猜想及其应用

【摘要】

猜想是重要的解题策略，对培养人的综合思维能力大有好处。

本文第一部分介绍了猜想的基本要求和实现步骤，提出了两种重要的得到猜想的方法：类比和归纳。

第二、三部分分别介绍了类比、归纳的基本性质、定义，以及解题的一般步骤。每一部分都通过若干道题目实际说明猜想在信息学竞赛中的应用。

最后总结全文，对猜想的方法、定义、分类、重要性等做了全面阐述。

【关键字】猜想　类比　归纳

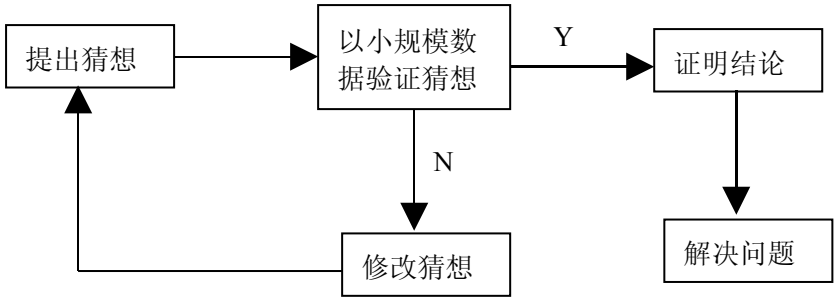
【正文】

1、引言

猜想，是解决问题最基本、最重要的方法，是完成从已知世界到未知世界探索的有力工具。在信息学竞赛中，随着试题的隐蔽性越来越高，“猜想”这一研究手段也扮演着越来越重要的角色。

猜想不是胡猜乱想，必须综合分析条件作出“言之成理”的推测，然后在猜测的引导下进行“持之有据”的论证。因此猜想绝不仅仅是运气或者投机，正如一个小学生很难提出费马大猜想（现在已经证明了），不经过深入的观察、分析我们也不可能提出任何有价值的猜想。

进行猜想的基本步骤是：



由此可见，猜想是在深入分析问题的基础上，不懈探索、反复修正的过程。决不可能一蹴而就。

“提出猜想”是整个过程的核心。得到猜想的方法主要有类比和归纳。

2、类比猜想

所谓类比，就是由两个对象的某些相同或相似的性质，推断它们在其他性质上也有可能相同或相似的一种推理形式。

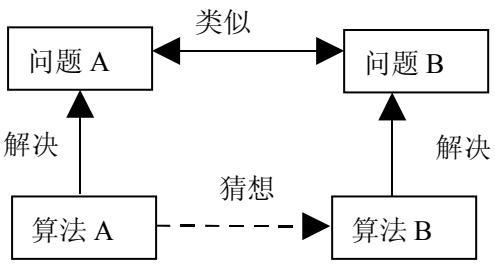
形式性的表示为：

事物 A 具有性质 a, b, c, d，事物 B 具有与之相应的类似性质 a’, b’, c’，从而猜想 B 也有与 d 对应的类似性质 d’。

类比是一种**主观的不充分的似真推理**，因此，要确认由类比而得的猜想的正确性，还须经过严格的逻辑论证。

§2.1 与熟悉的问题类比

若问题 A（熟悉）已经解决，问题 B（陌生）在形式、结构或性质方面与 A 类似，那么可以猜想：解决 B 的方法类似于解决 A 的方法。形象的表示为：



与熟悉的问题进行类比的基本步骤是：

1、分析。分析问题的特征，抽象出模型。

2、联想。与熟悉的，拥有类似特征、模型的问题类比。

3、比较。确定类比对象后将两者比较，分析异同。

4、猜想。根据已知问题猜想新问题的解决方法。

【引例】巨人与鬼

问题描述

平面上有 n 个巨人和 n 个鬼（1<=n<=50）。每个巨人都有新式激光武器，可以对鬼进行直线攻击。然而这种武器有一种很大的弱点：两个不同武器发出的激光不能交叉，否则就有爆炸的危险。

