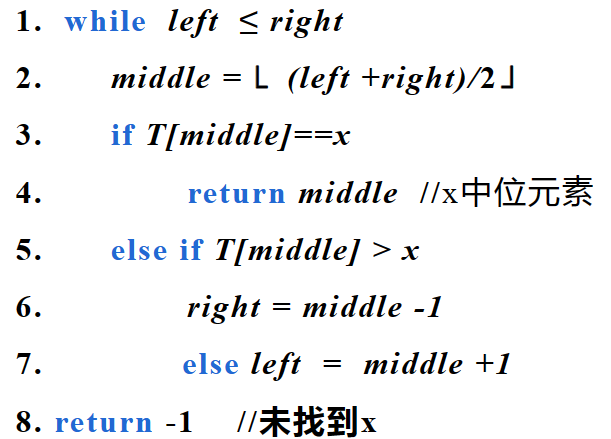
算法满足有限性、确定性、输入和输出

程序是算法用某种语言的具体实现，可以不满足有限性

选择算法描述方法的主要原则：**能够**表现算法的核心内容 **简单**、清楚、准确、无二义性 **与**实现工具无关

最大公约数

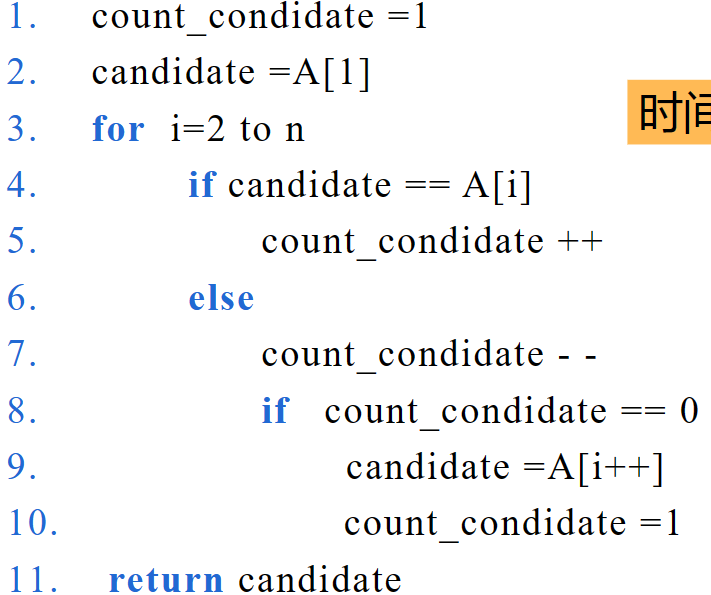
1、while m > 0

2. r = n mod m

3. n = m

4. m = r

5. return n

最大连续子序列之和问题

1. maxSum=0

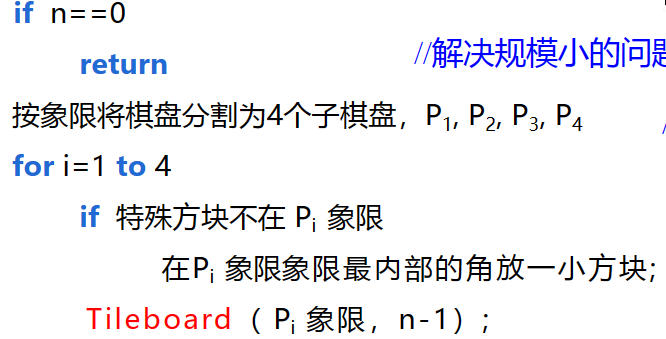
2. thisSum=0

3. i=0

4. for j=0 to n

5. thisSum=thisSum + aj

6. if thisSum > maxSum

7. maxSum=thisSum

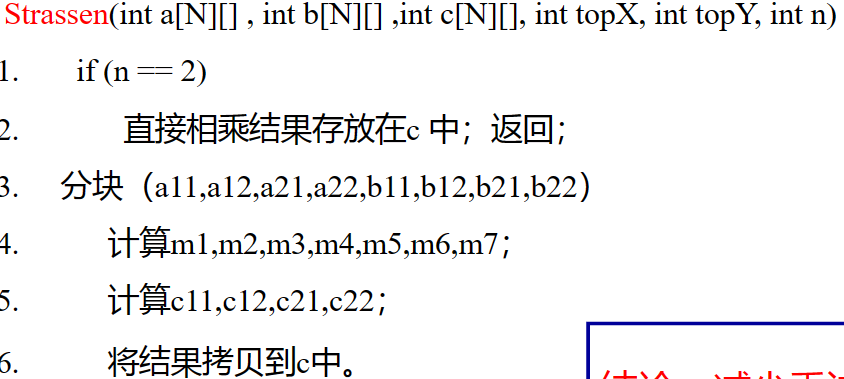
8. else if thisSum < 0

9. i=j+1;

