**PART 1 - 引入**

先来考虑这样一些有趣的问题：

一条始终水平的绳子剪三刀，最多能剪成多少个部分？

一块煎饼切五刀，最多能切成多少块？

一个蛋糕切四刀，最多能切成多少部分？

这些问题都指向了同样的一个主题：切割。

第一个问题可以这么看：在一条直线上画三个点，最多能分成几个部分？

第二个问题可以这么看：在一个平面上画五条直线，最多能分成几个部分？

第三个问题可以这么看：在一个空间上画四个平面，最多能分成几个部分？

因此这些问题似乎可以统一成一个问题：

在一个r维的东西上作n次分割（Division），最多可以分成多少个部分？

不妨将这个问题的答案记为，直线就是r为1的情况，平面r为2，空间r则为3，同时可以注意到，用于分割的“刀”是一个维的东西。

由此转化，上面三个问题便是求的值。

更加一般的，我们试着想象高维的分割：

一个三维空间，是否也会将四维空间分为两部分？

同样的，更高维情况下的“分割”会是怎样的情况？

为此，我们需要研究一切，其中n为非负整数，r为正整数。

当时，情况十分简单，n个点将直线分为了个部分，故显然有。

但是，当时，情况就没有那么显然了。

**PART 2 - 平面的分割**

我们先计算一些n较小时的情况：

当再加一条直线时，如果你愿意仔细数一数，则会发现。

在这里停一下，给自己一点时间，你发现这个数列的规律了吗？

也许直接看不那么明显，但是我们作出相邻两项的差，便能得到一个十分和谐的数列：

按照这个规律，我们似乎可以继续写下去：

那么，规律是否真的如我们猜测的一样简洁呢？

为此，我们需要找到增加一条直线时增添的区域数。

设平面上已有k条直线，作图可以发现，当增添区域最多时，第条直线需要与前面每条都相交，这时，它一共经过了个区域，也就将这么多个区域每个划分为两块，增添的区域数就是

可喜可贺，我们的规律是正确的。由此便可以知道，，等差数列求和后便成了。

但是这样一维一维得做下去，始终无法达到我们想要的任意维度的效果。更何况，三维时的情况已经十分难以直观看出了。不过，从刚才的做法中，我们似乎得到了一点思路：新增直线经过的区域数等于其增添的区域数。利用这个，也许我们可以推出一些结论。

为了更明显地揭示结论，我们来看一个具体例子。

**PART 3 - 与递推的出现**

当维度为3时，一些较小的情况仍然可以直观得到：

但是，当来到，就算我们画出图，也很难数清区域数，为此，我们必须寻求别的方法。

由划分平面时的启发，我们只需要知道增添的区域数就够了，而这意味着，我们需要知道这个新增的平面经过了多少个区域。

我们采取一种十分特殊的思路：

将第四个平面放进去，再取出来，看看这个平面会怎么被已有的三个平面切割。

再次给自己一点时间，看看能不能自己画出这个平面上的情况。

由于这个平面必须与其他三个平面都相交成一条直线，这个平面上留下的“刀痕”必然是三条不同的直线。又因为另外三个平面也需要以最多区域的方式进行划分，这三条直线必然两两相交于三个不同点。

有趣的是，这些直线将平面划分为的区域数，恰好就是平面经过的不同区域数。这个判断成立依赖以下两点：

同一个区域内，平面不可能被切分开；

平面的每一个被切分的部分，一定属于一个区域。

于是，这些直线将平面划分的区域数，就是新增的区域数！

那么直线究竟将平面划成了几个区域呢？

很显然，这是上一部分已经解决的问题，这里我们直接给出结果：

更巧妙的是，用同样的道理，我们似乎还可以把这个式子继续写下去：

我们甚至可以发现，这个规律的适用不只是在三维空间。运用这样考虑切割部分的方法，我们可以在更高维的情况实现一摸一样的操作，于是我们有了关键的递推公式：

我们将所有的这样列成一个表格：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n**  **r** | 0 | 1 | 2 | 3 | 4 | …… |
| 1 | 1 | 2 | 3 | 4 | 5 | …… |
| 2 | 1 | 2 | 4 | 7 | 11 | …… |
| 3 | 1 | 2 | 4 | 8 | 15 | …… |
| …… | …… | …… | …… | …… | …… | …… |

这个递推也就是说，表格中的每个元素等于它左侧的元素与左上的元素之和。

由于表格的第一行，而第一列显然有，用递推我们已经可以得到任意元素的值。

例如求，可以由这样的路径：

于是，问题在此已经变成了单纯的计算问题！

不过，为了算出最终结果，我们还要做一些知识的储备。

**PART 4 - 组合数**

我们再把递推公式放在这里：

有的观众可能会意识到，有一类数有着和这个极为相似的递推，它们就是大名鼎鼎的组合数。

所谓组合数是指这么一类数，它们被记为，代表从n个元素中不重复地选出k个元素的方法种数。举个例子，就代表从1234里选出两个数，可以是12，13，14，23，24，34，一共六种选择，因此。

事实上，有着精确的公式：

其中的感叹号表示阶乘，，特别地，规定。

不过为了今天的推导，我们不妨只记住它的原始意义：从n个元素中不重复地选出k个元素的方法种数。

这里值得注意的是，n是正整数，而k是非负整数。当k为0时，也就是“什么都不取”的取法，我们认为是一种。而当k大于n时，显然是不可能取出的，因此我们将它规定为0。

我们接下来就用定义来证明两个式子：

第一个，。

想象这样的场景：

我们需要从个不同的球中取出个球。我们将其中一个球涂红，那么，如果取出这个球，则需要在剩下的n个球中取k个（），如果不取，则需要在剩下的n个球中取个（）。因为这个红球要么取要么不取，两种情况的种类相加便是全部情况（），这便得到了公式的证明。

这里值得第三次停顿下来，自己去验证k大于等于n时的情况，最终便能发现，这个式子可以对所有合理的n与k成立。

第二个，。

左侧包含了从n个小球中取出任意多个数的情况，因此最后的结果，应该为从n个小球中取出一部分的情况数。

从另一个角度来看，既然是取出一部分，每个小球都有取与不取两种情况，一共n个小球，全部相乘恰好是。

由这两个组合数公式，我们终于可以得出结果了。

**PART 5 - 结果的计算与探索**

自然，是由递推唯一确定的值，所以对于我们猜测出的结果，只要验证它满足首行、首列与递推，就一定是一个正确的结果。

事实上，这个结果并没有那么容易猜测。首先，能联想到组合数就利用了下面两个式子的相似性：

紧接着，我们还要利用已经算出的一些结果，进一步猜测它究竟是怎样由组合数表示出来的。

完成这两步后，我们还需要代回原本的递推验证，才能最终确信我们的答案。

这三步在猜测的过程中缺一不可，而当你终于完成后，你将得到一个美妙的式子：

首行与首列的情况请观众自己试着验证，我们这里简单验证一下它符合递推：

到这里，题目似乎就做完了，不过如果你是个有心人，也许能在表格中发现一些其他的规律，比如：

那么，最后一次停下来，试着自己用组合数时推导的第二个公式证明这个结论吧。

同样，这里给一个简单的证明：

最后的最后，关于这道题其实还能有些其他的思考，例如：

高维空间中这样的分割究竟是如何被严谨定义的？

是否一定可以实现这样的分割来达到理论最大值？

达到最大值的条件怎样抽象表述？

限于视频时长，这里不能一一解答，但只要能一直带着探索的好奇心，相信总有一天你将能够自己解决这些疑问。

感谢观看！