已知任意三个个体（包括巨人和鬼）都不在一条直线上，每个巨人负责攻击一个鬼。当然激光不能交叉。

现在的问题是：请你求出任意一种攻击的方案。（即确定哪个巨人攻击哪个鬼）

输入

- 输入文件的第一行只有一个整数 n。
- 接下来 n 行每行两个整数 x，y，表示巨人的坐标。
- 接下来 n 行每行两个整数 x，y，表示鬼的坐标。

输出

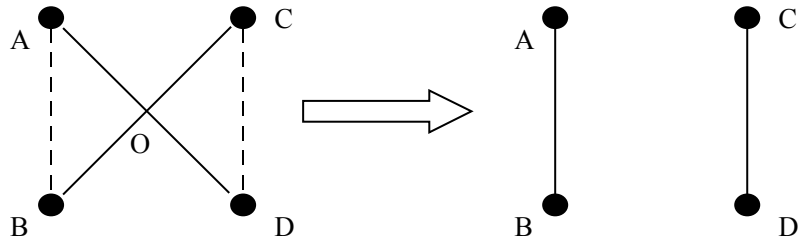
- 输出文件有 n 行，每 i 行包含一个整数 p，表示第 i 个巨人攻击第 p 个鬼。

分析

试题中是实际上是要确定巨人与鬼之间的一一对应关系——这使我们很自然的联想到了匹配。

将巨人、鬼都抽象成为顶点。巨人组成的点集记为 X，鬼组成的点集记为 Y。在所有的 x∈X、y∈Y 之间连一条边，接下来用匈牙利算法求二分图的完备匹配——

然而题目还有一个特别的条件：任意两条匹配边不能相交。所以简单的匹配还不行。我们不妨令两条边相交，研究其特性：



如果 A 攻击 D，C 攻击 B（如上图左所示），则匹配边会相交。一种自然的想法是将其变换到右图所示的形式。观察图形特点，发现：

$OA + OB > AB$，$OC + OD > CD$

$AD + BC > AB + CD$

新的匹配方案与老的匹配方案相比，“匹配边的长度总和”减小了。故而，每一个包含相交边的完备匹配都能够转换成为“匹配边长度总和”更小的一个完备匹配——换句话说，“匹配边长度总和”最小的完备匹配一定不包含相交边！

将原来的二分图完备匹配模型转换成二分图最佳匹配模型：每条边的权值设为它两端顶点的欧几里德距离。

问题解决了。

小结

本题二分图中的顶点具有特殊性：每个顶点都对应于平面上的一点。因此，顶点与顶点之间不仅有逻辑关系，还有**几何直观**关系。正是由于充分利用了“几何直观”这一隐蔽关系才使得算法模型豁然开朗。

【例 1】青蛙的烦恼

问题描述

池塘里有 n 片荷叶（1<=n<=1000），它们正好形成一个凸多边形。我们按照顺时针方向将这 n 片荷叶顺次编号为 1, 2, …, n。

有一只小青蛙站在一号荷叶上，它想跳过每片荷叶一次且仅一次（它可以从所站的荷叶跳到另外任意一片荷叶上）。同时，它又希望跳过的总距离最短。

请你编程，帮助小青蛙设计一条路线。

输入

- 输入文件第一行有一个整数 n。
- 接下来 n 行每行两个整数 x, y，表示第 i 片荷叶的坐标。

输出

- 输出文件只有一个整数，即最短总距离。

分析

这是一道难题。

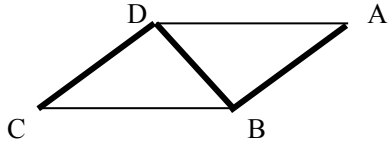
首先将每片荷叶抽象成一个顶点，在任意两个顶点 x, y 之间连一条边，边的权值设为顶点 x 与 y 的欧几里德距离。

