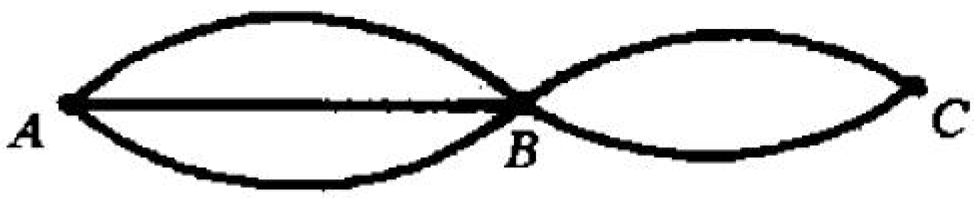
递　　推

加法原理

- 有位5男士和6位女士，从中选择1人去参加会议，有多少种不同的选择？
- 5+6=11

乘法原理

例 2 设从 A 到 B 有 3 条不同的道路，从 B 到 C 有 2 条不同的道路，如图 2.1.1 所示. 则从 A 经 B 到 C 的道路数为



加法原理与乘法原理

例 3 从 5 位先生、6 位女士、2 位男孩和 4 位女孩中选取 1 位先生、1 位女士、1 位男孩和 1 位女孩，共有 $5\times6\times2\times4=240$ 种方式(由乘法原则). 而从中选取一个人的方式共有 $5+6+2+4=17$ 种方式(由加法原则).

分类讨论

例 4 在 1000 到 9999 之间有多少个各位数字不同的奇数?

解 方法 1 如图 2.1.2 所示,第 4 位必须是奇数,可取 1,3,5,7,9,共有 5 种选择. 第 1 位不能取 0,也不能取第 4 位已选定的数字,所以在第 4 位选定后第 1 位有 8 种选择. 第 2 位不能取第 1 位和第 4 位已选定的数字,共有 8 种选择. 类似地,第 3 位有 7 种选择. 从而,满足题意的数共有 $5\times8\times8\times7=2240$ 个.

×	×	×	×
第1位	第2位	第3位	第4位

图 2.1.2

分类讨论

例 4 在 1000 到 9999 之间有多少个各位数字不同的奇数?

方法 2 把满足题意的数分成两类:

(i) 四位数中没有 0 出现. 类似于方法 1 的分析, 第 4 位数有 5 种选择, 第 3 位数有 8 种选择, 第 2 位数有 7 种选择, 第 1 位数有 6 种选择. 此类数共有 $6\times7\times8\times5=1680$ 个.

(ii) 四位数中有 0 出现. 这里, 0 只能出现在第 2 位或第 3 位上. 现假设 0 在第 2 位上, 则第 4 位照常有 5 种选择, 第 3 位有 8 种选择, 第 1 位有 7 种选择, 共有 $7\times8\times5=280$ 个数. 同理, 若 0 出现在第 3 位上, 也共有 280 个数.

由加法原则知, 合乎题意的数共有

$$1680+280\times2=2240(\text{个}).$$

小球排列

- 有4个不同的小球，从左往右排成一行，有多少种不同的排列方法？

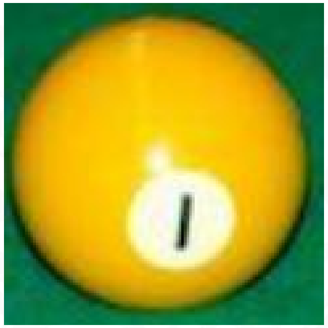
表达式的说明

- 我们用f(1)表示1个不同小球的排列的数量。
- 我们用f(2)表示2个不同小球的排列的数量。
- 我们用f(3)表示3个不同小球的排列的数量。
- 我们用f(4)表示4个不同小球的排列的数量。

- 显然，题目所求的答案是f(4)。

分类讨论最左边的小球

- 显然可以分成4大类：
- 第一大类是：最左边的小球是1。





这一大类的不同排列有多少种？

分类讨论最左边的小球

- 第二大类是：最左边的小球是2。

这一大类的不同排列有多少种？

分类讨论最左边的小球

- 第三大类是：最左边的小球是5。

这一大类的不同排列有多少种?

分类讨论最左边的小球

- 第四大类是：最左边的小球是8。



这一大类的不同排列有多少种？

原问题可以分成4大类

- 第一大类是：最左边的小球是1。
- 第二大类是：最左边的小球是2。
- 第三大类是：最左边的小球是5。
- 第四大类是：最左边的小球是8。

- 由于每大类的最左边的小球都确定了，那么问题就转化为求3个不同小球的排列的数量了。
- 根据前面的表达式说明和加法原理，可以得到：
- $f(4) = f(3) + f(3) + f(3) + f(3) = 4 \times f(3)$。

原问题与子问题

- 题目本来要求的是4个不同小球的排列数量。
- 这是原问题，即f(4)。
- 通过对最左边的小球进行分类讨论，只要能求出3个不同小球的排列数量，就能计算出f(4)。
- 而3个不同小球的排列数量，我们记为f(3)，相对f(4)来说，f(3)就是子问题。
- 那么f(3)又应该怎么算呢？也就是怎么算3个不同小球的排列数量呢？

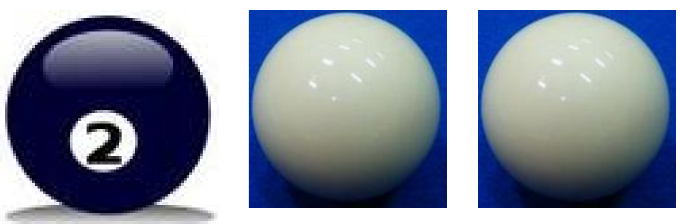
如何求f(3)?

