从《小H 的小屋》的解法谈算法的优化

安徽师范大学附属中学　　　　杨弋

题目大意

小 H 有一个院子，东西方向长为 100 单位。东墙和西墙均平行于 y 轴，北墙和南墙分别是斜率为 k_1 和 k_2 的直线。北墙和南墙分别围有多块草坪，每块草坪都是一个矩形，矩形的每条边都平行于坐标轴。相邻两块草坪的接触点恰好在墙上，接触点的横坐标被称为它所在墙的“分点”，这些分点必须是 1 到 99 的整数。北墙要有 m 块草坪，南墙要有 n 块草坪，并约定， $m \le n$ 。如果记北墙和南墙的分点集合分别为 X_1, X_2 ，则应满足 $X_1 \subseteq X_2$ 。

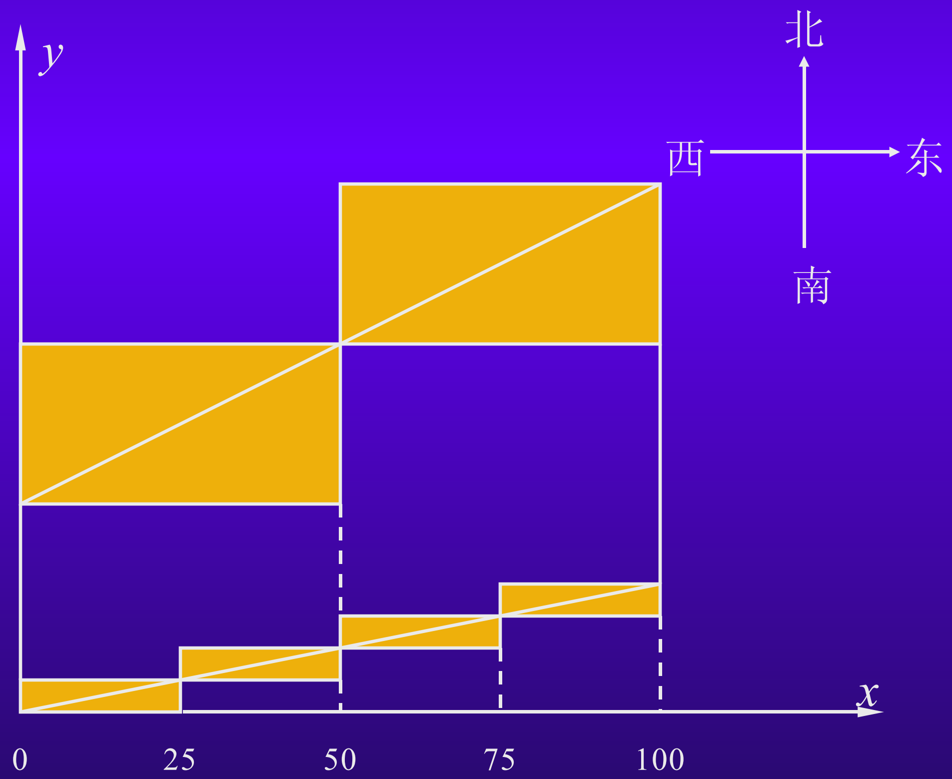
输入 k_1 ， k_2 ， m ， n 。k_1 和 k_2 为正实数， m 和 n 为正整数，且 $2 \le m \le n \le 100$ 。

