与原问题性质相同？

输入：好 好 好 坏 坏 坏 好

分组1：[好 好] [好 坏] [坏 坏] 好

淘汰后：好 坏 好

分组2：[好 好] [好 坏] [坏 好] 坏

淘汰后：好 坏

分组3：[好 好] [好 好] [坏 坏] 坏

淘汰后：好 好 坏 坏

分治策略

1. 好芯片比坏芯片多；2. 好芯片的测试报告一定准确

输入：好 好 好 好 坏 坏 坏

分组：[好 好] [好 坏] [好 坏] 坏

淘汰后：好 坏

- 处理办法：当n是奇数时，增加一轮对轮空芯片的单独测试（**蛮力算法**）。
- 如果该芯片为好芯片，则算法结束；如果是坏芯片，删除该芯片，是否能继续分组淘汰？
- 是否是偶数片芯片？是否好>坏？

```
k ← n;
while  k >3  do
      将芯片分成⌊k/2⌋ 组;      // 如有轮空芯片，特殊处理
      for  i ← 1  to ⌊k/2⌋  do
            if   2片都报好   then
                  任取1片留下;
            else
                  2片同时丢掉;
            end
      end
      k ←剩下的芯片数;
end
if  k ==3    then
      任取2片芯片测试;
      if 2片都报好  then
            任取1片被测芯片;
      else
            取没被测芯片;
      end
end
if  k ==2 or 1 then
      任取1片;
end
```

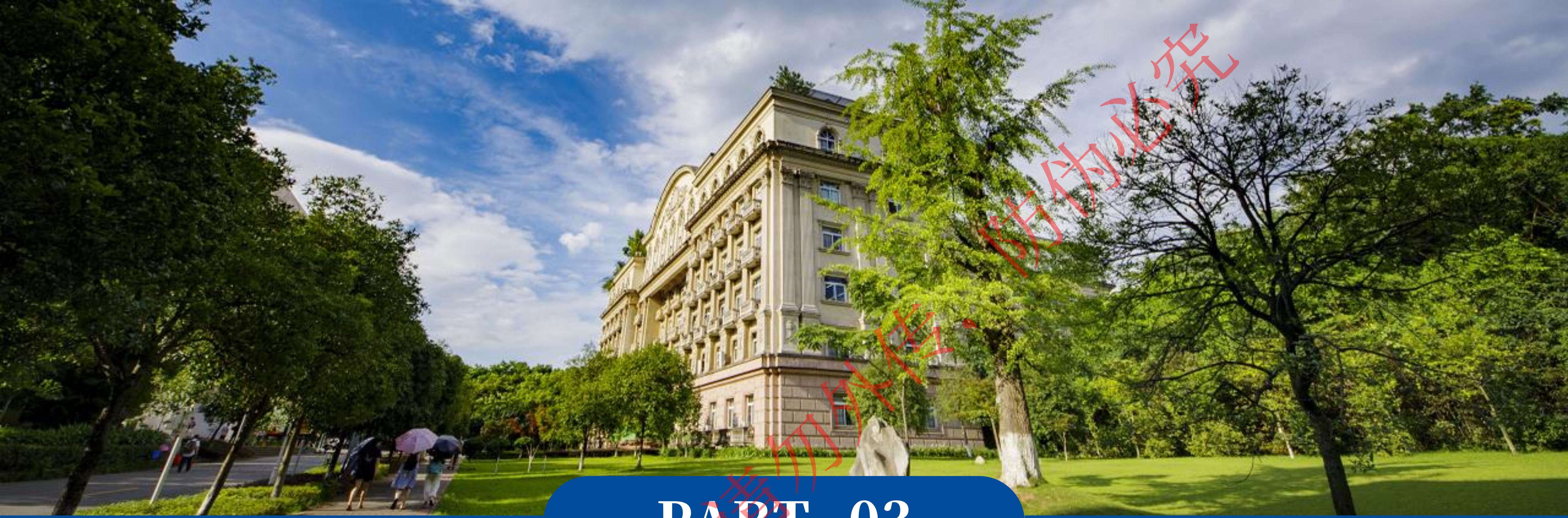
```
k ← n;
while  k >3  do
      将芯片分成⌊k/2⌋ 组;      // 如有轮空芯片，特殊处理
      for  i ← 1  to ⌊k/2⌋  do
            if  2片好  then
                  任取1片留下；
            else
                  2片同时丢掉;
            end
      end
      k ←剩下的芯片数;
end
if  k ==3    then
      任取2片芯片测试;
      if  1好1坏  then
            取没测的芯片;
      else
            任取1片被测芯片;
      end
end
if  k ==2 or 1 then
      任取1片；
end
```

```
k ← n;
while  k >3  do
      if  k是奇数  then
            任取1片芯片作为轮空芯片；
            采用蛮力算法对轮空芯片进行判断
             if 轮空芯片为真 then       //采用蛮力算法
                 return 轮空芯片
             end
            k  ← k – 1；删除轮空芯片；
      end
      将芯片分成 k/2  组;
      for  i ← 1  to  k/2  do
            if   2片都报好   then
                  任取1片留下；
             else
                  2片同时丢掉;
            end
      end
      k ←剩下的芯片数;
end
if  k ==3    then
      任取2片芯片测试;
      if 2片都报好  then
            任取1片被测芯片;
      else
            取没被测芯片;
      end
end
if  k ==2 or 1 then
      任取1片；
end
```