- 3个不同小球的排列数量是多少？
- 例如我们有如下3个不同的小球，有多少种不同的排列？

分类讨论

- 显然，可以分成3大类：
- 第一大类，最左边的小球是1。

- 第二大类，最左边的小球是2。



- 第三大类，最左边的小球是5。

如何求f(3)?

- 根据前面的分类：
- $f(3) = f(2) + f(2) + f(2) = 3\times f(2)$。
- 所以只要能求出f(2)，就能利用上面的公式算出f(3)。

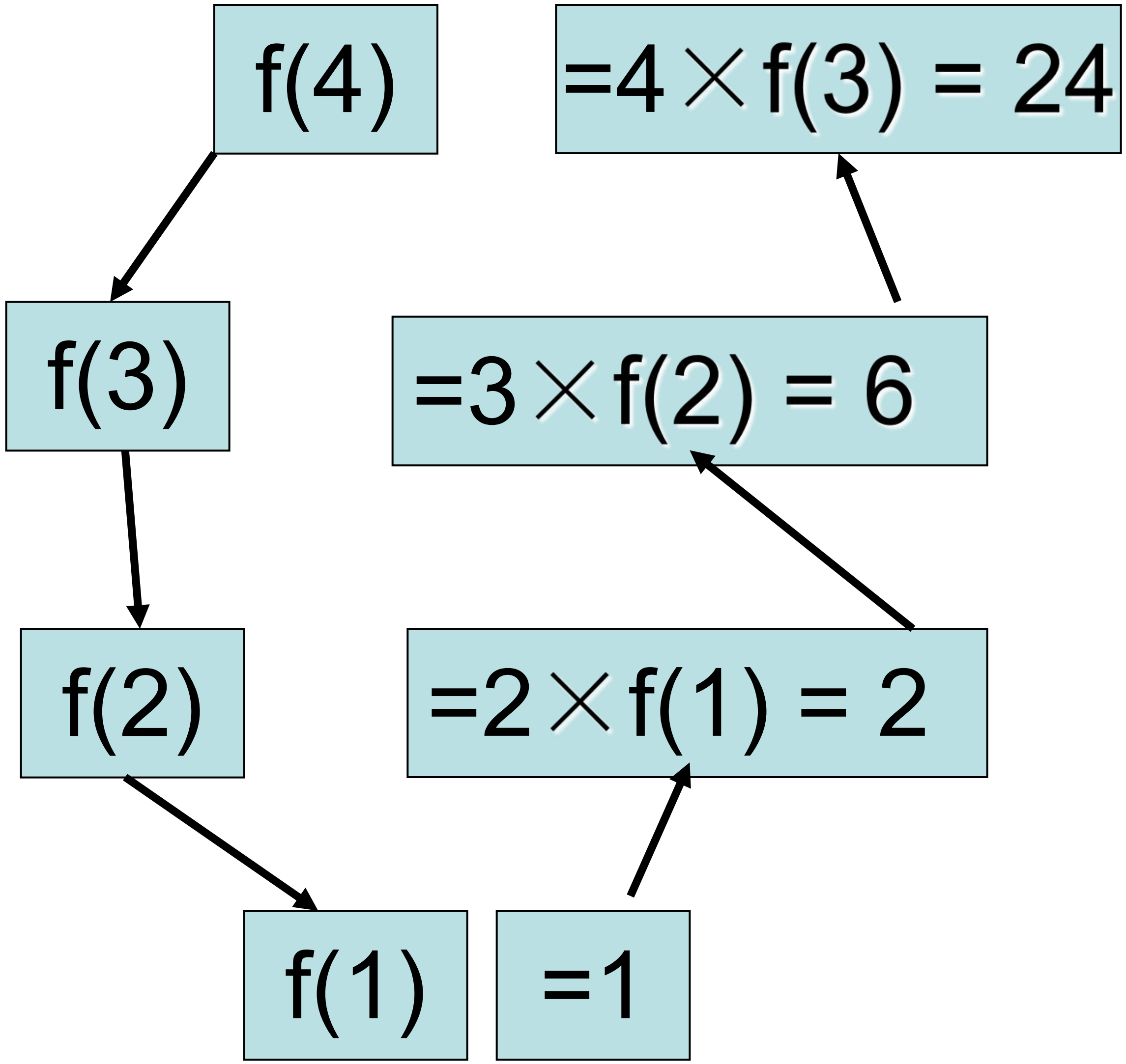
- 那么如何求出f(2)呢？
- 相对于f(3) 来讲，f(2)是子问题。
- 利用相同的分析方法，容易得到$f(2) = 2\times f(1)$。

