A. 双剑合并

题意

给你两个序列分别有n和m个数，让你在两个序列中找到两个是的他们的异或值最大。$n,m \leq 1e6, a_i \leq 1e9$.

分析

数据范围很大，暴力n^2的做法肯定不能用，所以就要考虑$XorTrie$来优化。所以只需要在Trie中先存入每一个a_i的二进制的01串，要反向倒着存，然后再将每一个b_i在Trie中扫一遍，每一次优先找当前位相反的位置，想着$digit[i] xor 1$的位置走就行了。这个贪心的正确性是很容易证明的。因为每一个数都是倒着存进去的，假设吧第i位变成1，可以增加2^i，如果不这样走的话，哪怕之后的所有低位全部加起来，即$\sum_{j=0}^{j=i-1} 2^j$也是小于2^i的。

思考

要注意异或的性质二进制表达下的特点，利用$XorTrie$可以使得达到$logn$的复杂度的优化，值得注意。

B. 单词替换

题意

把字符串中的子串A换成子串B。

分析

水题

思考

水题

C. 01的时间

题意

找到一个数n的最小的只包含0和1的倍数。$n \leq 466$。

分析

看到这题第一眼就是把所有的只包含0和1的用bfs找出来，然后碰到的第一个能够整除n的就是答案。结果发现答案会爆$longlong$，所以我就去用$JAVA$的$BigInteger$来写，然后又爆内存，没办法，直接把表打出来了，所以代码略长→_

$\rightarrow$。

思考

其实正解应该是用数组来保存每一位，不停地添加**0**和**1**，然后还需要用数组来保存余数来剪枝。因为数字很大，所以采取边乘边模的方法。

E. RunningPhoton's Nightmare

题意

给你一张$m*n$图，给定起点和终点,还有一些时间刷新点，必须在k步以内到达终点或者是时间刷新点，不然就会死。问你是否能够出去以及出去的最短时间。
$n,m \leq 600$。

分析

因为我们每一次走的目的都是到达时间刷新点或者终点，而中间的状态是不需要去管的，所以我们可以直接预处理处起点终点以及时间刷新点之间的距离，然后将距离小于k的连边，这样就可以构成一个新图，然后直接跑最短路。因为对于每一个时间刷新点来说，只可能走一次，走第二次与第一次到达所能产生的效果是一样的。
然而因为数据的问题，cin会挂。。。另外是小于k的连边，等于也不行。。。

思考

将这种网格状的图变成一般图可以从某种意义上缩小数据的范围，从而采用一些图论的算法高效求解。

F. 表达式

题意

给你一个x^n的表达式，问你最少几步可以使用乘和除吧x达到要求，你可以利用中间过程中的任意一个已经得到的变量。

分析

一起第一次做的时候使用的贪心，发现**90%**的答案都是对的，但是有一些是不对的，因为如果可能在某一步选一个较小一点的而是的直接达到最后结果，而不是选最大的然后超过再除。
接着考虑搜索，$IDA*$虽然限制了深度，但是如果剪枝不到位还是不行，因为我们是考虑一步一步来的，所以我们只能利用之前的结果来剪枝，这里就可以用贪心来剪枝，具体是这样的

```
1.     int maxv = *max_element(ans, ans + cnt); \\ 之前能得到的最大值
2.     if (cur << (maxd - dep) < n) return false;
3.     if (maxv << (maxd - dep) < n) return false;
4.     if ((maxv + cur) << (maxd - dep - 1) < n) return false;
```

也就是如果我们都以为用某一个尽量大的值，看是否能够达到目标，连这样都打不到的话，就不用继续搜了。

思考

贪心往往是剪枝的好办法，所以要仔细利用题目和搜索中之前的信息来贪心剪枝。

G. 神舟的宝藏

题意

给你一个数，让你找到由指定数字组成的正整数倍，要求长度小于500且值最小。$N \leq 5000$。

分析

这个题就和C其实是一个意思，算是加强版，我们只需要和C一样的用数组保存答案， 并且用余数是否出现过来剪枝就好了。

思考

同C题。

H. DNA序列

题意

给你n个长度不超过5的只有$AGCT$构成的字符串，让你找到一个最短的字符串是的所给定的字符串都是它的子序列。$n \leq 4$。

分析

```
int h() {
    int ret = 0;
    for (int i = 1; i <= n; i++) {
        ret = max(ret, len[i] - top[i]);// top[i]表示第i个子串的完成度
    }
    return ret;
}
if (cnt + h() > maxd) return false;
```

思考

想这一类要求最有方案的题目，$IDA\star$ 是一个很好用的方法。

I.小冰和小娜

题意

一个独轮车在一个$m*n$de 地图上走，没走一次独轮车向下的颜色会改变，问你最少要走多少部才能到达终点且绿色朝下（不管朝向）。$m,n \leq 50$

分析

这题虽然也是二维网格图的模型，但是和一般的网格图不同的是，走到相邻的格子的点的花费都是不一样的，也就是说每个点之间都是有权值的，这样的话，bfs就不能盲目的去拓展，必须是启发式的。
这题非常容易的错在两个地方，一个是走的时候颜色的变化和方向之间的关系，另外就是回头的话费是3，一定要注意。
知道这些就可以了，每次往外走拓展就随意搜了。

思考

对于bfs的方式要针对不同的题目会有细节的不同，要注意。

J. TooEasy Or TooDifficult

题意

给定一个长度为k的字符串$3 \leq k \leq 10^5$，对每一个字符定义一个战斗力$P[i]$表示以这个点为中心的回文串的长度，对于其中战斗力最大的字符定义为MZ，定义为

$$JD = (MZ^{k^3})\%(MZ/3*5+1)+MZ*4/5$$

再定义

$$FJD = max(P[i]xorP[i+1]xor...p[j]|(0<i\leq j\leq k)$$

当$JD > FJD$，输出$Tooeasy$，否则输出$TooDifficult$。

分析

其实就是两个模板合起来，一个是$manacher$，还有一个是$XorTrie$求最大异或子区间，如果都已模板的话直接套进去就好了。
一开始不听的超时，后来才发现原来是$manacher$的一个变量没有初始化。

思考

为啥$XorTrie$可以求最大异或子区间和，其实和第一题是一样的，异或的性质表示他可以求前缀和，那么在每产生一个新的前缀，把之前的前缀插入之后，就可以找和这个前缀异或最大的值，原因同第一题。而且这样也保证了可以得到所有的子区间。

K.奶牛合影

题意

最小表示法裸题，听说sa可做，之后再补。

分析

同上

思考

同上

M. 奶牛硬盘

题意

给你硬盘的容量，问你标准容量和实际容量的差值占标准容量的百分比。

分析

水题，可以直接找规律，然后我直接用$JAVA$的$BigDecimal$暴力算了一下。
$JAVA$大法好。

思考

我们可以知道1KB的丢失比例，那么可以递推1MB的，这样打个表就好。