A.我要学三角形

这是一道找规律的题目

1. 当 $n <= 8 \ or \ n = 10$ 时，结果为0；
2. 其他情况下若 n 为奇数，则对应三条边为

$$2, \frac{n-2}{2}, \frac{n-2}{2}+1 \tag{1}$$

3. 其他情况若 n 为偶数，则对应三条边为

$$3, \frac{n-3}{2}, \frac{n-3}{2}+1 \tag{2}$$

B.拯救DAG王国

由于这是DAG,我们考虑拓扑排序，拓扑排序中有个性质：

- 在拓扑排序中任意时刻在队列里的点都不能互相到达。

我们可以利用这个性质，求出每个点能到达的节点个数了。

但是**双一流A城市**和**双一流B城市**的定义是**互相到达**，所以我们需要两次拓扑排序，我们可以将原图的每条边反过来存一次图，称作反图。

把反图正图分别做一遍拓扑排序，再把两次拓扑排序求出来的每个点可以到达的节点个数加起来，就得到了每个点可以和其它点互相到达的节点个数。

如果在拓扑排序的过程中：

`q.size() == 0`，说明此时u能到达剩下的所有点（设为tot），则`d[u] += tot;`

`q.size()== 1`，说明此时u和队列中的点v曾同时在队列里,对于v,如果v存在$v->z$且z的入度为1,那么u显然不能到z,标记一下u即可。

最后枚举每个点,判断是否合法即可。

C.和生蚝一起做算术吧

我们需要从二进制角度考虑这道题。

为了方便起见我们可以把对x的$\times 2$操作等价于对y的$\div 2$，其代价为$

我们可以发现从x到y的变化可以分为三个阶段。

第一阶段：对x连续$\div 2$

第二阶段：对y连续$\div 2$

第三阶段：对x连续$x+1$知道等于y

如果x需要$\div 2$，那么一定出现在对其$+1$和$*2$之前，否则付出的代价会更大，所以第一阶段操作确定。

如果对$x\div 2$和$y\div 2$次数固定后，那么对$x+1$一定出现在最后阶段以减少$+1$操作的数量，所以第二第三阶段操作确定。

在确定下三个阶段后，朴素的想法是枚举前两个阶段操作的数量，再直接计算第三个阶段操作的数量，复杂度为$TO(logn*logn)$但这个复杂度无法通过此题。

实际上我们可以递归的维护前两种操作的数量。

对于当前的x和y

- $x>y$

  执行第一阶段继续递归

- $x<y$但二进制下位数相同

  执行第二阶段继续递归或者直接第三阶段，取最小值

- $x<y$且二进制下位数$lenx<leny$

  执行第一阶段继续递归或者直接第三阶段，取最小值

- $x=y$

  递归结束

注意爆longlong的情况，可以预先判断最大值，避免计算爆longlong的值。

复杂度为$TO(logn)$

D|E. 野兽追猎者塔维什/野兽追猎者塔维什plus

设出现米莎的事件为X，出现雷欧克的事件为Y，出现霍弗的事件为Z，雷欧克的贡献函数为f(Y)。

由于是独立事件，

$$E(\text{血量总和})=E(2X+4Y+4Z)=2E(X)+4E(Y)+4E(Z)=2n/3+4n/3+4n/3=10n/3$$

$$E(\text{攻击力总和})=E(4X+2Y+4Z+Y*(E(X)+E(Y)+E(Z)-1)*f(Y))=10n/3+(n-1)E(Y*f(Y))$$

对于野兽追猎者塔维什：

$$E(\text{攻击力总和})=10n/3+(n-1)n/3$$

对于野兽追猎者塔维什plus：

$$E(\text{攻击力总和})=10n/3+(n-1)\sum_{k=1}^{10}a_{k-1}\sum_{i=0}^{n}C_n^ip^i(1-p)^{n-i}i^k$$

预处理后时间复杂度为$O(10n)$

F.祝泥魔

本质上就是动态字符串集的多模匹配，在ac自动机基础上进行动态增加字符串。如果强行每次增加都重构fail树的话会超时，所以在每次增加字符节点后对节点fail指针指向的与fail指针指向新节点的节点进行更新即可。