- 而f(1)表示1个小球的排列数量，显然$f(1) = 1$。

分析小结

- 原问题是求f(4)，为了求出f(4)，必须先求出f(3)。
- 为了求出f(3)，必须先求出f(2)。
- 为了求出f(2)，必须先求出f(1)。
- f(1)是直接能算出来的。
- 就是这样，一直把大问题不断的转化为求小问题，直到问题的规模小到可以直接算出来或者已知为止，然后再用小问题的解递推出大问题的解。

左边是分析思考问题的过程，右边是实际递推的过程。



递推公式

- 上面讲的是4个不同小球的排列数量。
- $f(4) = 4 \times f(3)$
- $f(3) = 3 \times f(2)$
- $f(2) = 2 \times f(1)$。
- 可以容易推广到，n个不同小球的排列数量：
- 通过对最左边的小球的分类(共有n种不同的选择),根据加法原理和子问题的概念，容易得到递推公式:
- $f(n) = n \times f(n-1)$。
- $f(n-1) = (n-1) \times f(n-2)$。
- 。。。。。。。。。。。

  $f(1) = 1$

如何求f(n)的代码

- f[1] := 1;
- for i:=2 to n do f[i] := i * f[i-1];

骨牌铺法

- 有1×n的一个长方形，用一个1×1、1×2和1×3的骨牌铺满方格。

- 例如当n=3时为1×3的方格。此时用1×1、1×2和1×3的骨牌铺满方格，共有四种铺法。

- 那么有1×n的一个长方形，共有多少种铺法？

分析样例

- 把1×n的一个长方形填满的不同铺法的数量记为f(n)。
- 把1×(n-1)的一个长方形填满的不同铺法的数量记为f(n-1)。以此类推。
- 。。。。。。。。。。。。。。。。。。。。
- 先来分析样例：
- 如何求f(3)？
- 分类讨论！！
- 对谁分类讨论？？
- 对第一个格子！！
- 第一个格子是可以怎样填？

图7.1.1

- 由于每个格子最终都要被填满，所以第1个格子也一定会被填满。
- 第1个格子可以分成3大类：
- 第1大类：第1个格子被1×1的骨牌填。
- 第2大类：第1个格子被1×2的骨牌填。
- 第3大类：第1个格子被1×3的骨牌填。

- 第1个格子可以分成3大类：
- 第1大类：第1个格子被1×1的骨牌填。因为后面还有2个格子需要被填满，所以这一大类的方案数是f(2)。
- 第2大类：第1个格子被1×2的骨牌填。由于后面还有1个格子需要被填满，所以这一大类的方案数是f(1)。
- 第3大类：第1个格子被1×3的骨牌填。由于后面还有0个格子需要被填满，所以这一大类的方案数是f(0)。

- 于是, f(3) = f(2) + f(1) + f(0)。

递推公式

- 通过对第1个格子进行分类，可以分成3大类，可以用1×1的骨牌填第1个格子，也可以用1×2的骨牌填第1个格子，还可以用1×3的骨牌填第1个格子，每一大类后面都是变成要求一个子问题的解，利用加法原理，容易知道：
- $f(n) = f(n-1) + f(n-2) + f(n-3)$。
- $f(1) = 1$;
- $f(0) = 1$; 为什么？

攀天梯

- 【问题描述】
- 北武当山主峰四周几乎都是陡壁悬崖，只有一条人造“天梯”可攀，天梯由n级就山凿筑的石阶组成，欢欢打算通过天梯攀上北武当山主峰。攀天梯时，他有时一步一级石阶，有时一步两级，那么，他攀上这n级的天梯有多少种不同的方法？
- 【输入格式】
- 输入数据仅一行，这一行只包含一个整数n(1≤n≤80)。
- 【输出格式】
- 输出数据仅一行，这一行只包含一个整数，表示欢欢攀上这n级天梯的方法数。
- 【输入样例】
- 5
- 【输出样例】
- 8

$$f(n) = f(n-1) + f(n-2)$$

如何写出递推公式？

- 要写出递推公式，首先要找到子问题？
- 如何找到子问题？有没有什么常规的手段？
- 可以分类讨论最后一步的状态来确定子问题。有些题目可能是第一步。设从底走到第n级的走法有f(n)种，因为走到第n级有两种方法，一种是从第（n-1）级走1级，另一种是从第（n-2）级走2级，前者有f(n-1)种方法，后者有f(n-2)种方法，所以f(n)=f(n-1)+f(n-2)，且f(0)=1，f(1)=1。