假定南北墙距离很远，南墙草坪和北墙草坪不会重叠。

题目大意

让我们来看一个例子：

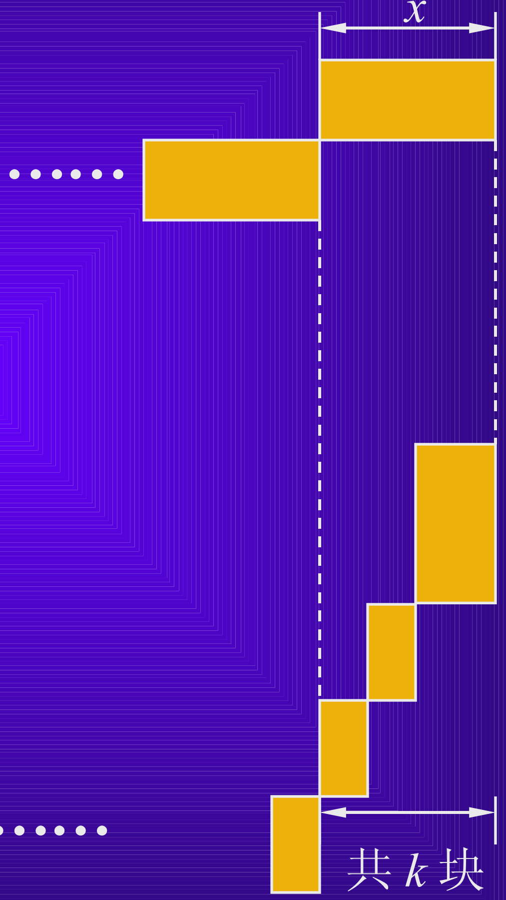
输入： 0.5 0.2 2 4



图中给出的就是这组数据的最优解，最小面积为 3000 。

算法一

- 看到题目，我们首先想到的算法是动态规划。
- 我们用 $f(w,u,v)$ 表示长度为 w，北墙 u 块草坪和南墙 v 块草坪时的最小面积。
- 令一块北墙草坪和其对应的南墙草坪为一个“块”，若北墙草坪长度为 x，南墙草坪块数为 k，则该块最小面积为 $area(x,k)$。



$$f(w,u,v) = \min\{f(w-x,u-1,v-k)+area(x,k)\}$$

算法一

这样，我们就得到了算法一：

1 、枚举所有合法的w,u,v ，对于每一组 w,u,v, 计算 $f(w,u,v)$ 。

2 、在计算 $f(w,u,v)$ 时，枚举所有合法的 x,k ，从而求出 $f(w,u,v)$ 。

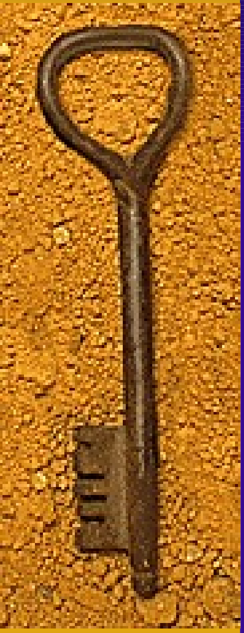
3 、 $f(100,m,n)$ 就是我们所要求的解。

该算法时间复杂度是

O(

太慢了！

空间复杂度是 O(ln)