容易看出，题目的本质是求一条从 1 出发的 Hamilton 链——这是一道经典 NP 难题。由于 n 可以多达 1000，所以首先排除搜索法。

我们首先很自然的猜想：

猜想 1 从 1 开始，按照顺时针顺序依次遍历每一个点，得到的路线是最短的。

然而一个简单的实例将之推翻了。



这里的 ABCD 是平行四边形。按照猜想 1，最佳路线应该是 ABCD；然而实际上的最短路线应该是 ABDC！因此简单的依次遍历凸包上的点还不行。

于是我们仔细揣摩题目的特性。分析发现，构造的图：

1、是一个完全图。

2、边的权值等于它两端顶点的欧几里德距离。

3、图中的每个顶点对应于平面上的一点。

这些性质使人不由自主的联想到【引例】。（这是表面上的类似 ）

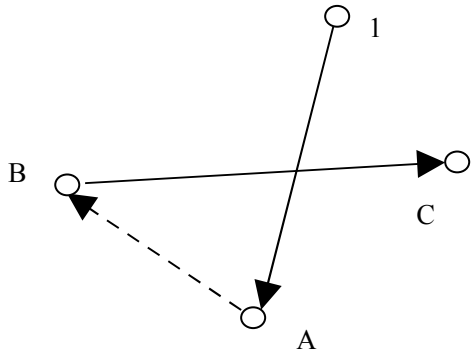
由于图中的每个顶点对应于平面上的一点，所以顶点之间不仅存在着由“边”连接的逻辑关系，还存在着特殊的“几何直观”关系。利用好这个隐含的条件是解题的关键——这正与【引例】的基本特性不谋而合！（这是本质上的类似）

解决【引例】时利用这个特性可以将“匹配边总长度”不断的缩短；而本题所求的问题也是一个最短路问题。

根据这两点相似之处，我们猜测：

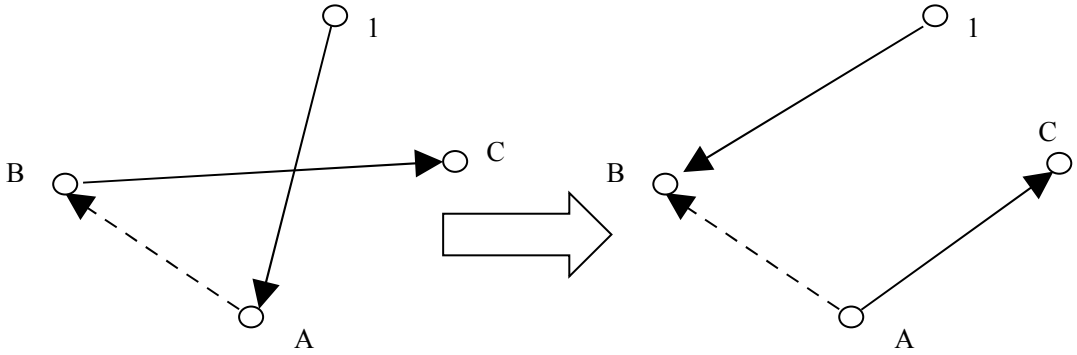
猜想 2 最短 Hamilton 链中不存在相交的边。

证明 设从 1 出发，首先到达顶点 A(2 < A < n)。由于所有点在一个凸包上 2<A<n，所以除 1、i 以外的顶点必然被分作两个部分，分别位于线段<1, A>的两侧。不失一般性，设从 i 出发经过若干步到左侧的顶点 B，再到右侧的顶点 C，如下图所示：



实线表示一步到达；虚线
表示多步到达

我们可以做这样的变换：



实线表示一步到达；虚线
表示多步到达

实线表示一步到达；虚线
表示多步到达

变换后，从 1 到 D 的距离显然比变换前的距离要短；故变换前的方案必然不是最优方案。所以从 1 出发，第一步只能到 2 或者 n。

同理，如果第一步到了 2，则第二步只能到 n 或者 3；如果第一步到了 n，则第二步只能到 n-1 或者 2……因此最后的最优方案中肯定不存在相交边。

猜想 2 是成立的!

接下来的任务就好办了。根据证明容易知道，从 1 出发，第一步只能到 2 或者 n。如果第一步到 2，则问题转化为“以 2 为起点，遍历{2..n}中的顶点一次且仅一次”；如果第一步是到 n，则问题转化为“以 n 为起点，遍历{2..n}中的顶点一次且仅一次”。

转化后的子问题与原问题结构相同，只是规模缩小，故可采用动态规划。

设 f[s, L, 0]表示从 s 出发，遍历{s..s+L-1}中的顶点一次且仅一次的最短距离；f[s, L, 1]表示从 s+L-1 出发，遍历{s..s+L-1}中的顶点一次且仅一次的最短距离。则：

```
f[s, L, 0] = min{dis(s,s+1) + f(s + 1, L-1, 0),
                 dis(s,s+L-1) + f(s + 1, L-1, 1)}
f[s, L, 1] = min{dis(s+L-1, s+L-2) + f(s, L - 1, 1)
                 dis(s+L-1, s) + f(s, L-1, 0)}
f[s, 1, 0] = 0, f[s, 1, 1] = 0
```