小结

- 通过前面的例子，相信大家应该有个强烈的感觉，那就是要求出大问题的解，必须先求出子问题的解！关键是如何找出子问题！最常规的手段就是通过分类讨论，运用加法原理，对第一个或者最后一个元素的所有情况进行考虑，然后就可以写出递推公式，有了递推公式，再从小问题的解推出大问题的解！

进阶例题

- 有x个不同的小球,现在从中取出y个小球, 有多少种不同的取法? 其中 y <= x。

  数学公式：C(x,y)，不是我们讨论的内容。

  递推思路：

  原问题记为：f(x, y)

  子问题呢？

  分析实例：f(4,2)

  现在有A、B、C、D,4不同小球, 从中取出2个。

A

B

C

D

现在考虑第一个小球A的取法：

不取小球A， 那么必然要在B、C、D

3个小球中取2个。有多少种不同取法?正是子问题f(3,2)

A

B

C

D

取小球A，　那么要在B、C、D三个小球中取1个，有多少种不同取法?

正是子问题f(3,1)

根据加法原理：

$f(4,2) = f(3,2) + f(3,1)$。

也就是说只要求出了$f(3,2)$和$f(3,1)$之后，就可以求出$f(4,2)$了。

可以看出，相对于原问题$f(4,2)$来说，$f(3,2)$是子问题，$f(3,1)$也是子问题。

那么$f(3,2)$怎么求呢？利用相同的分析思路，易知

$f(3,2) = f(2,2) + f(2,1)$。

为什么呢？请某位回答一下。

于是得到递推公式：

$f(x,y) = f(x-1, y) + f(x-1, y-1)$，其中$x >= y$

而边界条件是：$f(x, y) = 1$，当 $x = y$

$f(x,y) = x$，当 $y = 1$或 $y = 0$

	$y=1$	2	3	4
$x=1$	1			
$x=2$	2	1		
$x=3$	3	3	1	
$x=4$	4	6	4	1

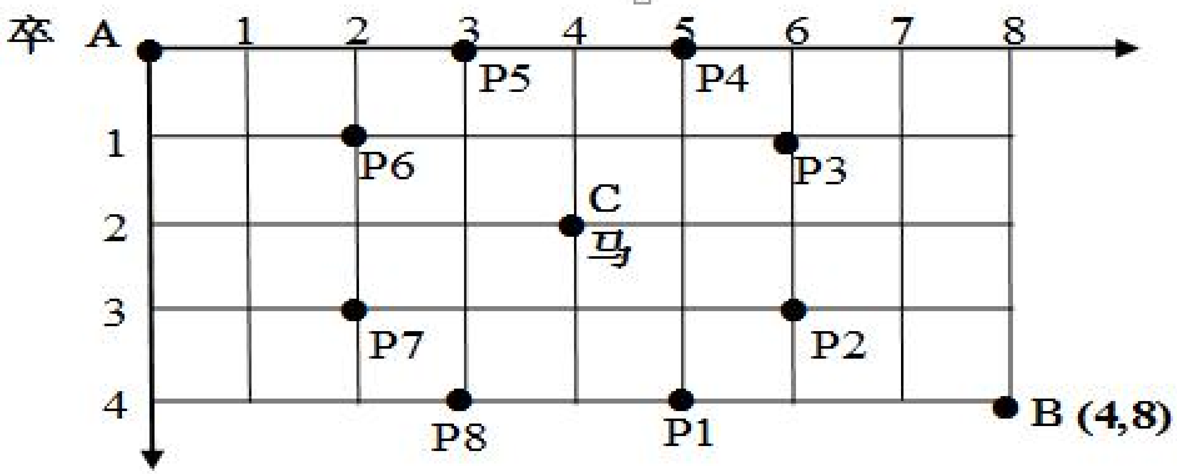
如何写程序？

- 通过上面的分析，应该从上往下推，同一行从左往右推，只需二重循环即可。
- for i:= 0 to x do f[i,0] := 1;   //边界条件
- for i := 1 to x do
-     for j := 1 to x do
-     begin
-         if j =1 then f[i,j] := i;
-         else if j = i then f[i,j] := 1;
-         else f[i,j] := f[i-1,j] + f[i-1,j-1] ;

    end;

马拦过河卒

如下图，A点有一个过河卒，需要走到目标B点。卒行走的规则：可以向下或者向右走。



- 同时在棋盘上的某一点有一个对方的马（如上图的C 点），该马所在的点和所有跳跃一步可达的点称为对方马的控制点。例如上图C点上的马可以控制9 个点（图中的p1，p2，…，p8和C）。卒不能通过对方马的控制点。棋盘用坐标表示，点A (0，0)、点B (n，m)(n，m为不超过20 的整数)，马的位置坐标是C点(x，y)(约定：C≠A，且C≠B)。现在要求你计算出卒从A 点能够到达B点的路径的条数。

简化问题

- 假如没有卒，要从点(0,0)走到点(4,8)，每一步只能向下走或者向右走，有多少种不同的方案？
- 请应用前面讲的分析子问题的方法和加法原理，写出递推公式。即f[i,j] := ？

请分析下页的f[4,8] := ??