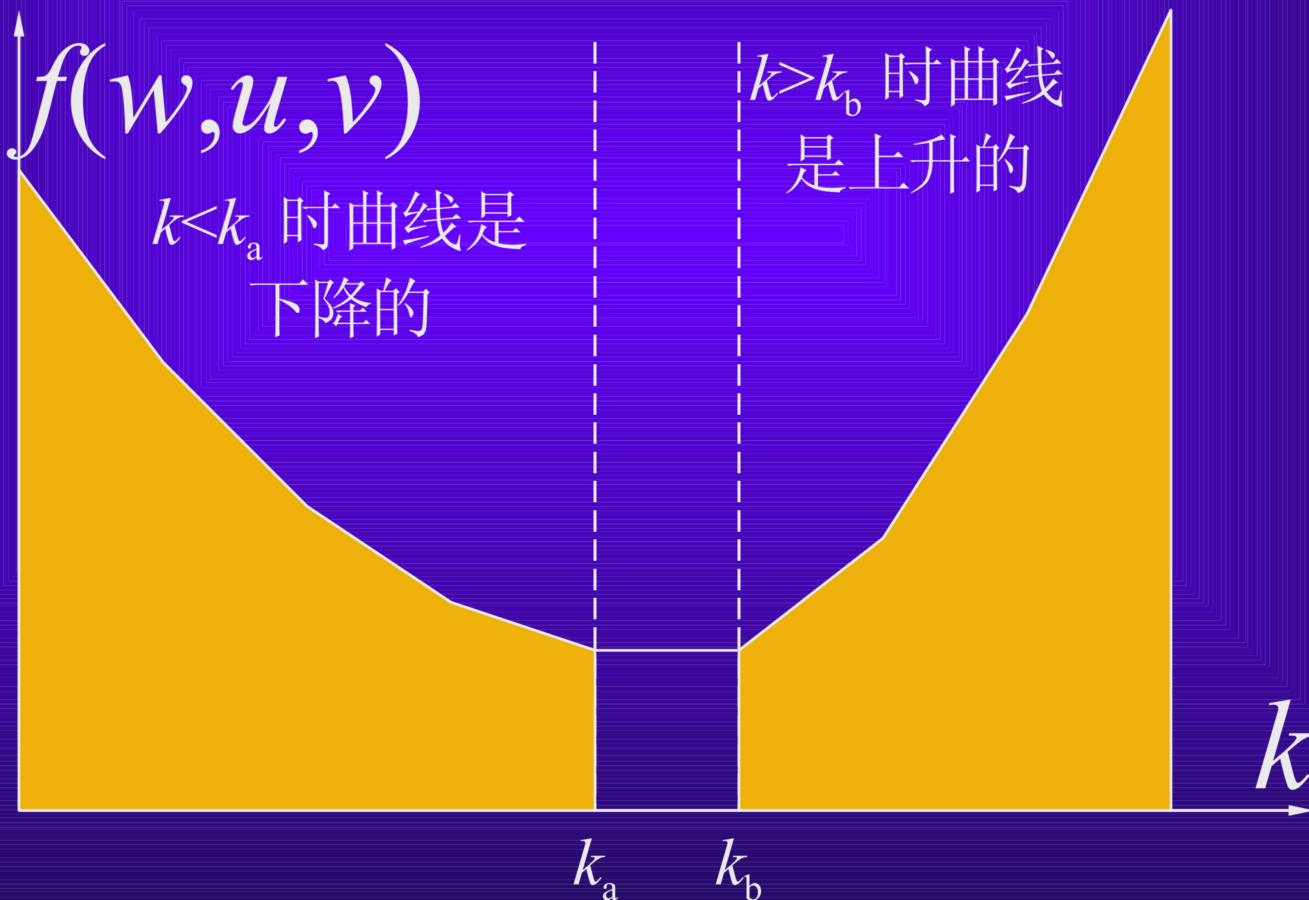


算法一

看来，我们必须优化我们的算法……

算法二

在状态转移时，假如我们已经确定了 x……



算法二

- 我们枚举 x ，当 x 增大的时候， k_a 不会减小。
- 这样，我们可以枚举 x 的同时计算 k_a 。

由于 x 和 k_a 都是递增的，该算法的时间复杂度降为 $O(l^2mn)$ ，空间复杂度仍然是 $O(ln)$ 。

算法二

我们注意到计算 $f(w,u,v)$ 的最优解所用的 x,k 必不小于计算 $f(w,u+1,v)$ 的最优解所用的 x,k（假如 $f(w,u+1,v)$ 存在的话），所以我们可以记录计算最优解时用的 x,k，从而进一步减少程序运行所用时间。