加一句，注意在更新节点fail指针时需要递归更新

G.小人国的粉刷匠

一个基础的dp题。

$dp[i][j][k]$表示第i块地板涂j色，前面i块地板被分成k组，所花费的最少体力。

可以用一个数组c来记录每块地板的初始颜色(要求)，初始颜色不为0说明有特殊要求，必须涂成某颜色，为0才可以涂色。

状态转移方程为：

当$c[i] > 0$，说明有要求，第i块地板涂c[i]色，pre为前一块地板颜色。

$if(pre == c[i])$

$dp[i][c[i]][k] = min(dp[i][c[i]][k], dp[i-1][pre][k] + cost[i][c[i]]);$

$else$

$dp[i][c[i]][k] = min(dp[i][c[i]][k], dp[i-1][pre][k-1] + cost[i][c[i]]);$

当$c[i]$等于0，无要求，第i块地板涂$c[i]$色，pre为前一块地板颜色。

$if(pre == j)$

$dp[i][j][k] = min(dp[i][j][k], dp[i-1][pre][k] + cost[i][j]);$

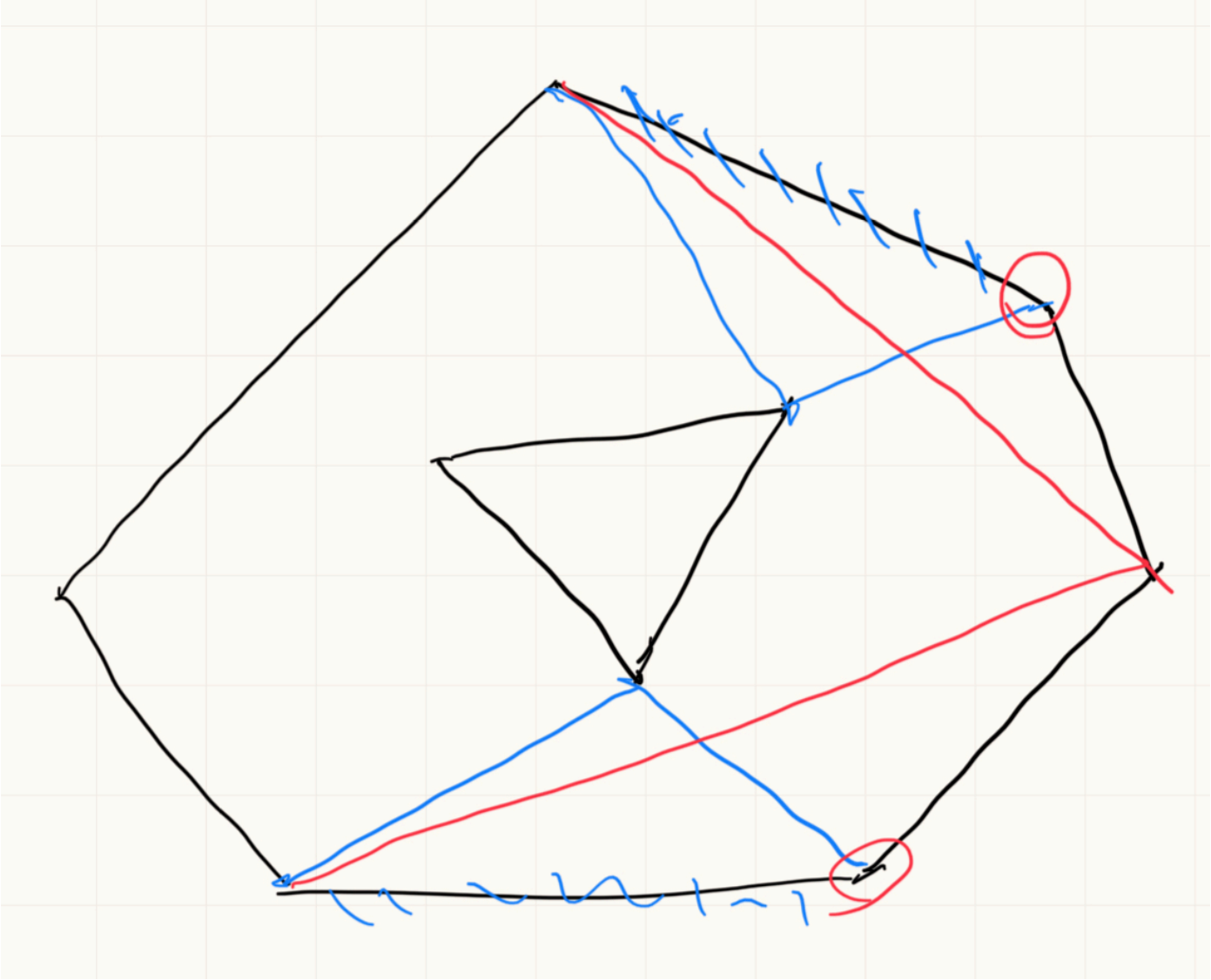
$else$

$dp[i][j][k] = min(dp[i][j][k], dp[i-1][pre][k-1] + cost[i][j]);$

