A

结论：每次使用普通攻击击杀血量最小的敌人，然后用再现攻击打血量最大的敌人。

下面是一些比较感性的证明。

注意到，所用次数最小，当且仅当再现攻击打出的伤害最多，因此，再现攻击打 ≥ 3 的敌人优于打 2 的优于打 1 的 ，且答案不小于 $\sum a_i - (2n-2)$。

首先发现，当血量最小的敌人血量 ≥ 3 的时候，答案就是 $\sum a_i - (2n-2)$ ，因为 $n-1$ 次再现都能打满伤害。因此，这些敌人先用普通攻击打哪个并不重要，再现打哪个也不重要，答案一定不会改变。

而当血量最小的敌人血量 $=1$ 的时候，你必然会优先用普通攻击攻击他，因为再现攻击打 1 是最劣的情况，严格劣于打 ≥ 2 的，所以 1 的优先级高于 2 。

而当血量最小的敌人血量 $=2$ 的时候，你也必然会优先用普通攻击攻击他，原因和 $=1$ 类似不再赘述。

由此可以得出结论前半部分的正确性。

当血量最大的敌人血量 ≤ 2 的时候，结论后半部分也显然正确。

考虑剩下的情况。根据前面的结论，我们只需要考虑最小值 ≤ 2 的情况。此时，结论中描述的操作显然是局部最优解，因为没有产生额外的浪费，重复执行这样的操作，一定能够转化为前面的某一种情况，从而达到全局最优解。

由此可以得出结论后半部分的正确性，结论得到了证明。由实现好坏可以得到 $O(n)$ 或 $O(n\log n)$ 做法。

B

首先，最大值一定不会改变。

发现答案可二分，考虑二分答案，问题转化为能否判定将所有 $< y$ 的数都变为 $\geq y$ 。显然的，所有 $< y$ 的数都必须被加法操作包含。且加法操作所加的值就是 $y - \min A$ 。但是这不一定合法，因为这可能会让区间里的一些数超过了最大值，导致了最大值改变，只需要判掉这种情况即可。

时间复杂度 $O(n\log V)$ ，当然二分是不必要的，有线性做法，但是二分跑得很快且这个题是签到，就都放过去了。

Fun fact:这个题本来撞了 P1463，但是被大神一眼看穿了，所以就换了一个题。

C

首先考虑有根树（也就是根必须最后删除）怎么做。

记 f_i 表示令 i 为根，删空整个子树的方案数。显然的，根一定在整个删除序列的末尾，而不同子树的删除之间互不影响，两个序列可以直接合并，序列合并的方案树为 $\binom{sz_i}{sz_i+sz_j}$ ，将所有子树直接合并起来即可。

对于无根树，一个比较暴力的想法是，我们钦定哪一个点最后删除，然后将所有方案加起来，复杂度是 $O(n^2)$ 的。

又可以注意到，我们并不需要对每个根做一遍 dp，可以直接换根，这样时间复杂度就变成 $O(n)$ 的了。出题人开摆了所以标程是 $O(n\log p)$ 的。

D

我们第一时间会想到一个 $O(n^2m)$ 的暴力算法，也就是直接模拟题意。

然后我们会想到一个 $O(nm^2)$ 的拆位做法，这个是典中典。

记 $V=nm$ ，直接根号分治可以做到 $O(V\sqrt{V})$ 。

注意到 $O(n^2m)$ 的做法，我们并不需要将所有的位单独拿出来算异或，而是可以将连续的 64 位压在一起算，总复杂度可以除一个 $\sqrt{w}$ ，可以多拿点分，但应该不能过。

考虑如何对 $O(nm^2)$ 的做法进行压位。设我们枚举的两位是 j 和 k ，我们事实上只关心这两位为 $01,10,11,00$ 四种情况的出现次数。其中 11 和 00 可以直接异或起来后数 pop count 。而 01 和 10 则可以将一位全部反转，转化为前面的情况。这个显然可以压位解决，总复杂度又可以再除一个 $\sqrt{w}$ 。

这个时候，我们的总复杂度应该就是 $O(\frac{V\sqrt{V}}{w})$ ，可以通过了。实现的时候，考虑到 $O(nm^2)$ 的压位常数较大，可以适当调整根号分治的阈值，当然时限放得很大，写的不好应该也能过。