算法二无论在时间复杂度上还是在空间复杂度上都足够应付这道题目的所有数据了。

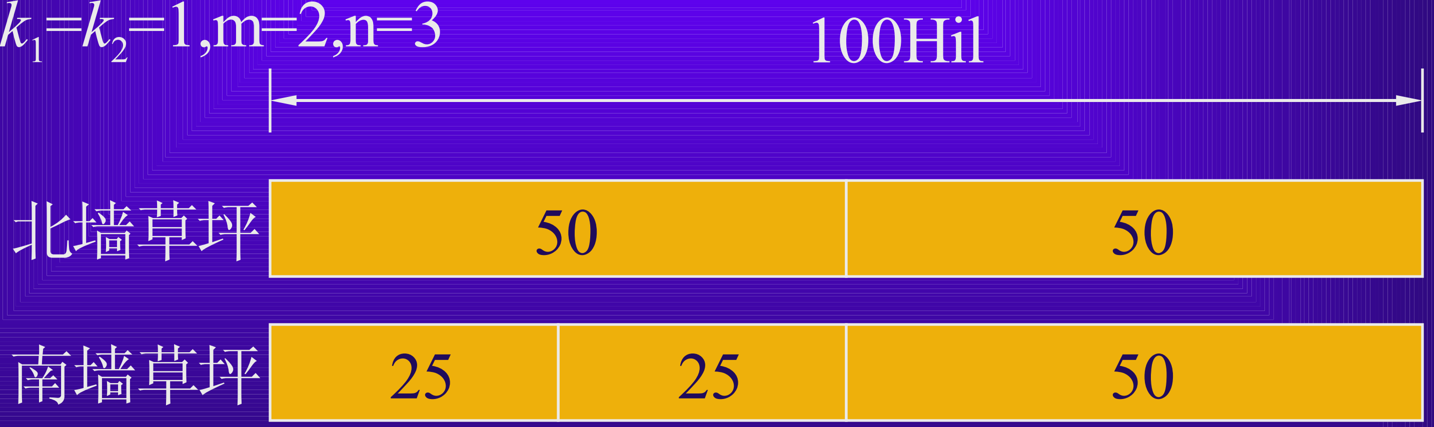
那么，究竟有没有更好的方法呢？

算法三

现在让我们彻底抛弃动态规划的思想，来试一试贪心算法在此题中的应用。

我们尽可能平均地分配北墙草坪的长度
然后再尽可能平均地为南墙草坪分配长度

$k_1=k_2=1, m=2, n=3$



$S=8750$

算法三

这样的贪心算法是错误的！

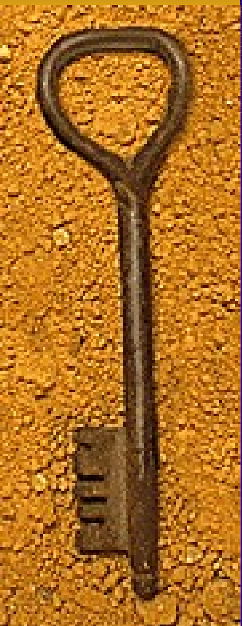
但是……

$k_1=k_2=1$,m=2,n=3

100Hil

北墙草坪	57		43
南墙草坪	28	29	43

$S=8572$

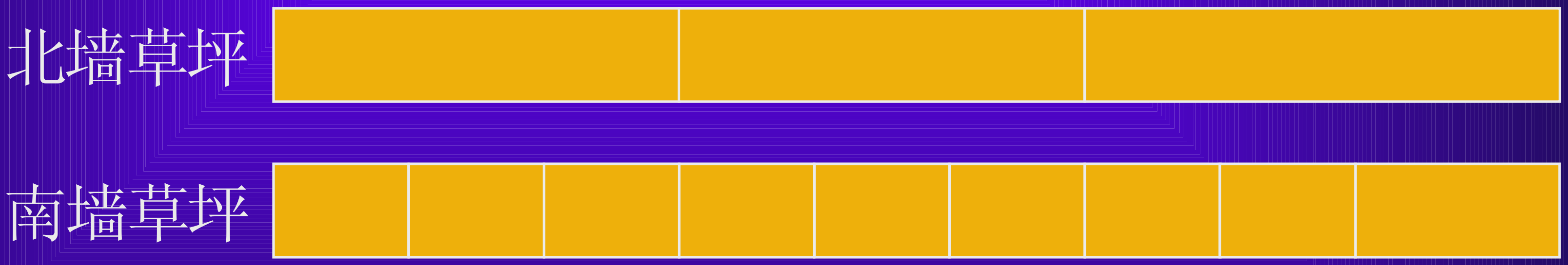


