Solution

Problem A. Frog

- 首先我们知道两人相遇有**相向相遇**和**追及相遇**两种，分别满足方程 $(V_1+V_2)t=(2k+1)L$，$(V_1-V_2)t=(2k+1)L$。
- 那么我们可以对于两种情况分别解出k的个数
- 但是如果在端点相遇时会同时满足两个方程，我们只能计算一次，需要将其减去。
- 我们用可以算出最短周期T的使两人在端点相遇并将其减去即可
- 设T_1,T_2分别为两人经过端点的周期，则$T=lcm(T_1,T_2)$，分数求lcm可以先通分，然后分子取lcm
- 若T时刻两人没有在端点相遇，不难发现两人一定不会在端点相遇，就不用减去任何东西

Problem B. Sort

- 首先将A数组用一次操作排一遍序
- 然后考虑从左往右逐一将A中的元素调整为B中的元素，并将已经调整至相等的公共前缀删去。
- 如果需要调整到开头的元素是序列的前缀最大值或前缀最小值，那么我们就可以通过一次操作达到目标
- 那么一开始排好序后满足所有元素都是前缀最大值或前缀最小值，当我们对序列前缀排序操作后，不难发现仍满足所有元素都是序列的前缀最大值或前缀最小值。
- 所以我们总可以进行操作直到两序列相同。
- 总操作次数$1+n-1=n$

Problem C. Battle

对于一轮而言，$\Theta(n)$ 模拟是可行的。具体来说，我们用一个变量记一下目前存活的 $type_i\&2=2$ 的人数，然后对于$type_i\&4=4$ 的增益效果，在每一轮模拟之后做一个后缀和就可以了。

不难发现 $type_i$ 为奇数的敌人最多只能存活 $\log life_i$ 轮，所以考虑模拟这些轮，直到只剩下 $type_i$ 为偶数的敌人。

现在我们就可以算出所有存活的人的存活轮数了。这里记住要先特判 $Atk=0$ 的情况。

算出每个人的存活轮数，然后在每个敌人死亡的那一轮来统计贡献， std 实现过程中使用了树状数组。

记最大的生命值为 m .每组数据时间复杂度 $\Theta(n\log m+n\log n+m)$. 实际上复杂度里可以没有 $\Theta(m)$ ，但是这样写起来会更麻烦。

Problem D. Match

考虑一个贪心：

我们在每个点上记两个堆：黑堆和白堆。 dfs 子树之后将子树的堆合并到自己的堆上，同时打一个整体权值减少的标记，然后考虑在当前点匹配。

但是这个贪心是不对的。

假如我们在子树 x 内部配对了一对权值为 0 的黑球和白球，然后子树外面有一个权值为 10^{12} 的白球，除此之外没有别的黑球和白球。

此时需要将这个白球和 x 内部的黑球匹配才能获得最优解。

因此我们需要让这个贪心支持反悔。

具体来说，我们在一个点匹配的过程是取出黑堆和白堆中的最大值并将他们匹配之后，将这两个权值乘以 -1 , 然后加进对方颜色的堆里。 不难发现这样就实现了反悔贪心。

设 m_a, m_b 和 n 同阶。时间复杂度 $\Theta(n \log n)$