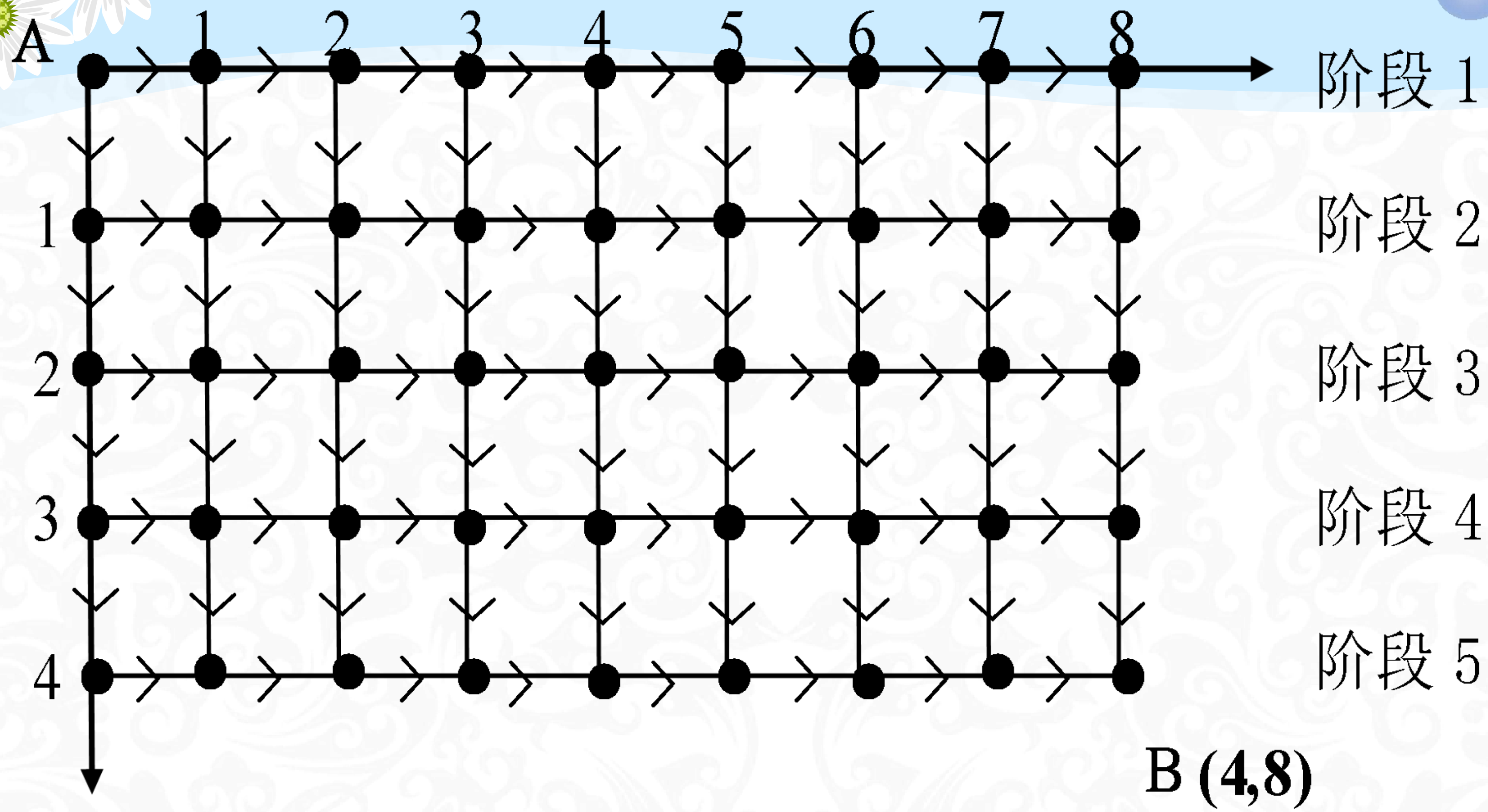


图 7.4-2

回到原题，分析题目

- 我们可以设f[x,y]记录从起点到达点(x,y)的路径条数，能到达点(x,y)的有点(x-1,y)和点(x,y-1)，很显然它们的关系符合排列组合的“加法原理”，那么可得出以下递推式：
- 当(x,y)不在马控制点上时，则f[x,y]=f[x-1,y]+f[x,y-1]；
- 当(x,y)在马的控制点上时，则f[x,y]=0；

