T1

回忆gcd可以差分求得（暑期我们讲过的），所以将区间$+k$的部分进行差分，假设将第l个到第r个同学$+k$，那么我们的gcd可以表示为：

$$\begin{gathered}\gcd(\gcd(a_1,\ldots,a_{l-1}),\\ \gcd(a_l+k,a_{l+1}-a_l,\ldots,a_r-a_{r-1}),\\ \gcd(a_{r+1},\ldots,a_n))\end{gathered}$$

我们注意到前缀的gcd是越来越小的，而且$\gcd(a_1,\ldots,a_l)$是$\gcd(a_1,\ldots,a_{l+1})$的倍数。所以前缀$\gcd$只有$\log a$种。同理，后缀gcd也只有$\log a$种，于是第三项gcd的取值只有$\log a$段，每一段取值均相同。

当固定一个l的时候，对于第三项取值相同的每一段，我们发现r越小越好，越小的话第二项越大，于是我们对每一段取r最小的那种情况。

我们枚举每一段r最小的那个位置，然后枚举l即可，用双指针实现。

时间复杂度$O(n\log a)$

T2

将轮数作为第一关键字，当前轮走的距离作为第二关键字，定义成最短路径的距离使用dijkstra算法。

假设一个结点的最短路径距离是(r,d)，考虑dijkstra更新它的一个后继结点，如果这条边和r轮颜色a_r相同，且$d+l$小于等于这一轮的路径总长b_r，那么后继结点更新到$(r,d+l)$，否则在r轮之后寻找颜色和这条边的颜色c相同且总长大于等于l的一轮r'，更新到(r',l)。

后者可以用很多种做法解决。比如对每种颜色建立动态开点的线段树后在线段树上二分，也可以对每种颜色建立ST表，在ST表上二分。

时间复杂度$O(m\log m+m\log k+k\log k)$

T3

你发现a_i很小，于是公差也不会超过100，所以我们从-100到100枚举公差。

用$f[i][j]$表示以a_i结尾，公差为j的等差数列数量。那么有

$$f[i][j]=1+\sum_{a_i-a_x=j}f[x][j]$$

用一个桶来存$f[x][j]$对应的和，时间复杂度$O(an)$。

T4

注意到除了没经过的最多20条边之外，其余边恰好经过了一次，这意味着除了受这20条边影响的最多20个点外，其他结点出度恰好为1。我们将这样结点连的边标记成关键边。这些点的下一个结点是确定的。

那么没有被关键边标记的点我们暴搜即可，时间复杂度为$O(2^{m-n}(m-n))$