SOLUTION

爱哭

`Ai Ku`

这题就不用多说了吧。

直接把板子放过来不用改就A了。

唯一需要注意的是数据范围。

不要以为堆优化`dijkstra`一定比朴素`dij`快就行了。

一般我们分析`dij`都是$O(m\log m)$（`c++ stl`堆）或者$O(m\log n)$（线段树优化或其他玄学优化）。

本机测`stl`堆优和线段树优化都会T，不知道有没有老哥卡过……

`spfa`？他已经死了。

~~话说有人用`spfa`卡过的吗？~~

爱看

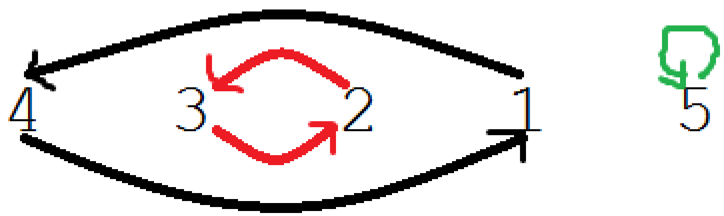
`Ai Kan`

先看第一问。

我们用样例2来解释：

```
4 3 2 1 5
```

我们画一下某个数应该去的位置：



我们发现是一个个的环。

于是我们就可以想到每个排列都可以用几个循环的“乘积”。

比如：

$$\begin{pmatrix}4 & 3 & 2 & 1 & 5\end{pmatrix} = \begin{bmatrix}1 & 4\end{bmatrix} \times \begin{bmatrix}2 & 3\end{bmatrix} \times \begin{bmatrix}5\end{bmatrix}$$

不难发现每个循环是独立的，我们可以分开来算。

对于某个循环，比如$a \to b \to c \to a$，我们选择使用冒泡，交换$n-1$次。

所以假设我们可以分解为k个循环乘积，而长度为n，那么答案就是$n-k$。

实际上策略就是这样的：

```
for(int i = 1; i <= n; ++i)
    while(i != a[i])
        swap(a[i], a[a[i]]);
```

接下来是第二个问题。

又是白送的10个点。

如果你想到了第一问的具体推到过程，那这一问就很好想了：

令$f_{i,j}$表示长度为i，答案为j的方案数。

我们考虑从$i-1$转移到i，考虑第i个数放在何处。

首先我们考虑直接插到循环里，那么就有每个长度为l的循环都有l种放法。

而我们也可以把第i个放在单独的循环里。

所以$f_{i,j} = f_{i-1,j} \times (i-1) + f_{i-1,j-1}$

爱困

`Ai Kun`

~~想不到把这题是个大结论题。~~

前6个点是白送的。

第$7\sim14$个点考虑`dp`，令f_i表示$n=i$时的答案。我们有$f_i=\max(f_{i-j}\times j)$。

我们不难想到分成的数应该不会太大，我们枚举j的时候之枚举枚举$1\sim k$，k取个小一点的值，比如说10。

80分到手。

我们考虑$n \le 10^{18}$的情况。

显然是结论嘛……

那我们就先打张表（手打都可以）：

```
1 = 1
2 = 2
3 = 3
4 = 4
5 = 3 + 2
6 = 3 + 3
7 = 3 + 4
8 = 3 + 3 + 2
9 = 3 + 3 + 3
...
```

打多一点，仔细观察之后就会发现如果给出一个n，我们应尽量把n分成$3+3+3+\cdots+3$，如果最后一个剩下来的是1，那么就把其中一个3和1合并变成4。

然后快速幂大力跑一波就行了。

如果是想理性证明的话，首先我们发现肯定是要均匀的（均值不等式），所以这个函数大概就是$x^{\frac{n}{x}}$。

如果你是熟练的选手，发现取对数一波后这个函数是可以直接导的，AK。

如果你是个不怎么熟练的选手，直接暴力枚举所有正整数发现在3的时候最大，发财。

爱卡

`Ai Ka`

这题需要用到线段树。

但据我统计了一下截止2月28日（出这题的时间）已有14人A了【模板】线段树1。

所以我放心地出了。

开局送你70分，只要认真订正zd的模拟赛就行了。

和zd的那题没有任何区别（我才不会告诉你我部分数据也是用了他的）。

如果你A了线段树模板1你还拿不到80分那就真不应该了......