10. thisSum = 0

11. return maxSum

时间复杂度O表示fn<=c\*n,Ω相反、θ为c1\*n<=fn<=c2\*n(n>n0)

归并排序与快速排序

x = q[l+r>>1],i =l-1,j = r + 1;

while(i < j){

do i ++; while(q[i] < x);

do j --; while(q[j] > x);

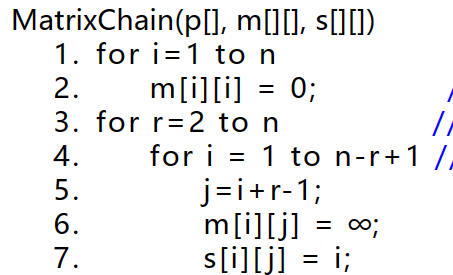
if(i<j)swap(q[i],q[j]);}

while(i <= mid && j <= r)

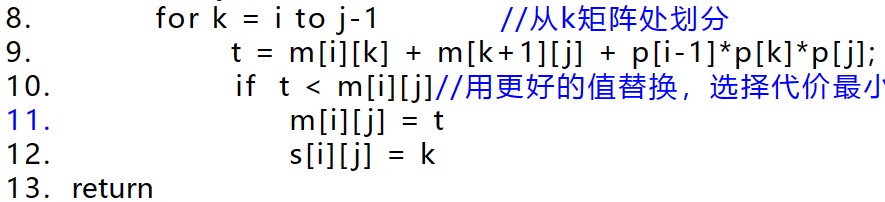
if(q[i] <= q[j]) tmp[k ++] = q[i ++];

else tmp[k ++] = q[j ++];

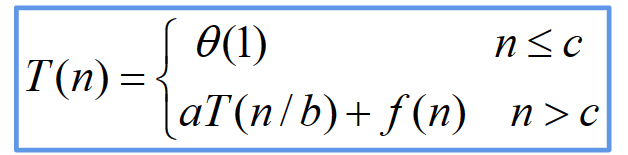
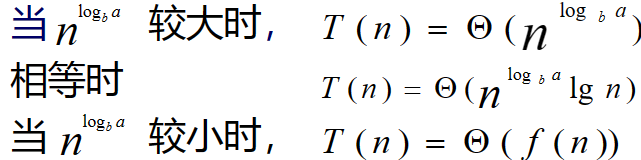
while(i <= mid) tmp[k ++] = q[i ++];

 while(j <= r) tmp[k ++] = q[j ++];

分治算法思想

将一个难以直接解决的规模较大的问题分解为若干个规模较小的子问题，这些子问题互相独立且与原问题相同。递归地解决这些子问题。然后将各个子问题的解合并得到原问题的解

时间复杂度分析方法

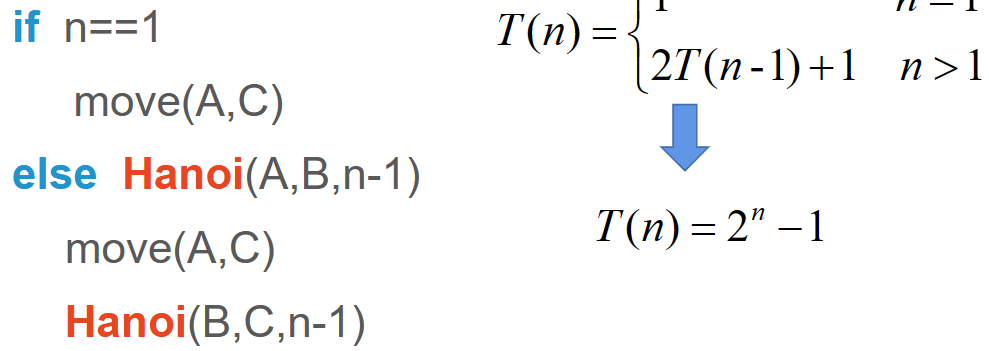
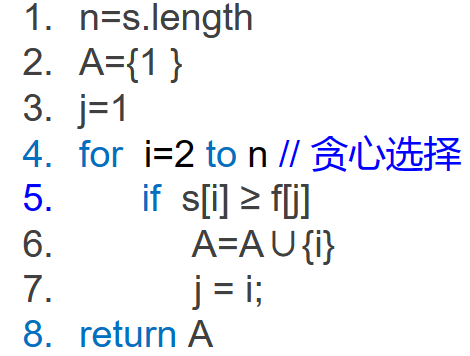
换名法与主方法