算法三

为什么说这个贪心算法是错误的？

因为每一块北墙草坪对应的南墙草坪数量并不是相等的！

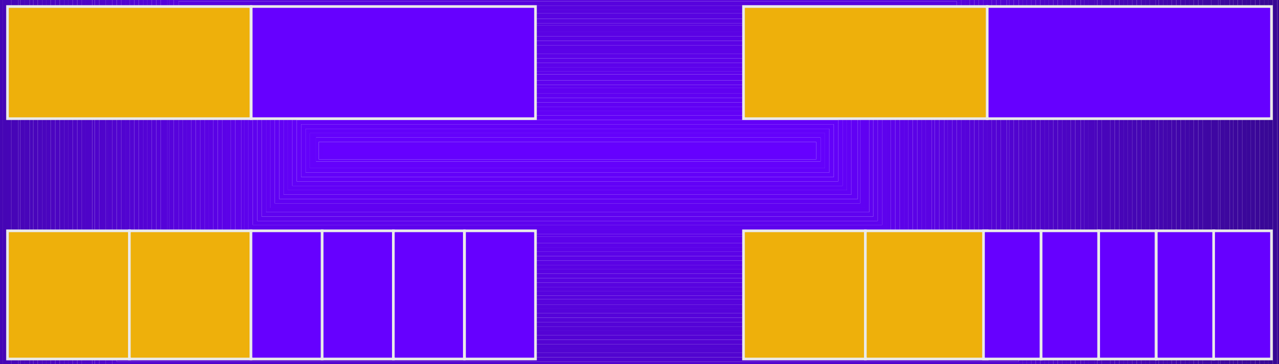
如果我们能使每一块北墙草坪对应的南墙草坪数量相等……



显然，这时尽量平均分配得到的就是最优解了

算法三

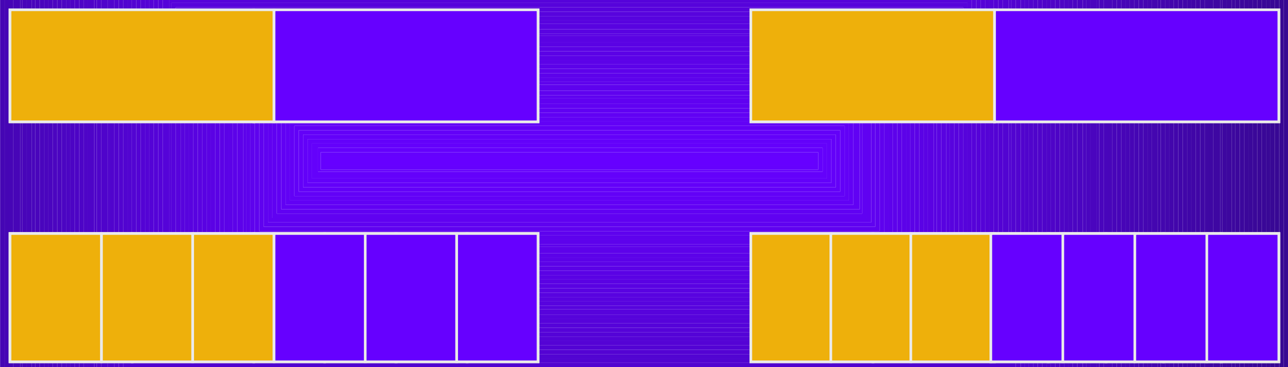
现在我们假设最优解有这样的情况



北墙草坪对应的南墙草坪块数的差距大于 1 了。

算法三

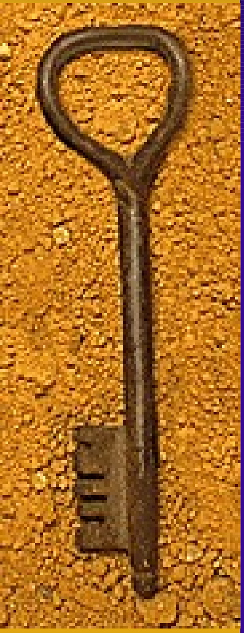
那么……



总存在这样的调整，调整以后差距会小于等于 1，并且总面积不会比原来大。

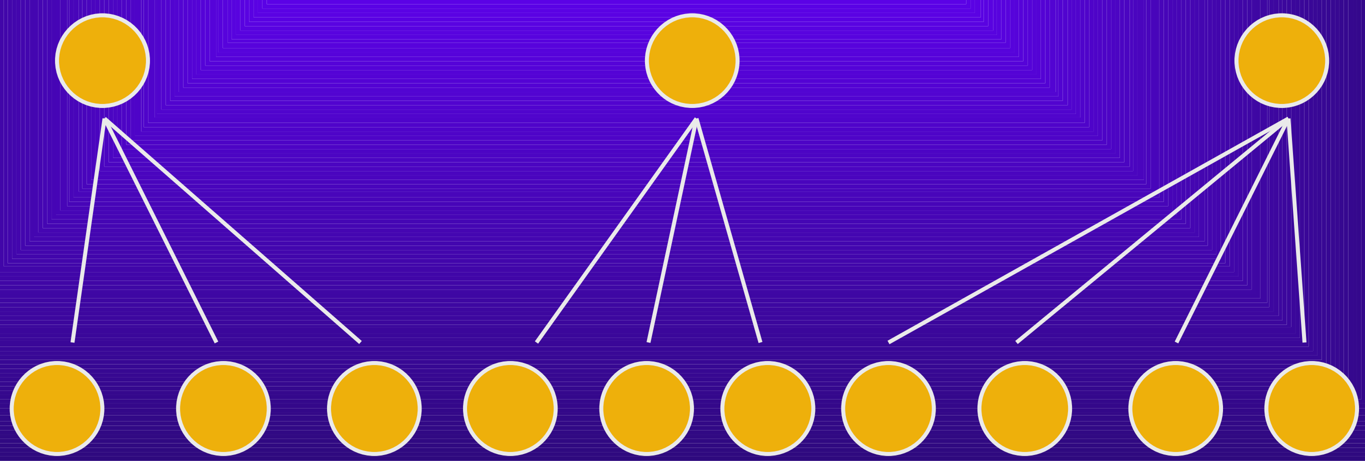