其中 dis(a, b)表示第 i 个顶点和第 j 个顶点之间的欧几里德距离。

算法的时间复杂度是 $O(n^2)$。

题目小结

【例 1】是一道比较难的题目。解题的突破口在于与【引例】的类比——几何直观上的类似，触发了一个大胆的猜想，分析了最优路线的特性，为动态规划创造了条件。甚至猜想的证明方法，都和【引例】分析的一些思想密切相关。这与两个问题本质、特点的类似不无关系。

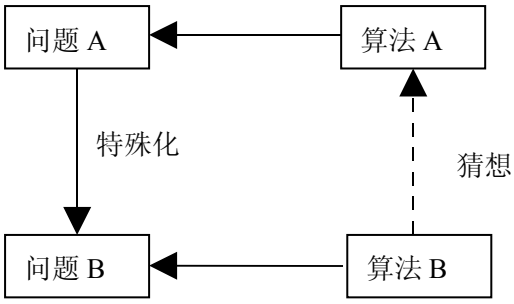
本节小结

与熟悉的问题类比的前提是对事物深入、透彻的分析。只有分析出事物的本质特性，才有可能将那些为纷繁复杂的具体要素所掩盖的相似模型联系起来，才有可能根据模型的具体情况提出合情合理的猜想。

同时观察、联想是实现类比的基础。通过观察，得到问题的本质模型；通过联想，将模型同熟悉的、类似的问题联系起来，提出猜想。

§2.2 与特殊的问题类比

特殊化类比是又一类常用的类比方式。它将问题中的某些条件特殊化、极端化、边界化，通过对极端情况下的问题的分析，试图找出解题的规律；然后将其与原问题类比，提出猜想。可以形象的表示为：



特殊类比的一般步骤：

1、观察。观察题目条件的特征。

2、特殊化。将一些条件的位置、数量、状态等特殊化。

3、分析特殊问题。

4、类比。将特殊问题与原问题作比较。

5、猜想。根据特殊化问题与原问题的联系和不同，以特殊问题的解决方案为蓝本，猜想原问题的解法。

【例 2】圆圈点问题

问题描述

平面上有 n 个点（3<n<=10000，n 是奇数），任意三点不共线、任意四点不共圆。现在请你编程：从中找出三点确定一个圆，使得(n－3)/2 个点在圆外、(n-3)/2 个点在圆内。

输入

- 输入文件第一行是一个整数 n。
- 接下来 n 行，每行两个整数 x, y，表示第 i 个顶点的坐标。

输出

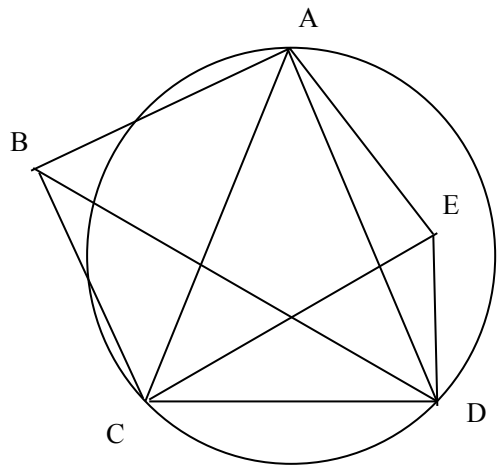
- 输出文件仅一行，包括三个整数，分别表示选择的三个点的编号。

分析

看完题目，条件反射的想到枚举：枚举 3 个点确定一个圆，然后一一判断其他点与圆的位置关系，直到找到一个满足题目要求的解为止。

然而这种算法的致命伤就是时间复杂度太高——n <= 10000！我们不得不另谋它法。

由于 n 个点关系凌乱、位置分散，很不利于我们思考——于是我们考虑将 n 个点的位置关系特殊化：令 n 个点组成一个凸包。同时令 n=5（最小值）。



如上图所示，A、B、C、D、E 五点组成一个凸包。选择 ACD 三点确定一个圆，则 B 在圆外、E 在圆内。

为什么选择 ACD 三点确定一圆，可以使得 B、E 分处圆的内、外呢？这是因为五个点组成一个凸包，所以 A、B、E 都在 CD 同侧，故而∠CAD、∠CBD、∠CED 是 CD 同侧的张角；且∠CBD<∠CAD<∠CED、∠CAD 是圆上一点，则 B 在圆外、E 在圆内。