例2.1 芯片测试

- 递归截至条件：$n \le 3$
- 3片芯片，一次测试即可得到好芯片；
- 1片或2片芯片，则全为好芯片；
- 时间复杂度的递推式：
  - W(n) = W(n/2) + O(n-1+(n-1)/2)　　n>3
  - W(n) = 1　　n≤3

$$\begin{cases} W(n) = W(\frac{n}{2}) + O(n) & n > 3 \\ W(n) = 1 & n \le 3 \end{cases} \Rightarrow W(n) = O(n)$$



PART 03

改进的CW算法

轮空芯片的处理

一次至少淘汰一半芯片

- 当芯片总数是奇数，除轮空芯片（剩余偶数片芯片），可以在分组淘汰后，利用蛮力算法对轮空芯片进行测试么？

- 子问题1：当芯片总数是奇数，分组淘汰后，能否根据芯片的报告来判断轮空芯片的好坏么？
- 若轮空芯片为好，**return**；否则：
- 子问题2：如果轮空芯片为坏，子问题（分组淘汰后的芯片）是否仍与原问题性质相同（好>坏）？

- 如何实现：
- Idea：假设已知芯片真假，得出报告情况；然后，根据报告情况判断芯片真假情况

分组淘汰后，蛮力算法能行不？

（书上29页 命题2.1）

- 轮空芯片真坏———>剩余芯片真好>真坏———>分组淘汰后：真好>真坏———>报告：好<坏
- 轮空芯片真好———>剩余芯片真好≥真坏（可分>和=讨论）
- 轮空芯片真好———>剩余芯片真好>真坏———>分组淘汰后：真好>真坏———>报告：好>坏
- 轮空芯片真好———>剩余芯片真好=真坏———>分组淘汰后？

命题2.1（增）

命题2.1（增） 当n是偶数时，如果好芯片与坏芯片一样多，经过一轮淘汰后，剩下的好芯片不会比坏芯片少.

证 设A与B都是好芯片有 i 组，A与B一好一坏有 j 组，A与B都坏有 k 组。

$$2i + 2j + 2k = n;$$

$$2i+j = 2k+j;$$ （初始好芯片数等于坏芯片） $\Rightarrow i = k$

一轮淘汰后好芯片有 i片，坏芯片至多 k（即≤ k）片。因此，剩下的**好芯片个数≥坏芯片个数**。

具体分析过程

- 轮空芯片真坏——>剩余芯片真好>真坏——>分组淘汰后：真好>真坏——>报告：好<坏

- 轮空芯片真好——>剩余芯片真好>真坏——>分组淘汰后：真好>真坏——>报告：好>坏
- 轮空芯片真好——>剩余芯片真好=真坏——>分组淘汰后：真好≥真坏——>报告：好≥坏

- 轮空芯片真好——>剩余芯片真好≥真坏——>分组淘汰后：真好≥真坏——>报告：好≥坏

分析结果

- 轮空芯片真坏——>剩余芯片真好>真坏——>分组淘汰后：真好>真坏——>报告：好<坏
- 轮空芯片真好——>剩余芯片真好≥真坏——>分组淘汰后：真好≥真坏——>报告：好≥坏