我们可以把南墙草坪按块数尽可能平均地分配给北墙草坪！

算法三



这里是一个例子

我们有 3 块北墙草坪和 10 块南墙草坪，该怎么分配呢？



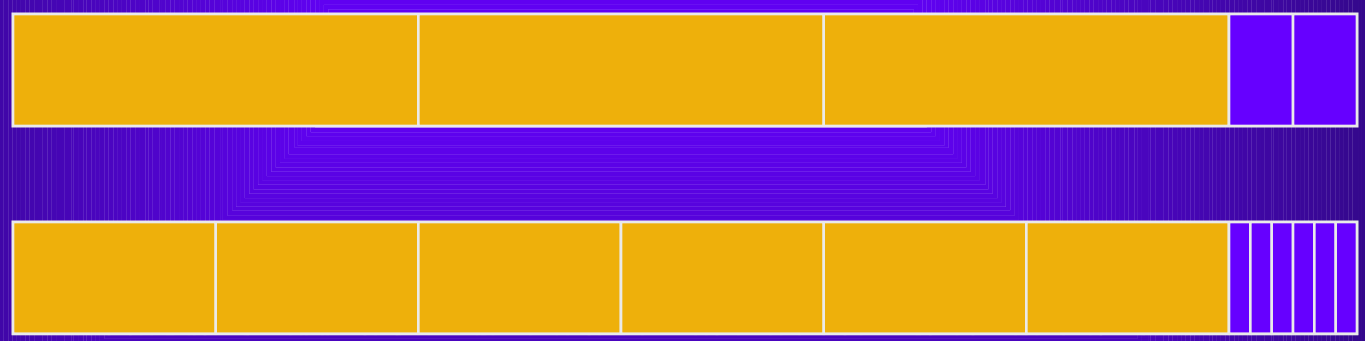
算法三

于是，我们得到了一个全新的算法：

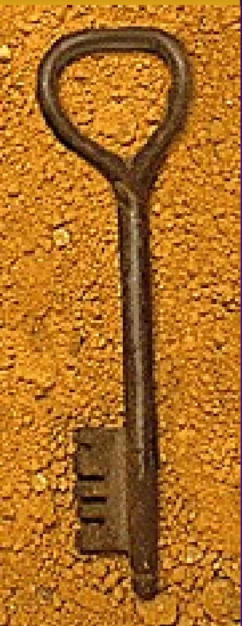
- 我们先把整个墙分成两部分，第一部分包含 m_1 段北墙草坪，每段对应 n_1 段南墙草坪；第二部分包含 m_2 段北墙草坪，每段对应 n_2 段南墙草坪。如果两部分的长度确定了，在两部分内，都可以直接贪心求该部分的最小面积 。
- 我们枚举其中一个部分的长度，另一个部分的总长度也就确定了 ，此时的最优解可以在 O(1) 时间内计算出来。