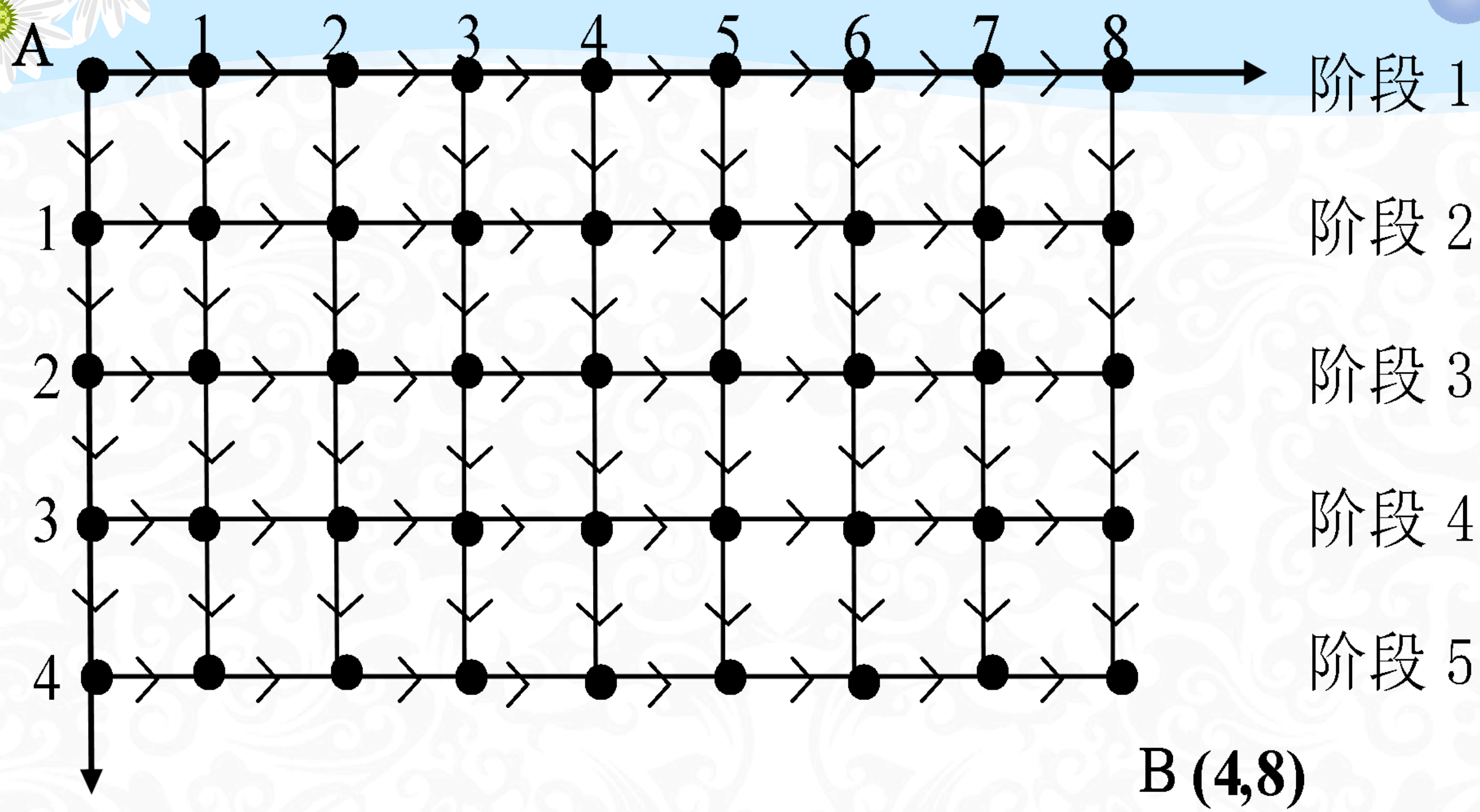


图 7.4-2

```
const
  dx:array[1..8] of integer=(2,1,-1,-2,-2,-1,1,2); //记录方向值
  dy:array[1..8] of integer=(1,2,2,1,-1,-2,-2,-1); //记录方向值
var f:array[0..20,0..20] of longint; //f数组记录从点a到点(i,j)的路径条数
    g:array[-2..22,-2..22] of boolean; //记录各个点的可行状况
    i,j,n,m,x,y:integer;
begin
  readln(n,m,x,y);
  fillchar(g,sizeof(g),true);
  g[x,y]:=false;
  for i:=1 to 8 do g[x+dx[i],y+dy[i]]:=false; //设置9个障碍点
  if g[0,0] then f[0,0]:=1;
  for j:=1 to m do
    if g[0,j] then f[0,j]:=f[0,j-1]; //设置上边边界
  for i:=1 to n do
    if g[i,0] then f[i,0]:=f[i-1,0]; //设置左边边界
  for i:=1 to n do
    for j:=1 to m do
      if g[i,j] then f[i,j]:=f[i-1,j]+f[i,j-1];
  writeln(f[n,m]);
end.
```

字符传输

- 在网络上传输由**a,b,c**三种字母组成的长为**n**的字符串，若字符串中有两个**a**连续出现，则网络上不能传输，令**f(n)**表示网络上可以传输的长为**n**的字符串的个数，求**f(n)**的递推公式。