- 两种情况下，报告是互斥的。
- 因此子问题1可以解决：可以根据淘汰后芯片对轮空芯片的报告，判断轮空芯片的好坏；

- 如果轮空芯片为坏芯片，淘汰后的芯片：真好>真坏。
- 因此，子问题2可以解决。即如果轮空芯片为坏，淘汰后芯片仍与原问题性质相同。

思考过程（步骤）

- 轮空芯片的情况：真好，或者真坏；
- 分析：两种情况下，剩余芯片的真实好坏情况；
  两种情况下，分组淘汰后剩余芯片的真实好坏情况；
  两种情况下，分组淘汰后测试报告的情况；
- 两种情况下：
- 1.分组淘汰后的测试报告情况是否互斥，互斥则方案可行，否则不可行；
- 2.若轮空芯片为坏芯片，测试和删除后，剩下的芯片中好芯片是否比坏芯片多。

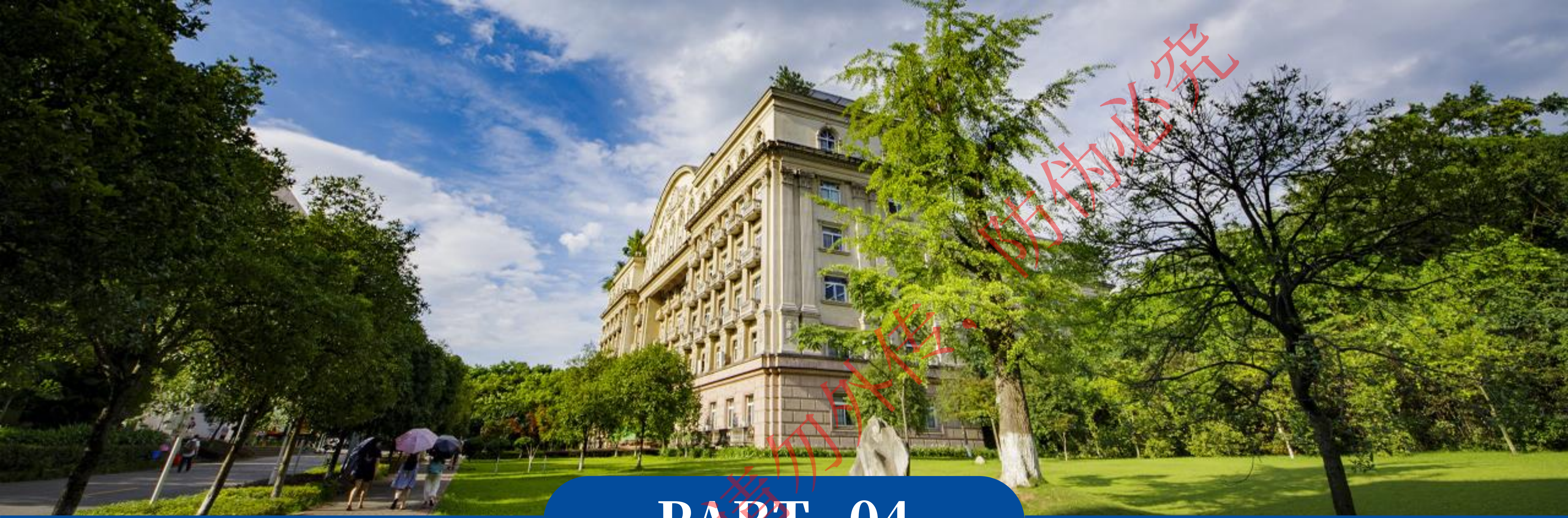
```
k ← n;
while  k >3  do
    if  k是奇数  then
            任取1片芯片作为轮空芯片；
            k  ← k – 1；
    end
    将芯片分成k/2组;
    for  i ← 1  to  k/2 do
        if   2片都报好   then
            任取1片留下；
        else
            2片同时丢掉;
        end
    end
    k_s  ←剩下的芯片数;
    if  轮空芯片 存在  then         //蛮力算法的一次抽取
        if  k_s 个芯片对轮空芯片的报告：好<坏   then
            k ← k_s , 扔掉轮空芯片；
        else
            return 轮空芯片；
        end
    end
end
if  k ==3    then
    任取2片芯片测试;
    if 2片都报好  then
        任取1片被测芯片;
    else
        取没被测芯片;
    end
end
if  k ==2 or 1 then
    任取1片 ;
end
```

