

A

在从根向下 dfs 的过程中，搜索到叶子的时候所访问到的节点就是从根到叶子路径上的所有节点。

对于前 k 大的限制条件，使用 set 维护路径上前 k 大数，每次搜到一个节点将该节点的值加入集合。若集合内数字数量大于 k ，则删除最小的，并用另外一个集合维护在该节点删除的数，回溯过程中若有需要把另一个集合中最大的那个数加回去即可。

时间复杂度 $O(n \log k)$

B

类似康托展开求排列的过程，求对应字典序的序列时可以从前往后一位位确定。记以 c 开头且长度为 n 的序列数量为 $f(n, c)$ ，则有

$$f(n, c) = f(n-1, c-1) + f(n-1, c) + f(n-1, c+1), \quad c \leq n$$

当 $c > n$ 时，易知 $f(n, c) = f(n, c-1)$

对于首位，我们找到最大的 c 满足 $f(n, c) \leq k$ ，这个可以通过预处理完毕后枚举 $\leq n$ 的 c 或在 $f(n, n) \leq k$ 的情况下直接通过单次除法获取 c 的值。剩下的位依此类推。

C

树上独立集问题，颜色数量很小，因此可以对颜色进行状压。

设 $f(v, C, 0/1)$ 代表以 v 为根的子树，在颜色选择情况为 C 且 v 选/不选时的最大点权和。在对于节点 u 维护 f 时先令 $f(u, 2^{c_u-1}, 1) = a_u$ 和 $f(u, 0, 0) = 0$ ，考虑将 u 的每个子树合并进去：

$$\begin{aligned} f(u, C, 1) &= \max_{T \subseteq C, S=C \setminus T} f(v, T, 0) + f(u, S, 1) \\ f(u, C, 0) &= \max_{T \subseteq C, S=C \setminus T} \max(f(v, T, 0), f(v, T, 1)) + f(u, S, 0) \end{aligned}$$

最终答案为 $\max_S f(1, S, 0/1)$

时间复杂度 $O(n \times 3^C)$

D

每个字母独立，设 $f(u, c)$ 代表节点 u 选择字母 c 时子树内最小的边权之和，则

$$f(u, c) = \sum_v \max_{k=0}^3 (f(v, k) + [k \neq c])$$

E

$|S|$ 很小，且要求出现恰好两次，考虑设 $f_{i,j,0/1/2}$ 代表当前字符串长度为 i ，目前匹配到 S 的第 j 位，总共匹配好 $0/1/2$ 次的方案数，有

$$\begin{cases} f_{i,j,0/1/2} \rightarrow_{+c} f_{i, \text{match}[j][c], 0/1/2}, & \text{match}[j][c] \neq |S| \\ f_{i,j,0/1} \rightarrow_{+c} f_{i+1, \text{pos}, 1/2}, & \text{match}[j][c] = |S| \end{cases}$$

其中 $\text{match}[j][c]$ 代表匹配到第 j 位时插入字母 c 后匹配的位置，pos 代表有一个完整串后匹配的位置。

矩阵快速幂加速转移即可。

F

令 f_i 表示长度为 i 的所有位置都被一个好的区间包含的序列的个数，令 g_i 表示在一个 $1 \sim k$ 的排列后插入 i 个数满足最后 k 个数仍然组成一个 $1 \sim k$ 的排列，**且在插入前 $i - 1$ 个数的任何时候最后都无法形成排列（去重）**，则

$$f_i = \sum_{j=1}^k f_{i-j} \times g_j$$

对于排列而言，数字之间的大小关系可以忽略，故以 $1, 2, 3, \dots, k$ 的排列为例求出 g_j 。首先 $g_j = j!$ ，需要减去不合法的方案数，即 $\exists i \in [1, j-1]$ ，满足 g 中最后一个不合法的前缀，即前 i 个数刚好是 $1 \sim i$ 的排列，这样的方案数是 $i!$ ，后面 $i+1 \sim j$ 的数必须**合法**，方案数为 g_{j-i} ，故有

$$g_j = j! - \sum_{i=1}^{j-1} i! \times g_{j-i}$$

时间复杂度 $O(nk)$ 。

G

考虑每个数在期望中的系数，设 $f_{i,j}$ 代表左边有 i 个数，右边有 j 个数时的期望贡献，则

$$f_{i,j} = \left[\frac{i-1}{i+j} + \frac{1}{2(i+j)} \right] f_{i-1,j} + \left[\frac{j-1}{i+j} + \frac{1}{2(i+j)} \right] f_{i,j-1}$$

$$(i+j)f_{i,j} = (i - \frac{1}{2})f_{i-1,j} + (j - \frac{1}{2})f_{i,j-1}$$

$$(i+j)!f_{i,j} = (i+j-1)![(i - \frac{1}{2})f_{i-1,j} + (j - \frac{1}{2})f_{i,j-1}]$$

令 $g_{i,j} = (i+j)!f_{i,j}$ ，则

$$g_{i,j} = (i - \frac{1}{2})g_{i-1,j} + (j - \frac{1}{2})g_{i,j-1}$$

转化为网格图路径计数问题， i, j 等价，所以每条路径的权重相同，均为

$\prod_{k=1}^i (k - \frac{1}{2}) \prod_{k=1}^j (k - \frac{1}{2})$ ，路径条数为 $\binom{i+j}{i}$ ，预处理后可以 $O(1)$ 计算。