A

注意到可以直接对$O(C^2)$个状态建图跑最短路，因为边权都是1，最短路就是BFS，期望得分60。

注意到本质有效的状态只有$O(C)$个，直接跑BFS即可。

B

背包合并复杂度是M^2的，不可接受。但如果只合并两个背包，求单点值是可以做到$O(w)$的。

建立猫树，直接做空间接受不了，因此把询问离线到每一层做即可。

C

考虑序列 b 的最终状态，其中的每个数一定都是初始时 b 中所有数的 $\gcd$ 。

每次合并相邻的两个数，因此 b 的最终状态的每个数都对应了初始序列的一个区间的 $\gcd$ ，要使得操作次数最少，那么就是让分割出的区间最多。

先用ST表计算出 $a[l..r]$ 的 $\gcd$ ，设其为 g 。

考虑贪心，从左端点 l 开始，跳到最小的 i ，使得 $\gcd(a[l..i]) = g$ ，然后令 $l = i + 1$ 并重复该过程直到碰到右端点 r 。最后可能有一段无法得到 $\gcd$ 为 g 的一段，将其并入上一段即可。

预处理 $nxt_{i,g}$ ，也就是从 i 开始跳到的最近点，使得该区间的 $\gcd$ 为 g 。以 i 为左端点， $i..n$ 为右端点，不同 $\gcd$ 的数量是 $O(\log A)$ 的，因此只有 $O(n \log A)$ 个需要计算的 nxt 。

要加速从 l 跳到 r 的过程，使用倍增即可。

复杂度为 $O(n \log n \log A + q \log n)$ 。

D

注意到能被阳光照射的方格只有$6N^2$个，因此模拟题意建图之后直接BFS即可，本题主要考察代码细节的实现能力。

E

注意到本质不同的，由六种不同颜色各两个构成的染色方案只有不到2×10^4种，对于每一种，确定染色方案之后从回文串中间往两侧动态规划复杂度是$O(2^{n/2} \times n)$的，因为只需要记录出现的颜色是哪些。

最后对于每种本质不同的染色方案，只有不到2×10^5种，可以把他们建立一个DAG，只用计算上述的每一种染色情况，然后在DAG上推一遍就可以得到所有的情况了。

F

首先令$x_i = \sum_{j=1}^{i-1}[P_i < P_j]$。 x与P双射，于是就可以通过x刻画好位置了。 那么就是每次冒泡排序后统计x中> 0 的数的连续段个数。 每次冒泡排序后x中大于0 的元素会减一然后整体前移。

可以发现序列x的贡献为$\sum \max(x_i - x_{i-1}, 0)$ ，这个东西可以拆开然后算贡献，但复杂度比较大。

进一步观察发现 $\sum \max(x_i - x_{i-1}, 0)$等价与$\sum v_i$ ， v_i表示第i个位置后面比P_i小的那些数对应的位置的连续段个数。 这个就比较好算，枚举一个位置i ，一个位置$j(i < j)$当且仅当$P_i > P_j$且$P_{j+1} > P_i$ 。 算的时候枚举填啥，然后其他数随便填，得到一个$O(n^3)$的算法。 预处理组合数，结合前缀和优化可以做到$O(n^2)$。

进一步展开组合数发现可以从前往后用BIT或者线段树维护组合系数的变化，总复杂度$O(n \log n)$