

# 匹配

## 算法一

直接根据题意暴力搞，时间复杂度 $O(n^2)$ 。期望得分60分，实际得分70分。

## 算法二

注意到 $B$ 串如果最后一位和 $A$ 串对应位相同，那么答案就是 $\min\{l_a, l_b + 1\}$ 。否则从后往前判，不相等就停下来，可能会快一些。期望得分60分，实际得分70分。

## 算法三

注意到 $B$ 串和 $A$ 串除了最后一位之外并没有什么区别，而 $B$ 串和 $A$ 串全部匹配的情况显然很好判断。所以我们要求的就是 $B$ 串的最长前缀与对应后缀相等。这就是 $kmp$ 算法可以解决的问题。直接写 $kmp$ 即可通过。时间复杂度 $O(n)$ ，期望得分100分，实际得分100分。

## 算法四

其实我们可以通过优化比较字符串来通过这道题。因为朴素的比较是 $O(n)$ 的，然后只需要把字符串哈希起来，就可以把比较优化到 $O(1)$ ，于是就可以通过这道题了。

为什么暴力可以跑70分？主要还是因为我卡不掉.....

# 回家

## 算法一

根据题意暴力搞，依次删除每个点然后判断1号点和 $n$ 号点是否联通。时间复杂度 $O(n^2)$ ，可以获得30分。

## 算法二

注意到第11、12个点输入是一颗树。而树上两点之间的路径是唯一的。把1到 $n$ 路径上的点统计一下个数，全部输出即可。

## 算法三

其实算法二是在提示我们。考虑对于给定的 $m$ 条边，可以先随便构一颗树出来，于是我们就得到了一颗树和一些非树边。当没有非树边时，算法二即可解决。现在有了非树边，我们就考虑一下每条非树边有什么影响。我们可以考虑先把1到 $n$ 路径给提取出来，于是整个图形就变成了一条链，每个点下面挂了若干个点。

显然，一开始必经点我们是可以统计出来的。然后，每一条非树边都会可能使一些原本是必经点的点变为非必经点。于是，我们可以把这条1到 $n$ 路径上的点进行重标号，每个不在路径上的点编号就是 $dfs$ 到的最近的路径上点的编号。于是，这就变成了一个序列上的问题了。每条非树边假如对应的区间是 $[l, r]$ ，那么路径上区间 $(l, r)$ 对应的点显然就变为了非必经点。于是我们只需维护区间赋值的操作即可。用线段树可以解决。

时间复杂度 $O(n \log n)$ ，可以获得80到100分不等。

## 算法四

我们接着考虑算法三。注意到只有区间赋值操作，所以实际上没有必要使用线段树。我们只需把区间全部存下来，最后排遍序扫一边即可。

时间复杂度 $O(n \log n)$ ，可以获得80到100分不等。

## 算法五

这个算法是另外一种思路。我们发现如果一个点是必经点，那么把它删除之后原图一定变得不连通了。于是必经点肯定是无向图的割点。但是割点并不一定都是必经点，因为一个割点不一定把1号点和 $n$ 号点分到两个不同的块去。

所以，我们考虑如何改进这个算法。实际上，在使用*Tarjan*求割点的时候，我们得到的是一颗*dfs*树和一些返祖边。于是，我们可以从1号点开始*dfs*，同时记录哪些节点的子树中有着 $n$ 号点。如果这些点是割点，那么它们就是必经点。于是一遍*Tarjan*即可解决。

时间复杂度 $O(n + m)$ ，可以得到100分。

其实这道题我本来是想把算法三和算法四都卡到80分的.....后来发现我根本卡不掉.....

# 寿司

## 算法一

爆搜，枚举所有走法。可以获得0到20分不等。

## 算法二

注意到每种状态实际上是可以压缩的。所以搜索的时候对于较小的数据把状态记录下来，用最小表示法去个重，对于稍大的数据可以用IDA\*搜索，对每个状态进行估价来进行加速。

可以获得20到40分不等。

## 算法三

一般环上的问题一般都要把序列复制一遍，然后转成了序列上的问题。这道题也不例外。经过观察后我们显然可以得到一个结论：对于最优解，一定有一个断点使得所有的交换都不经过这个点。因为如果所有相邻的位置都交换了一次，那么这些操作并没有改变这个环的形态，也就是没有任何作用，完全可以直接去掉。

于是，我们就可以枚举断点，然后就转变为了一个序列了。我们的目的就变成了使一种颜色的寿司全部靠边。然后可以预处理出每个寿司移动到左边界需要多少步，右边界需要多少步，取 $\min$ 后加起来即可。

时间复杂度 $O(n^2)$ ，可以得到40分。

## 算法四

发现对于一个序列，肯定是左边一部分往左靠，右边一部分往右靠，于是可以每次二分出这个边界点。

时间复杂度 $O(n \log n)$ ，可以得到80到100分。

## 算法五

如果算法四你都想出来了，那么算法五应该顺理成章的出来了。当断点顺时针移动的时候，分界点显然不会逆时针移动。于是这玩意儿是有单调性的，弄一个单调指针扫一扫就可以了。

时间复杂度 $O(n)$ ，可以得到100分。