例题:

二分与多数元素（同归于尽法）棋盘覆盖

复杂度为4的n次方

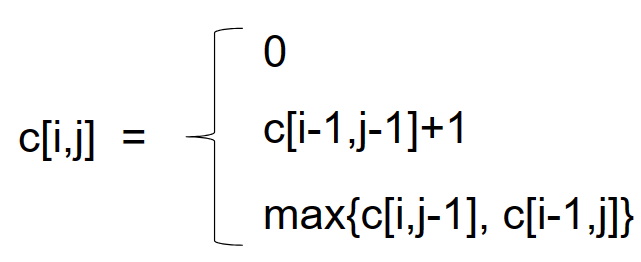
矩阵乘法的分治

汉诺塔的递归

矩阵连乘（动态规划）

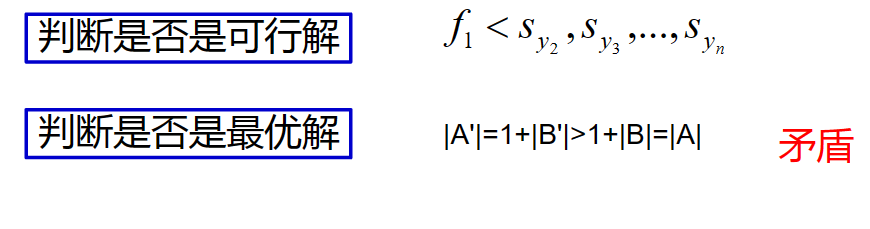
m为最小乘法次数

复杂度为On方

最长公共子序列

最长上升子序列

状态表示为f[i]，表示从第一个数字开始，以w[i]结尾的最大的上升序列。

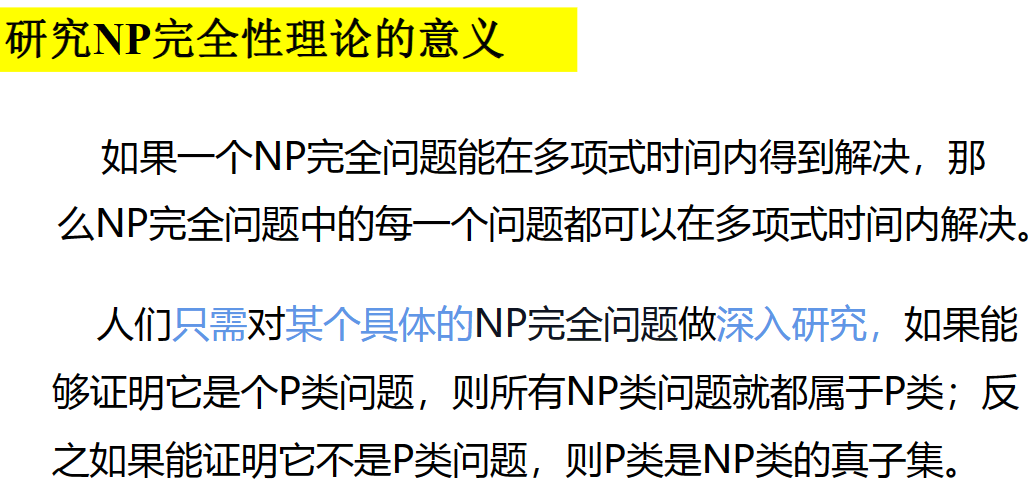
状态计算：当w[i]>w[j]时

f[i] = max(f[i],f[j] + 1)

边界条件为当前面没有比w[i]小的时f[i]=1，时间复杂度为On方

int mx = 1;

for(int i = 0;i < n;i ++)

{

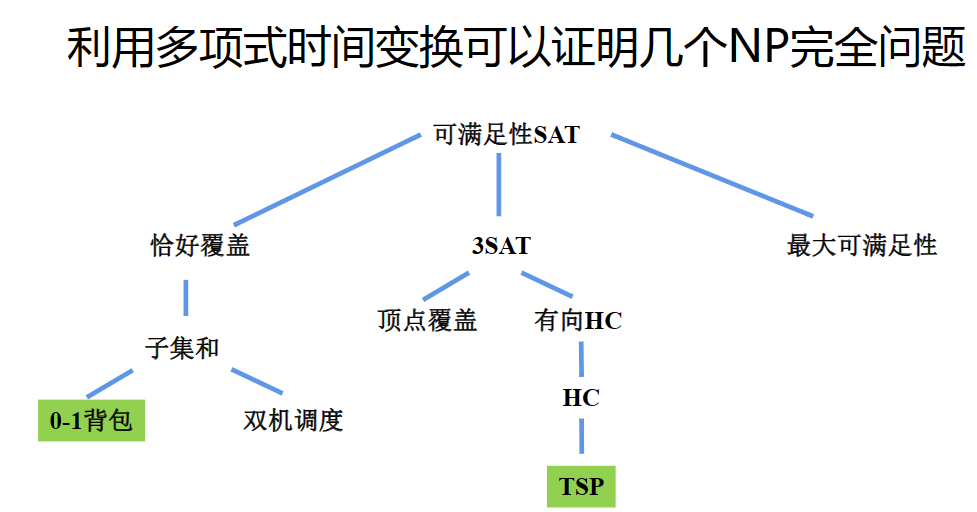
f[i] = 1;

for(int j = 0;j < i;j ++)