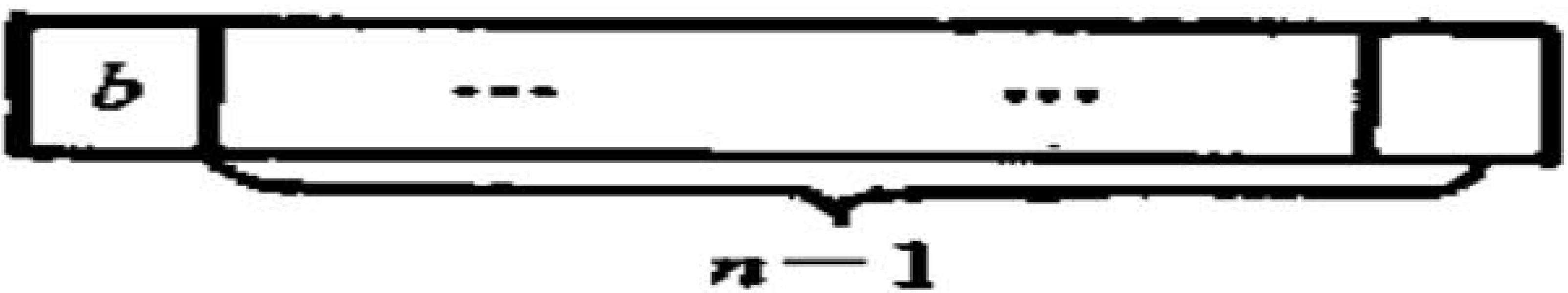
如：　**aba**是合法的长度为**3**的字符串，而**baa**则是非法的长度为**3**的字符串。

