Solution

Problem 1

Task1 对于50%的数据

考虑一个长度为i的序列的最后一个人，如果加入这个人，这个人要多久才能打到饭。

由此定义$f[i][0/1]$表示最后一个人是$0/1$的情况下，最后一个人打到饭的时间之和，0表示最后一个人是男生，1是女生。

定义$g[i][0/1]$表示最后一个人是$0/1$的情况下的总方案数。

$f[i][0] = f[i-a][0] + f[i-b][1] + d * g[i][0]$
$f[i][1] = f[i-c][1] + f[i-b][0] + e * g[i][1]$
$g[i][0] = g[i-a][0] + g[i-b][1]$
$g[i][1] = g[i-c][1] + g[i-b][0]$

记得取模，复杂度$O(L)$

Task2 对于100%的数据

容易发现，L非常大，但是a, b, c非常小。

因此每次转移时所需要的$i-a, i-b, i-c$非常靠近i，因此可以考虑使用滚动数组转移。

但滚动数组并没有对时间上做出优化。

可以用矩阵乘法来代替滚动数组的转移，构造一个$4 * max(a, b, c)$阶的转移矩阵即可。

复杂度$O((max(a, b, c) * 4)^3 log(L))$

Problem2

Task1 30%

直接将每个人的名字和笔名求最长公共前缀，然后进行二分图最大权匹配,

Task2 100%

我们可以直接 利用trie树来贪心，从叶子到根，如果当前这个节点能作为 若干对点的 lca，那么就让这若干对点匹配，复杂度为 字符总个数 级别

Problem 3

对于 10%只有操作 2

直接不输出即可。（没拿到 10 分的小朋友可得自我批评了）

对于另外 20%的数据，保证 r - l + 1 <= 7

首先我们注意到每次操作一个区间最多 7 个元素 我们对于操作 1 可以用搜索来枚举是否最终总价值会变为 0。对于每个数的系数，有三种可 能 1,0，-1。1 表示选入上午，0 表示不参加，-1 表示选入下午。如果最终价值变为 0 即可提 前结束搜索。每次最坏复杂度 3^7。对于操作 2，暴力修改即可。

对于另外 30%的数据，保证只有操作 1

设$r-l+1=len$。则子集合方案数为2^{len}，区间值域为$[len，len \times v]$.只要使 $2^{len}>len \times v$，就保证操作 1 一定会输出 Yes。 $2^{len}>len \times 1000$解得 len 的最小正整数解为 14。 所以当 $len \geq 14$，操作 1 可以直接输出 Yes。 当 $len<14$。首先会想到像上一个分段一样搜索，但每次搜索复杂度最大为$3^1 3$。 考虑采用二分优化，首先搜索完[l，mid]内所有集合的可能值域，再去搜索[mid + 1，r]一旦 发现得到的某个值在之前出现过，即可直接结束搜索。当然如果两个搜索区间内搜索到值为 0 也可提前结束搜索。 每次搜索复杂度最坏为3^6+3^7

对于 100%的数据

对于操作 2，考虑用线段树 lazy-tag 实现区间修改。 区间幂不好区间修改，但考虑到每次最多调用 13 个数，可以下放到叶子节点时才释放 lazy 标记。 lazy 标记释放时，快速乘或暴力修改常数是巨大的。 发现 b[i]值域在[0，v - 1]，可以提前处 理好[0，v - 1]的幂的表格，用倍增实现。 $data[i][0]=i^3$，$data[i][j]=[data[i][j-1]][j-1]$。