从这个特殊情况中，我们可以概括出如下要素：

1、A、B、E 在 CD 的同侧。

2、∠CBD<∠CAD<∠CED。

根据*要素 1*∠CAD、∠CBD、∠CED 是 CD 同侧的张角，因此这三个角可以用作判定 A、B、E 与圆的位置关系的条件；*要素 2* 使得 B、E 分处圆内和圆外。

两个要素是算法正确性的根本保障，缺一不可。

接着我们将其与原问题类比，提出类似于要素 1、2 的猜想：

如果原问题中 A、B、C 三点满足：

1、 **AB 使得除 A、B 外的其他点都在 AB 的同侧。**

2、 **将所有的∠AXB 排序（X <> A, B），∠ACB 排在第(n-3)/2+1 位。**

则 A、B、C 必然是满足要求的三点。

证明 由于满足条件 1，所有的∠AXB 都是 AB 同侧的张角。因此：如果∠AXB > ∠ACB，则 X 在圆 ABC 内；如果∠AXB < ∠ACB，则 X 在圆 ABC 外。

∠ACB 排在第(n-3)/2+1 位，所以有(n-3)/2 个角比它大、(n-3)/2 个角比它小。因此有(n-3)/2 个点在圆外；(n-3)/2 个点在圆内。

所以 A、B、C 是满足要求的三点。

故而我们可以先求 n 个点的凸包，任选相邻的两点 A、B，则其余的点必在 AB 同侧。然后将其余点按照与 AB 的张角排序，取排在第(n-3)/2+1 的那一个点 C。则 A、B、C 即为所求。

题目小结

此题的实质性突破在于对问题进行了特殊化。特殊化后的问题，点与点之间的关系简洁、明了，我们稍作分析就轻松解决了此问题。

之后我们分析了特殊情况中的一些解题要素，得出了一个较普遍的结论，并据此猜想原问题的算法。

本节小结

与特殊的问题类比是得到猜想的重要方法之一。由于特殊化后的问题仍然保留了原问题的大部分特征、性质，所以根据其研究成果进行猜想相对其他类比要容易些。

运用特殊化类比法要特别注意比较特殊化后的问题与原问题的不同——绝不能机械的推广特殊算法。

特殊类比的前提是积累。到底将什么条件特殊化？特殊成一个什么样子？这都要靠平时训练时多留心、多积累，丰富解题的经验。没有扎实的积累，“特殊类比”只能是一座空中楼阁。

§2.4 类比小结

类比是得到猜想的重要方法。与特殊的问题类比、与熟悉的问题类比则是类比的基本方式。

前者是将陌生的问题与熟悉的类似问题相类比，猜测陌生问题的解决方法。即陌生□ 熟悉。

后者是将复杂的问题简单化、特殊化，通过对简单问题的分析，得出一般的解题规律，并将之与原问题类比，达到解决问题的目的。即一般□特殊，复杂□简单。

这两种方法的本质都是“化归”——从已知世界到未知世界的转化。这种转化的实现，必须建立在积累、观察、分析、联想的基础上，而绝不仅仅是运气。

3、归纳猜想

§3.1 归纳猜想的理论基础

所谓归纳，是指通过对特例的分析来引出普遍结论的一种推理形式。它由推理的前提和结论两部分构成：前提是若干已知的个别事实，是个别或特殊的判断、陈述，结论是从前提中通过推理而获得的猜想，是普遍性的陈述、判断。其思维模式是：设M_i（i＝1，2，…，n）是要研究对象M的特例或子集，若M_i（i＝1，2，…，n）具有性质P，则由此猜想M也可能具有性质P。

如果$\bigcup_{i=1}^{n} M_i$＝M，这时的归纳法称为完全归纳法。由于它穷尽了被研究对象的一切特例，因而结论是正确可靠的。完全归纳法可以作为论证的方法，它又称为枚举归纳法。