算法三

就像这样：



这样，整个算法的时间复杂度是 O(l-n) 的，空间复杂度是 O(1) 的。



算法四

我们的优化还没有结束！

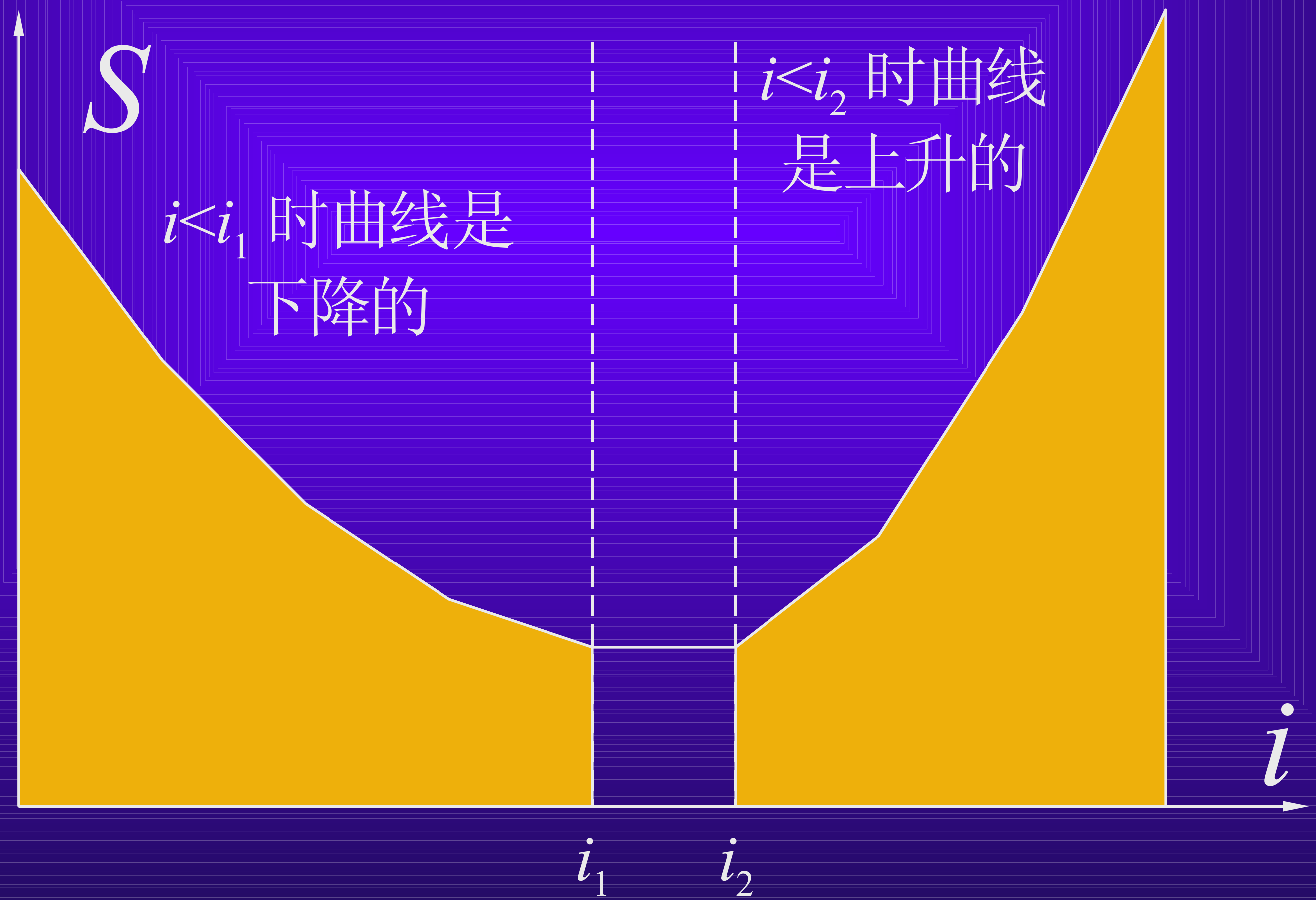
现在让我们来看看算法四吧

我们设分给第一部分的东西向长度为 i 。设能使S取到最优解的最小 i 为 i_1 ，能使 S 取到最优解的最大 i 为 i_2 。

和我们在思考算法二时的情况类似，我们发现 $i<i_1$ 时 S 随 i 增大而减小， $i>i_2$ 时 S 随 i 增大而增大，在 $i_1 \leqslant i \leqslant i_2$ 时 S 始终能取到最小值。

算法四

就像这样



算法四

所以，我们只需要找到一个 i_0 ，使得 i 取 i_0-1 和 i_0+1 时总面积都不会变少，那么此时总面积就是要求的最小值。

但是，这个优化并不能使整个程序的速度有很大的提升。

算法四

我们现在去掉“草坪长度是整数”这一限制，然后再来思考本题。

我们令 $q \quad \frac{k_1}{m_2} . \frac{k_2}{m_2 n_2}$ ， $p \quad \frac{k_1}{m_1} . \frac{k_2}{m_1 n_1}$