关键是最后20分。

我们发现无法维护。

既然我们无法从数据结构下手，我们从gcd方向考虑。

想想gcd满足什么性质。

有没有想到$\gcd(a,b)=\gcd(b,a-b)$？

差分？？？

比如说我们要求

$$f(l,r)=\gcd(a_l,a_{l+1},a_{l+2},\cdots,a_r)$$

其实我们就是要两两求gcd：

$$\gcd(\gcd(a_l,a_{l+1}),\gcd(a_{l+1},a_{l+2}),\cdots,\gcd(a_{r-1},a_r))$$

然后我们就利用刚才那个性质：

$$\gcd(\gcd(a_l,a_{l+1}-a_l),\cdots,\gcd(a_{r-1},a_r-a_{r-1}))$$

然后展开来：

$$\gcd(a_l, a_{l+1} - a_l, a_{l+1}, \cdots, a_{r-1}, a_r - a_{r-1})$$

也就是

$$\gcd(a_{l+1} - a_l, a_{l+2} - a_{l+1}, \cdots, a_r - a_{r-1}, f(l, r-1))$$

消去了一个r？

我们接着消。

由于

$$f(l,r-1)=\gcd(a_{l+1}-a_l,\cdots,a_{r-1}-a_{r-2},f(l,r-2))$$

而这一段$\gcd(a_{l+1}-a_l,\cdots,a_{r-1}-a_{r-2})$对于前面$f(l,r)$的式子是没有意义的。

于是我们就可以发现，对于任意$k<r$

$$\gcd(a_{l+1}-a_l,a_{l+2}-a_{l+1},\cdots,a_r-a_{r-1},f(l,r-k))$$

所以我么你要求的其实就是

$$\gcd(a_l, a_{l+1}-a_l, a_{l+2}-a_{l+1}, \cdots, a_r-a_{r-1})$$

我们用一棵线段树维护差分后的gcd。由于差分数组可以将区间加转化为单点加。所以这个维护就是80分的那一档。

我们再开一棵线段树（或者树状数组）维护区间加单点查询，这样就能维护处a_l了。

小结

`Small junction`

~~一般吧“小”和“结”扔到百度翻译里就能得到正确翻译~~

这次模拟赛并不是很难。

为了让大家都拿到分，故意把暴力分设得很大。

显然`T1`是用来搞笑的，出题的目的就是为了提醒一下大家：**完全图千万别用堆优！！！**

曾见到某神仙选手完全图打堆优`prim`然后炸成暴力分所以灵感大发出了这道题......

赛前预测这道题得分率最高。

赛前预测得分率第二高的应该是`T4`。

由于`GGF`~~今~~去年出原题，出题人就也就~~搬~~出了道原题~~的弱化版~~。前50%的数据和上次那场完全相同，后50%是自己造的。

建议前60没拿到的同学先把过去几场模拟赛的至少A,B题认真订正一下。

预计是会有14为同学拿到80分，因为我统计了一下，截至出题时间应该是有14为同学过了线段树模板1，会打线段树的同学应该是可以很快拿到20分的。

最后$\mathbf{20}$分是给神仙选手准备的，感觉确实挺难。

如果有人现场A了我只能伏地魔%%%%%%%%

`T2` 和 `T3` 感觉难度应该差不多。

`T3` 大概学习的是17年的$D1T1$。

这种题目如果不能一眼秒建议直接打表。

当然如果你足够神你是可以一眼秒的，比如 ~~（为保护他人的肖像权和姓名权，已打码）~~ ：

`T2` 学习的是去年 `D1T3` 的风格，乍一眼看好难（可能是出题人太菜了），仔细想想之后发现也就这样。

考前的估分

最单纯的暴力分：$100 + 50 + 30 + 70 = 250$

发挥较好的选手：$100 + 80 + 80 + 80 = 340$

高水平选手的底线：$100 + 80 + 100 + 80 = 360$

最后猜一波平均分：$80 + 40 + 70 + 75 - 10 = 255$

考虑到$T2$暴力比较难打，$T2$的平均分可能会比暴力分少。

最后的减10是由于失误引起的失分。