例2.1 芯片测试

- 递归截至条件：$n \le 3$
- 3片芯片，一次测试即可得到好芯片；
- 1片或2片芯片，则全为好芯片；
- 时间复杂度的递推式：

• W(n) = W(n/2) + O((n-1)/2+(n-1)/2)　n>3
• W(n) = 1　n≤3

$$\begin{cases} W(n) = W(\frac{n}{2}) + O(n) & n > 3 \\ W(n) = 1 & n \le 3 \end{cases} \Rightarrow W(n) = O(n)$$



PART 04

改进的WZZ算法

算法思想

- 步骤一：当芯片总数为偶数时，扔掉一个（好>坏）；
- 步骤二：当芯片总数为奇数时（好>坏）：

1. 任选一个芯片轮空，将剩余的偶数个芯片**分组淘汰，淘汰后剩余芯片**（不含轮空）**个数为m**；

2. 若m为奇数，扔掉轮空芯片；若m为偶数，则保留轮空芯片；保证芯片总数为奇数；

- 步骤三：重复步骤二，直到只剩下一个芯片，该芯片就是好芯片

需要证明的命题：

在经过该算法的一轮操作后，子问题与原问题性质相同

即仍满足好芯片数量大于坏芯片数量

推导过程

原芯片堆好芯片>坏芯片

芯片堆为偶数 →好芯片--坏芯片>2 →扔掉一块芯片，好芯片--坏芯片>1→好芯片>坏芯片

芯片堆为奇数n → 轮空一枚芯片，剩余好芯片≥坏芯片 → 分组淘汰后有m片，好芯片≥坏芯片

淘汰后芯片为奇数：

淘汰后好芯片至少有(m+1)/2 →好芯片>坏芯片→ 扔掉轮空芯片 → 好芯片>坏芯片

m为偶数：

命题2.1

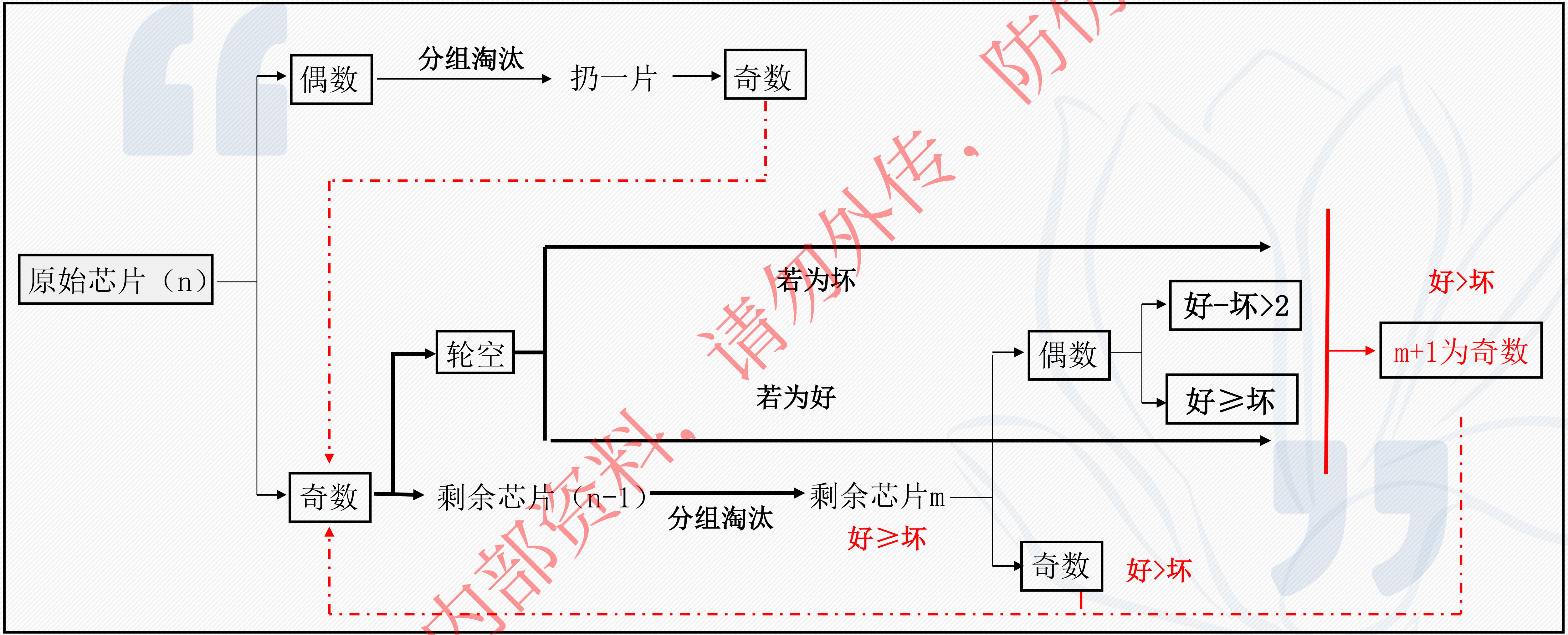
若轮空芯片为坏，剩余n-1块芯片好>坏 → 淘汰后好芯片>坏芯片（总数>0）→淘汰后好芯片比坏芯片至少多两片（好芯片至少有m/2+1） → 拿回轮空芯片（坏） → 好芯片>坏芯片