时间复杂度：$O(n^4)$

H.仁慈的富蚝

最大面积为一个凸包，我们考虑在这个凸包（称为外凸包）上删除一个点或者一条边，使得损失的面积最少。



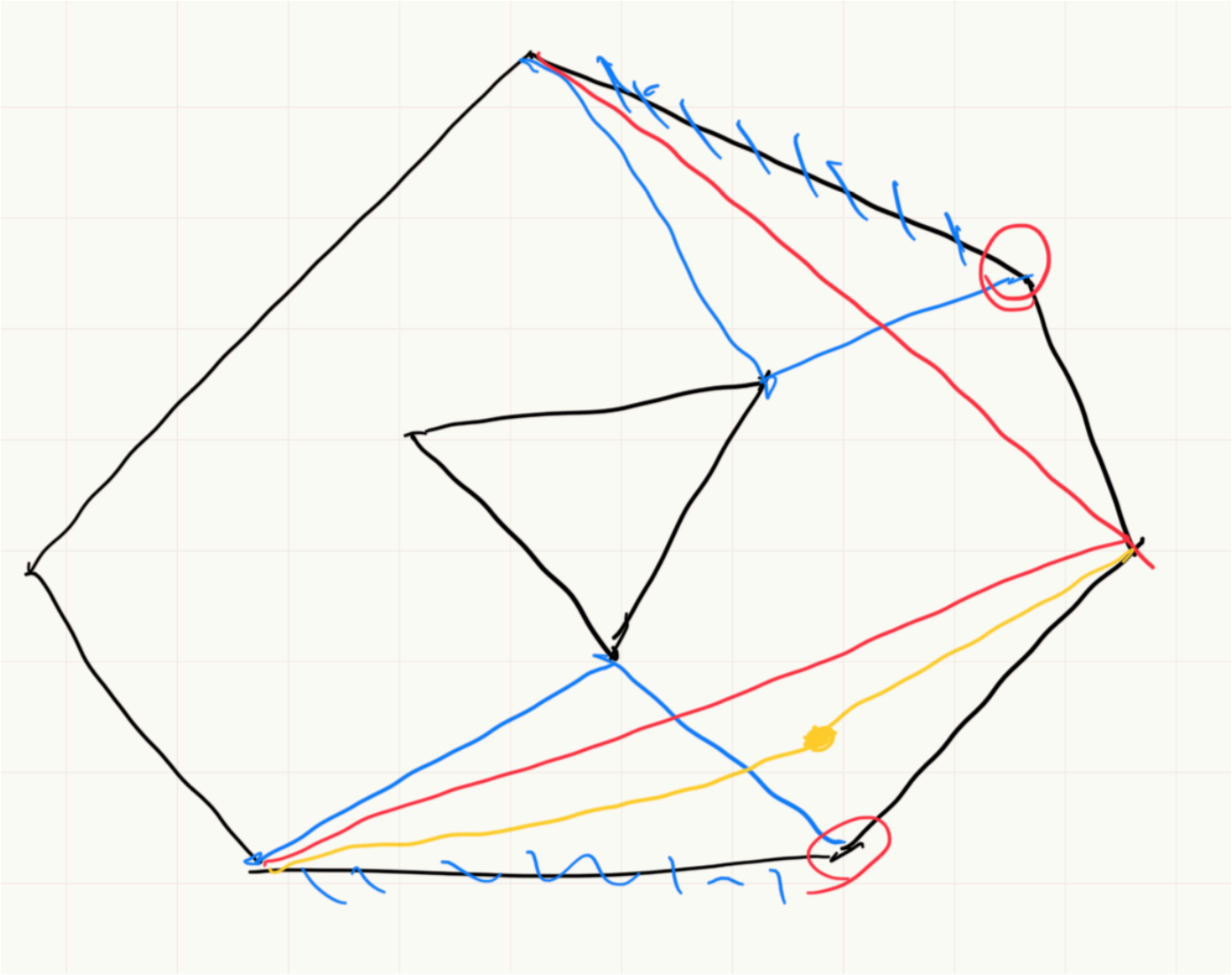
红色的为删除点操作：

我们选择连接外凸包上删除点相邻的两个点。

蓝色为删除边的操作：

我们需要选择非外凸包上的一个点，对于这些点我们只需要在维护一个凸包（称为内凸包），枚举外凸包的边类似旋转卡壳维护内凸包上选择的点即可。

删除点操作也有可能会出现非外凸包上的点更优比如下图黄点，但这个点显然由删除边操作选择更优秀。



时间复杂度为$O(nlogn)$

I. 立体绘图方块

对于每一列单独考虑。

每次对于相应的标志，我们进行暴力枚举，结合已有的标记，更新标记哪些方块是一定是某种颜色以及哪些方块一定不是某种颜色。如果某个方块两种颜色都为不可能那么其为空。

如果有方块更新了标记，则把该方块对应的三个维度上的列放入队列中等待再次暴力枚举。

极限时间复杂度为$O(n^4 2^n)$

J.神奇的魔法

题意是给两个字符串S和R，现在可以打乱R的字符变成字符串T，要求T包含字符串S所有的子串，输出满足条件且字典序最小的字符串T。

要知道包含字符串S所有子串的最短字符串是它本身。因此，经过变换的答案字符串T必须要把字符串S作为子串，然后字符串R剩余的其他字符，可以按照字典序从小到大排序，通过比较字符和字符串S的大小关系，来判断是加在S之前还是S之后，即可得到最终的答案。