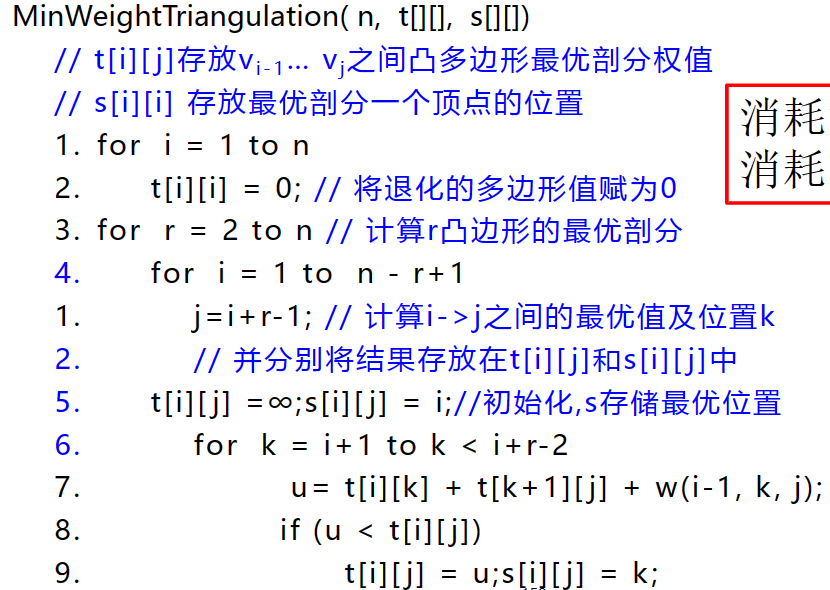
if(w[i] > w[j])

f[i]=max(f[i],f[j] + 1);

mx = max(mx, f[i]);

}、

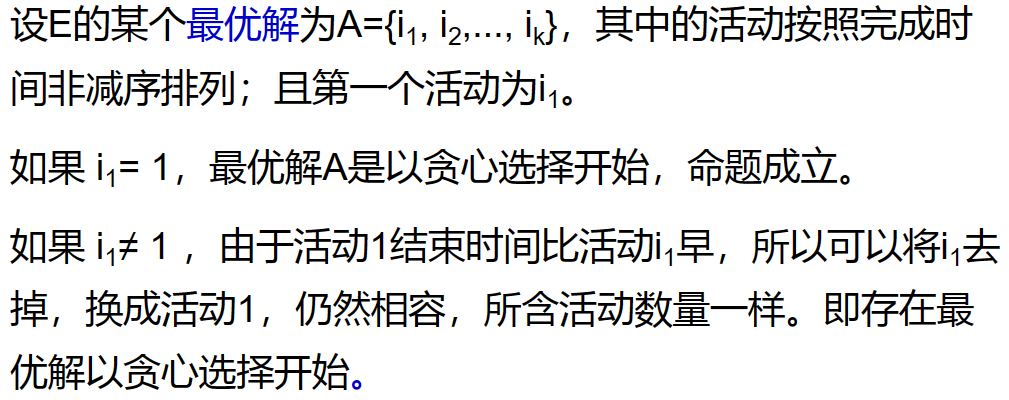
凸多边形最优三角划分

定义t[i][j]为最优值，则由最优子结构性质t[i][j]的值应为t[i][k]的值加上t[k+1][j]的值，再加上三角形 的权值

活动安排问题（贪心）

n个活动要使用场地，时间是s[i]和f[i]，若两个活动时间不相交则相互兼容，找出最大的兼容的活动集合。

贪心的证明

一、证明具有狭义贪心选择性质

二、具有最优子结构

假设B=(x2,...,xk)不是活动1结束之后活动安排问题（与活动1相容的活动进行安排的问题）的最优解，则存在最优解B'=(y2,...,yn), 所以|B'|>|B|

扩大问题规模，活动1开始之后的活动集合，构造解A'=(y1=1)UB'，A=(x1=1)UB