如果 $\bigcup_{i=1}^{n} M_i$ 是 M 的真子集，这时的归纳法称为不完全归纳法。由于不完全归纳法没有穷尽全部被研究的对象，得出的结论只能算猜想，结论的正确与否有待进一步证明或举反例。

在信息学中，我们一般先使用“不完全归纳法”得出普遍结论，然后用“完全归纳法”证明之（一般采用数学归纳法）。

归纳猜想的一般步骤：

1、 列举。将一些特殊的、简单的、小规模的数据列举出来。（这一步可以用手推，或者编写简单的搜索小程序）

2、 观察。观察列举数据的规律。

3、 猜想。根据部分数据猜想一般结论。

4、 证明。证明猜想的正确性。（一般采用数学归纳法）

§3.2 归纳猜想在信息学中的应用

【例 3】青蛙过河

问题描述

大小各不相同的一队青蛙站在河左岸的石礅（记为 A）上，要过到对岸的石礅（记为 D）上去。河心有几片荷叶（分别记为 Y1…Yn）和各石礅（分别记为 S1…Sn）。

青蛙的站队和移动方法规则如下：

1、 每只青蛙只能站在荷叶、石礅，或者仅比它大一号的青蛙背上（统称为合法的落脚点）。

2、 一只青蛙只有背上没有其他青蛙的时候才能够从一个落脚点到零一个落脚点。

3、 青蛙允许从左岸 A 直接跳到河心的石礅、荷叶和右岸的石礅 D 上，允许从河心的石礅和荷叶跳到右岸的石礅 D 上。

4、 青蛙在河心的石礅之间、荷叶之间以及石礅和荷叶之间可以来回跳动。

5、 青蛙在离开左岸石礅后，不能再返回左岸；到达右案后，不能再跳回；

6、 假定石礅承重能力很大，允许无论多少只青蛙直接站在上面，而其他的青蛙只能依规则 1 落在比它大一号的青蛙的背上。

7、 荷叶不仅面积不大，而且负重能力也有限，至多只能有一只青蛙站在上面。

8、 每一步只能移动一只青蛙，而且移动后需要满足站队规则。

9、 在一开始的时候，青蛙均站在 A 上，最大的一只青蛙直接站在石礅上，而其他的青蛙依规则 6 站在比其他大一号的青蛙的背上。

青蛙希望最终能够全部移动到D上，并完成站队。

设河心有m片荷叶和n个石礅，请求出这队青蛙至多有多少只，再满足站队和移动规则的前提下，能从A过到D。

输入

- 文件仅有两行，每一行仅包含一个整数。
- 第一行的数字为河心的石礅数n（0<=n<=25）。
- 第二行的数字为荷叶数m（0<=m<=25）。

输出

- 文件仅包含一个整数。即至多有多少只青蛙能过河。

分析

乍看此题毫无头绪。注意到最后的结果只与n, m这两个参数有关，所以我们初步猜想：结果是一个关于n, m的数学表达式。我们从简单情况入手，看看有什么规律。

设通过n个石礅、m片荷叶可以使f[n,m]只青蛙从左岸跳到右岸。那么：

当n = 0时，显然f[0, m] = m + 1。

当m = 0时，我们可以编一个简单的程序搜索出部分结果，发现：f[1, 0] = 2, f[2, 0] = 4, f[3, 0] = 8。于是猜测：$f[n, 0] = 2^n$。

当m = 1时，根据简单搜索的结果可以知道：f[1, 1] = 4, f[2, 1] = 8, f[3, 1] = 16。于是猜测$f[n, 1] = 2^{n+1}$。

我们总结一下（均是猜测）：

m = 0时，$f[n, 0] = 2^n$。

m = 1时，$f[n, 0] = 2^{n+1}$。

n = 0时，f[0, m] = m + 1。

通过观察，我们联想到：

m = 0时，$f[n, 0] = 2^n = (0 + 1) * 2^n = (m + 1) * 2^n$。

m = 1时，$f[n, 0] = 2^{n+1} = 2 * 2^n = (m + 1) * 2^n$。

n = 0时，$f[0, m] = m + 1 = (m + 1) * 1 = (m + 1) * 2^n$。

所以我们提出猜想：

猜想 $f[n, m] = (m + 1) * 2^n$。

下面设法证明。

当n = 0时。

先令m只青蛙跳到荷叶上，然后令编号为m + 1的青蛙跳到D，最后让荷叶上的m只青蛙按照编号由大到小的次序依次跳到D即可。总共可以容许$m + 1 = (m + 1) * 2^n$只青蛙过河。即f[0, m] = m + 1。