注意，在比较和字符串S的大小关系时，在处理S[0]这个字符时，不能直接加到字符串S之前，要判断和S后续字符的关系。

K. 蚝蚝蚝大王的字符串

首先用sam预处理出加入位置i字符时新产生的本质不同子串的左端点区间，称为产生区间。加入位置i字符新产生的endpos即代表了这些新的本质不同子串。

然后我们考虑离线处理所有查询区间，先按查询区间右端点排序。

对于满足查询区间右端点的产生区间，我们区间加1，区间查询处理即可。

事件复杂度$O(nlogn)$

L.生蚝吃蚝蚝

把数组中大于x的变为1，小于y的变为0，做前缀和，对前缀和数组求顺序对，若数组值为0则将答案加上前一个结果，否则加上该位置的顺序对。

M.完美生蚝

看到寻找两个端点之间满足一定要求数字的个数立马可以想到是数位dp。该题需要满足数同时有$[0,k]$所有数的存在，假设使用k+1位参数代表0~k是否存在较为麻烦，本题使用状态压缩，一个数存在与否只有两种状态，将[0,k]数字存在的状态组成的二进制代表当前状态。接下来只要在记忆化搜索时退出条件根据状态压缩的值判断[0,k]是否都存在即可。

N.wqy's easy problem

排完序后维护以某个位置为终结位置的连续上升序列长度，计算后添加到答案中，复杂度$O(nlogn)$。注意数据范围longlong。