找子问题

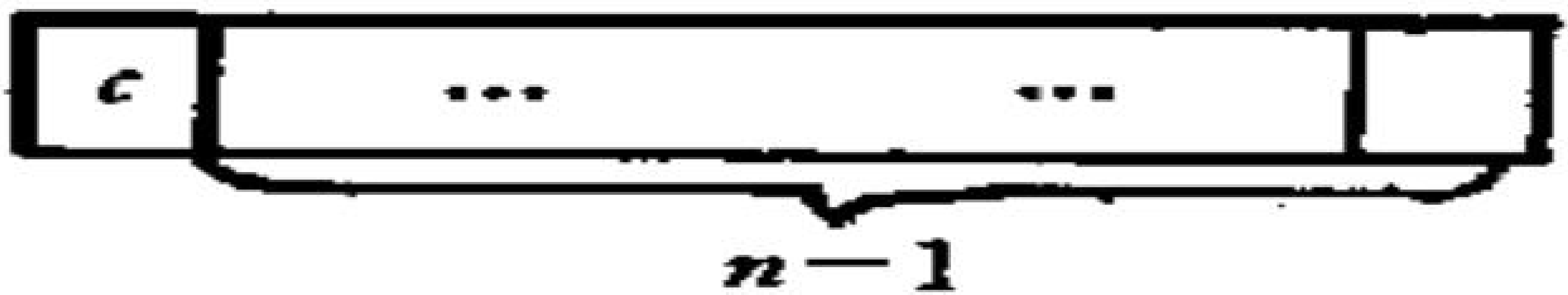
- 还是采用分类讨论的方法，找子问题。

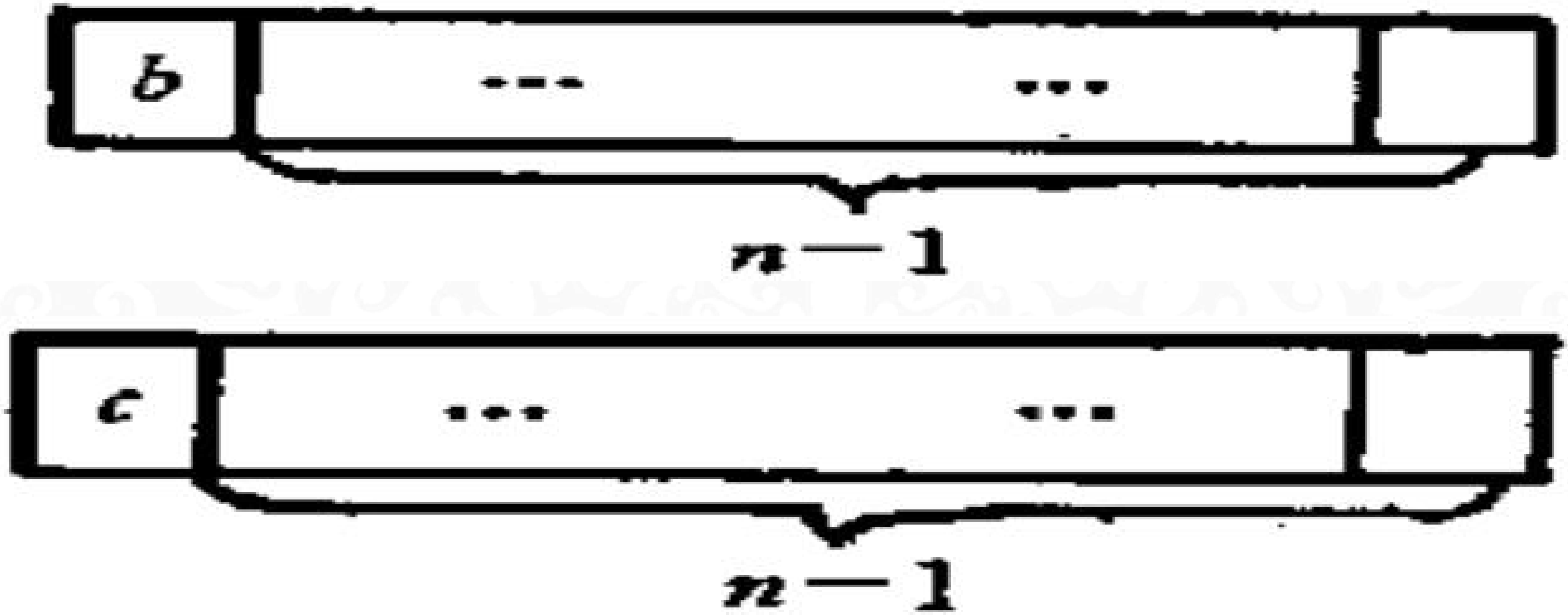
考虑第一个字符是什么，共有**a**、**b** 、**c**。

先看第一个字符是**b**的情况：



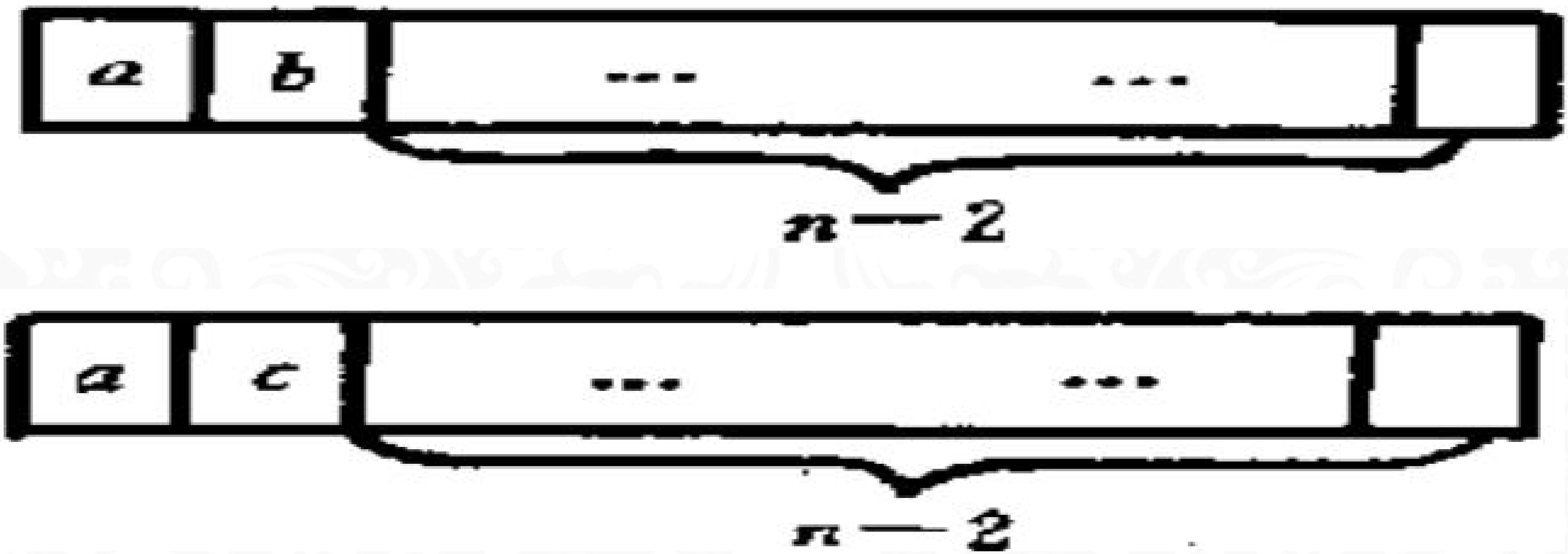
再看第一个字符是**c**的情况：





- 对于这两种情况，容易看出，只要后面的n-1个字符没有连续的两个a,那么在那基础上在开头多加一个字符b或c肯定也是合法的字符串，于是原问题求f(n) 就可以先求出子问题f(n-1)了。 也就是可以先求出长度是n-1的合法字符串有多少个。

第一个字符是a的情况：



对于这两种情况，子问题都是f(n-2).

于是综合上述情况，我们可以得到递归表达式：$f(n) = 2\times f(n-1) + 2\times f(n-2)$

总结

- 递推的关键是递推公式。
- 递推公式的关键是找出子问题。
- 找出子问题的一个常规实用手段是，考虑第一个或者最后一个的各种可能情况，然后根据加法原理或者乘法原理，得到子问题。