当n <> 0时。

显然必须先令编号小于f[n, m]的青蛙跳到荷叶、石礅上暂留；然后令编号为f[n, m]的青蛙跳到D；最后再让荷叶、石礅上的f[n,m]- 1只青蛙跳到D上。所以要使得过河的青蛙最多，必须让尽量多的青蛙跳到荷叶、石礅上去。

首先考虑使编号为n的石礅上站尽量多的青蛙。可以利用其他的n-1个石礅和m片荷叶，所以首先能有f[n-1,m]只青蛙跳到编号为n的石礅上。

同理最多可令f[n-2,m]只青蛙跳到编号为n-1的石礅上，……，最多可令

f[0, m]只青蛙跳到编号为1的石磴上。

故而f[n,m]=f[n-1,m]+f[n-2,m]+…+f[0,m]+m+1。

可以看到，这里的参数m不起作用。令g[n] = f[n, m]，则：

g[i] = g[0] + g[1] + … + g[i-1] + m + 1

g[0] = m + 1

求：g[n]

稍加分析可知：f[n, m] = g[n] = (m + 1) * 2^n。

猜想正确。

本题小结

这是NOI2000第二试第二题，有一定难度，不用归纳猜想很难做出来；即便做出来，也要花费相当惊人的时间。

需要注意的是：归纳而得的猜想不一定正确，必需严格的证明。证明的方法大多是数学归纳法。这是由归纳猜想本身的特性决定的。

【例4】排列问题

问题描述

在整数1，2，……，N的排列中，有些排列满足下面一个性质A：该排列中除了最后一个整数以外的每一个整数后面都跟有（不必直接紧跟）一个同它相差为1的整数。例如：N = 4，排列1432是具有性质A的，而2431则不满足。

设有一个N个数的排列，已知其中P(P <= N)个位置上的数，求共有多少个这样的排列——在P个位置上具有已知的数，且具有上述性质A。例如：N = 4，且已知第1位、第2位分别是1和4，则1432，1423就是这样的两个排列。

输入

文件第一行为N（N <= 50）。

第二行为N个数，未知数用0表示。

输出

输出满足上述条件的排列总数。

示例：

输入示例

4

1 4 0 0

输出示例

2

分析

题目具有相当强的隐蔽性，序列需要满足的条件比较复杂，使人无从下手。我们试探性的写出一些序列，观察其规律：

当n = 5 时，有16组满足要求的序列：

1 2 3 4 5	1 2 3 5 4
1 2 5 3 4	1 2 5 4 3
1 5 2 3 4	1 5 2 4 3
1 5 4 2 3	1 5 4 3 2
5 1 2 3 4	5 1 2 4 3
5 1 4 2 3	5 1 4 3 2
5 4 1 2 3	5 4 1 3 2
5 4 3 1 2	5 4 3 2 1

仔细观察发现（观察发现是一个蕴含了无数的奥妙、技巧和乐趣，但却不是能用文字能表达的过程）：任何一个排列的后 k 位（1<=k<=n）是若干连续整数组成的集合。例如：51234。后一位是{4}；后两位是{3,4}；后三位是{2,3,4}；后四位是{1,2,3,4}；后五位是{1,2,3,4,5}。其余排列也满足此规律。

于是我们可以大胆的提出猜想：

猜想 1 任何一个排列的后 k 位（1<=k<=n）是若干连续整数组成的集合。

证明 约定序列的第 i 位用 v[i]表示。设序列后 x 位（x>=1）是若干连续整数，这些整数构成集合{s..t}。那么倒数第 x+1 位上的数 v[n-x+1]必然等于 p-1 或者 p+1（p∈{s..t}）。显然 v[n-x+1]∪{s..t}={s-1..t}或者{s..t+1}，还是若干连续整数。