我们不难得出，i 取 $\frac{ql}{p.\ q}$ 时总面积最小

现在，我们再回到原问题中

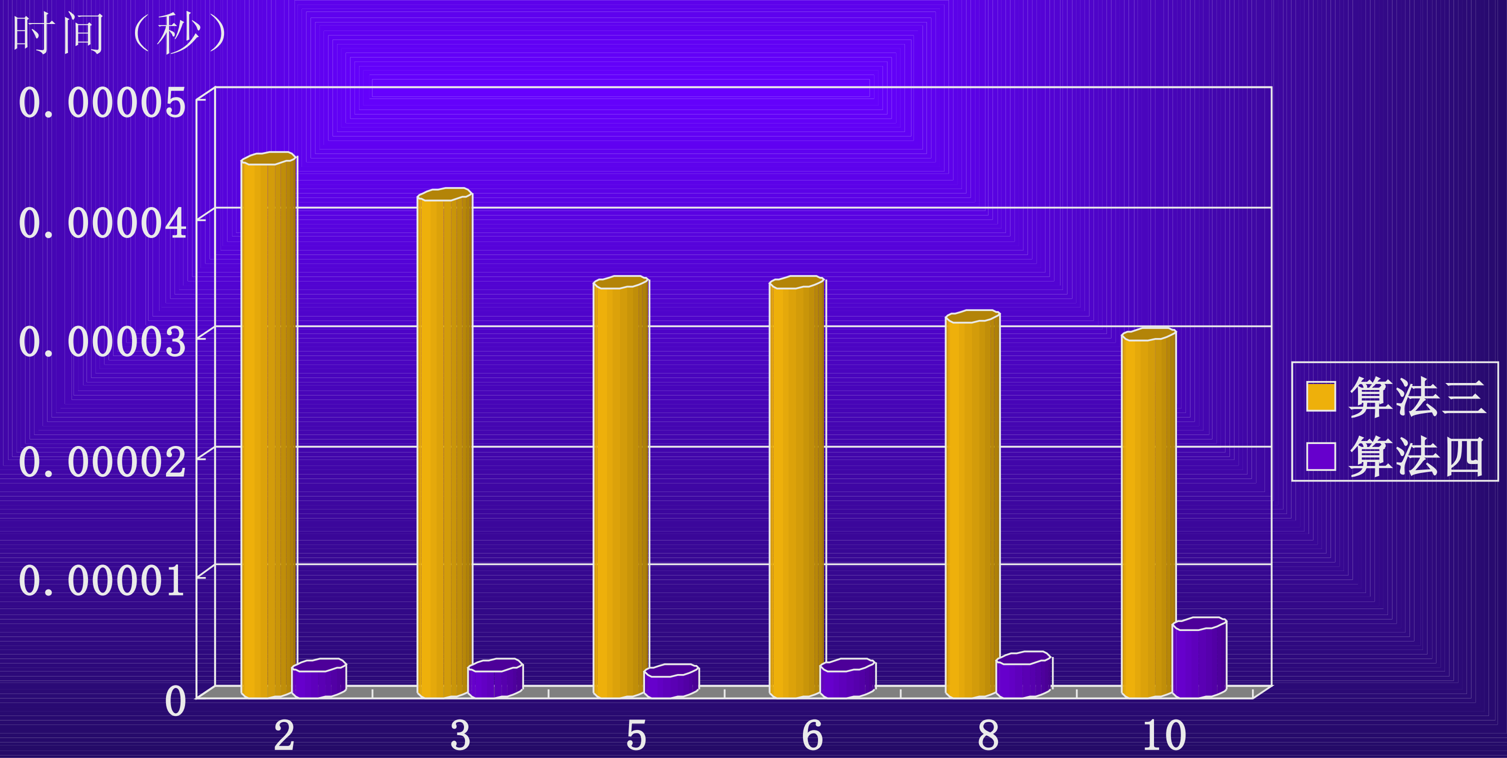
最优解的 i 一定是接近 $\frac{ql}{p.\ q}$ ，的。我们不妨先令 $i = \left[\frac{ql}{p.\ q}\right]$

然后再不断地加 1 或减 1 以使解更优，直到求出最优解

算法四

由于 S 和 i 满足前面给出的那个关系，因此算法四是正确的。

虽然算法四的时间复杂度不好分析，但经过测试，我们发现算法四要比算法三快很多！



总结

我们在解决本题的时候，首先想到的算法是动态规划，通常解决这类试题的方法也就是动态规划。并且，我们通过优化状态转移使动态规划的性能得到了极大的提升。

然而，动态规划是一个适用面很广的算法，在本题中，它就并不是最优秀的算法了。因为本题的一个很重要的思想“均分”在动态规划中并没有得到很好的体现。

总结

而贪心算法却能很好地体现均分的思想。因此，最终我们大胆地放弃了动态规划算法而使用贪心算法来获得更高的效率，而这种突破常规的做法最终给我们带来了速度上极大的提高。

我们在优化贪心算法的时候，再一次抓住了题目的特点，找出了 S 和 i 的关系，并由此找出一个方法，能快速地找出一个接近最优解的方案。

在不断的优化中，我们最终将 $O(l^2mn^2)$ 的时间复杂度降至 $O(l\text{-}n)$ 以下，空间复杂度也降至 $O(1)$ 。

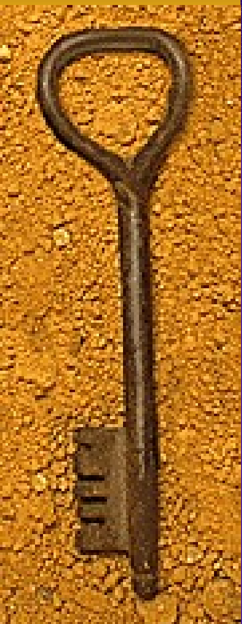
总结

由此我们看出，我们在解题过程中应该：

抓住题目的特点

大胆猜想和尝试

敢于突破常规



谢谢