

玄学一些不可思议的例子

olphe的代码

我都不知道他是怎么过的...这数据不知道是怎么造的

随手卡掉

```
7 7
aaaaaaa
aaaaaaa
aaaaaaa
aaaaaaa
aaaaaaa
aaaaaaa
aaaaaaa
```

Output  
NO

Answer  
YES (显然)

但这个算法还是有一些可取之处的，它的核心思想是：当你用每行的所有字母个数，相同的行的奇偶性等东西卡一波NO之后，当数据往大了走后，你很难构造出一组无法被前面这些剪掉的NO的数据，所以他最后暴打YES

一些较好的例子

- maryanna2016,kawatea,PECaveros的代码，均预测1ms
- 这部分代码都很好的做了预处理：例如r[i][j]表示第i行和第j行能否匹配，c[i][j]表示第i列和第j列能否匹配，这些都是显然的可以在搜索之前做掉的事情
- 我的代码里还做了一件事情：pr[i]表示第i行是否可能回文，pc[i]表示第i列是否可能回文；当然，你可以用搜索的时候先把单独的回文那行先去掉的方法来获得同样的优化
- dreamoon的代码没有预处理，但中间也有很多剪枝，效率尚可，预计5ms+

一些不太好的例子

- august14的代码，中途没有剪枝，最后的比较比较残暴，不看好它的速度
- 预测：100ms+
- kmcode的代码，写的比较丑，比较长，比较难理解，中途用了set，不看好效率预测：

一些很不好的例子

- mamekin的代码，没有预处理，用rollinghash来在最后check,hash是有常数的，况且还要取模，而且拉链法竟然用map(也是有原因的模数太大数组存不下) 又多了一个log
- 预测：100ms+
- kmjp的代码，比较残暴，中途没有任何剪枝，最后用set来判重，set里存的都是string，常数自然大到飞起
- 预测：500ms+
- eddy的代码，能优化的地方他都没有优化