根据“证明”，我们进一步猜想：

猜想 2 如果一个排列的后 k 位（1<=k<=n）是若干连续整数组成的集合，则这个排列符合题目要求。

证明 约定该序列的第 i 位用 v[i]表示。设{v[n], v[n-1], …, v[n-k+1]}={s..t}。因为{v[n-1],…,v[n-k+1]}也是若干连续整数的集合，所以 v[n] = s 或 t。

如果 v[n] = s，那么必有：s+1∈{ v[n-1],…,v[n-k+1]}

如果 v[n] = t，那么必有：t-1∈{ v[n-1],…,v[n-k+1]}

即这是一个满足要求的序列。

因此问题转化为：求后 k 位（1<=k<=n）是若干连续整数组成的集合的排列总数（由 1，2，……，N 组成）。

接下来的工作就好办了。根据两个猜想，我们很容易想到动态规划。设 g[s,r]表示满足下面条件的序列 C 的总数：

C 由集合[s..s+r-1]中的数组成，且后 k 位（1<=k<=r）是若干连续整数组成的集合。如果原问题中倒数第 i 个位置上的数已经确定为 x（1<=i<=r），那么 C 的倒数第 i 个位置上的数也要是 x。

则：

$$g[s,\ r] = \begin{cases} g[s+1,r-1] & \text{如果倒数第 r 位确定为 s} \\ g[s,r-1] & \text{如果倒数第 r 位确定为 s+r-1} \\ g[s,r-1]+g[s+1,r-1] & \text{如果倒数第 r 位不确定} \end{cases}$$

0　　其他情况

g[s, 1] = 1

求：g[1, n]

算法时间复杂度为 $O(n^2)$。

本题小结

此题是湖南省队集训试题，具有相当难度的题目。主要是因为给定的条件比较复杂，很不便于转化利用。因此在考场上，诱使大多数选手使用搜索。（当时只有两名选手使用了速度较快的动态规划）

本题的突破口在于两个“惊世骇俗”的猜想的提出。通过这两个猜想，将很难直接利用的条件转化到我们便于使用的熟悉形势，为后面使用动态规划，打开了通路。

然而从“列出简单数据”□“提出猜想”还是有一段相当的距离。没有丰富的经验积累、解题灵感、观察能力、概括能力、联想能力是很难发现什么有价值的结论的。

§3.3 归纳小结

本节介绍了归纳法的基本分类、方法，以及在信息学中的应用实例。

归纳是除类比以外，得到猜想的一个重要方法。它的基本思想是从特殊□一般，从简单□复杂，从小规模□大规模。也是一种“化归”的思想。

另外值得注意的是，归纳不一定是直接归纳猜想出问题的结果。可能像【例4】那样，根据归纳而得的猜想不一定与最后的结果直接挂钩；但猜想的结论却对解题思路、解题方向的发展有重大帮助。

进行归纳猜想时要注意发散思维，充分想象——往任何可能的方向去想、去靠。不要让转瞬即逝的灵感溜走。

4、总结

其实我们解题时都在不知不觉的猜想——猜想问题的性质；猜想模型的形式；猜想算法的内容……猜想是从已知向未知探索的重要途径。

猜想≠偶然+运气。猜想是建立在观察、分析、联想、归纳基础上的一种主观不充分似真推理；是观察能力、概括能力、联想能力、创新能力的综合体现。猜想的水平，最能反映一个人思维品质的高低。

根据得到猜想方式的不同，又可以分为：类比猜想和归纳猜想。上文已经详细介绍。

但无论那种方法都离不开观察、分析和实践。可以说：观察是猜想的血液——贯穿于整个猜想过程之中；分析是猜想的灵魂——提出猜想的前提和基础；实践是猜想的骨架——支撑猜想的基石，脱离了实际，猜想毫无意义。

随着信息学竞赛试题向隐蔽化、多维化、抽象化的方向发展，试题难度越来

越大，对选手创新能力的要求也越来越高。猜想这一重要的解题策略，在其中扮演的角色也必将越来越重要。

【参考文献】

1. 欧阳维城，《初等数学解题方法研究》，湖南教育出版社

2. 2000 年湖南集训试题

3. NOI2000 试题

4. 中国基础教育网（www.cbe21.com）的有关资料