

Solution

dy0607

March 31, 2018

1 Max

Source: CSA Expected Max

注意到 m 比较小，不妨分开考虑每个元素在哪些操作中被加了。

考虑状压DP，先计算 $f(i, S, j)$ 表示 A_i 在进行 S 集合的操作之后，值为 j 的概率。

设 $dp(i, j, S)$ 表示前 i 个元素的最大值为 j ，用掉了 S 集合的操作的概率。转移时枚举当前元素用掉了哪些操作，以及进行这些操作后的值。即：

$$dp(i-1, j, S_1) \times f(i, S_2, k) \rightarrow dp(i, \max(j, k), S_1 + S_2)$$

$$O(n(cm)^2 3^m).$$

2 Paint

Source: AtCoder Regular Contest 063 F : Snuke's Coloring 2

题目实际上是要找一个周长最大的矩形，内部不包含任何关键点。

可以发现一个小性质：答案的下界为 $2 \times (\max(w, h) + 1)$ ，因此这个矩形一定会经过 $x = \frac{w}{2}$ 或 $y = \frac{h}{2}$ 。先考虑经过 $x = \frac{w}{2}$ 的情况，另一种情况是一样的。

先将坐标离散化。枚举矩形的上边界 y_R ，对于每一个下边界 y_L ，我们可以计算出矩形的最优左边界 $x_L = \min\{X_i | Y_i \in [y_L, y_R], X_i > \frac{w}{2}\}$ ，以及右边界 $x_R = \max\{X_i | Y_i \in [y_L, y_R], X_i \leq \frac{w}{2}\}$ ，此时可以找到一个周长为 $2 \times (x_R - x_L + y_R - y_L)$ 的矩形。

直接做是 $O(n^2)$ 的，但该算法可以用线段树优化，在将上边界往上移的过程中动态维护每个位置的 x_L, x_R ，并维护全局最小值，不难发现只需要左右各开一个单调栈，在更新单调栈时在线段树树上进行区间加减即可。 $O(n \log n)$ 。

就算没有观察到上面的小性质，也可以多套一层分治解决，复杂度多一个 \log 。

3 Decompose

3.1 Chain

方便起见先考虑链上 $L = 2$ 的情况，不难想到DP：

$$dp(i, 2) = dp(i+1, 1) + w[i][2]$$

$$dp(i, 1) = \max\{dp(i+1, 1), dp(i+1, 2)\} + w[i][1]$$

我们可以将这个转移写成矩阵，注意这里的矩阵乘法是重新定义的—乘法变为加法，加法变为取 \max 。不难发现此时它仍满足结合律（与Floyd算法类似）：

$$\begin{pmatrix} dp(i+1, 1) & dp(i+1, 2) \end{pmatrix} \times \begin{pmatrix} w[i][1] & w[i][2] \\ w[i][1] & -\infty \end{pmatrix} = \begin{pmatrix} dp(i, 1) & dp(i, 2) \end{pmatrix}$$

$L > 2$ 的情况是类似的，用线段树维护一个区间的转移矩阵的乘积，每次只需要单点修改一个矩阵， $O(q \log n L^3)$ 。

3.2 Tree

考虑树链剖分，重链上的转移与链上的情况是类似的，只是此时还要加上轻儿子信息。设 $f(i) = \max_{j=1}^L dp(i, j)$ ，下面仍以 $L = 2$ 为例。

$$\begin{aligned} dp(i, 2) &= \max_{c \in \text{child}(i)} (dp(c, 1) - f(c)) + \sum_{c \in \text{child}(i)} f(c) + w[i][2] \\ dp(i, 1) &= \sum_{c \in \text{child}(i)} f(c) + w[i][1] \end{aligned}$$

设重儿子为 s ，记轻儿子信息

$$\begin{aligned} \text{sumc}(i) &= \sum_{c \in \text{child}(i), c \neq s} f(c) \\ \text{maxc}(i) &= \max_{c \in \text{child}(i), c \neq s} (dp(c, 1) - f(c)) \end{aligned}$$

可以写出矩阵：

$$\begin{pmatrix} dp(s, 1) & dp(s, 2) \end{pmatrix} \times \begin{pmatrix} w[i][1] + \text{sumc}(i) & w[i][2] + \max\{\text{maxc}(i), 0\} + \text{sumc}(i) \\ w[i][1] + \text{sumc}(i) & w[i][2] + \text{maxc}(i) + \text{sumc}(i) \end{pmatrix} = \begin{pmatrix} dp(i, 1) & dp(i, 2) \end{pmatrix}$$

将轻儿子信息视为常数，我们就可以快速弄出一条重链的转移矩阵，然后对于一条重链的父亲，我们需要更改它的轻儿子信息，其中 $\text{maxc}(i)$ 需要开一个set维护。

$L > 2$ 的情况是类似的， $O(q \log^2 n L^3 + n L^3)$