命题2.1（增）

若轮空芯片为好，剩余n-1块芯片好≥坏 → 淘汰后好芯片≥坏芯片（总数≥0） → 拿回轮空芯片（好） → 好芯片>坏芯片

含弘光大 继往开来

推导过程



原始芯片（n）
偶数
分组淘汰
扔一片
奇数
奇数
轮空
若为坏
若为好
剩余芯片（n-1）
分组淘汰
剩余芯片m
好≥坏
偶数
好-坏>2
好≥坏
奇数
好>坏
好>坏
m+1为奇数

```
Test(n)
    if n为偶数 then
        随机丢掉一片;
        n ← n-1;
    end
    k ← n;
    while k>1 do
        轮空一个芯片
        k ← out(k-1);
        if  k为偶数 then
            拿回轮空芯片
            k ← k+1;
        end
    end
    return 剩下的最后一个芯片
```

迭代实现

输入：n片芯片构成的数组，其中好芯片至少比坏芯片多1片

输出：1片好芯片

```
out(k)    //分组淘汰，k应为偶数
    将芯片分成k/2组
    for i←1 to k/2 do
        if 2片好 then
            任取一片留下
        else
            2片同时丢掉
        end
    end
    k←剩余芯片数;
    return k;
```

分组淘汰的时间复杂度为O(n/2)

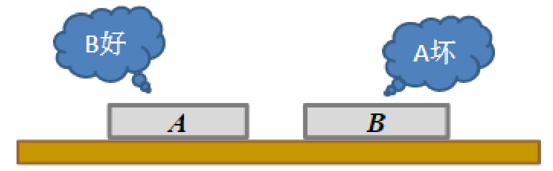
- 书上算法时间复杂度递推方程：
- $W(n) = W(n/2) + O(n-1+(n-1)/2) \quad n>3$
- $W(n) = 1 \quad n\leq 3$
- CW算法时间复杂度递推方程：
- $W(n) = W(n/2) + O((n-1)/2+ (n-1)/2) \quad n>3$
- $W(n) = 1 \quad n\leq 3$
- WZZ算法时间复杂度递推方程：
- $W(n) = W((n-1)/2+1) + O((n-1)/2) \quad n\geq 1$
- $W(n) = 1 \quad n=3$

算法优势分析：最坏情况下的时间复杂度均为O(n)

但容易看出，新算法的实际测试次数比原算法少很多，效率更高

总结

- 问题：从n块芯片（好>坏）中，使用最少测试挑出1片好芯片.
- 工具：好芯片的测试报告一定准确；
- 算法设计思路：在具体问题中，假设已知芯片真实组成情况下，分析报告的情况，据此进行算法规则设计



- 最小规模子问题可如何求解？（ n=1，2，3 ）
- n>3时，算法设计目标？淘汰规则？偶数？奇数？时间复杂度？
- CW算法：起点？具体问题？分析过程？算法设计？时间复杂度？
- WZZ算法：起点?分析过程？算法设计？时间复杂度？