树形 DP

150137

MSOI

2018 年 6 月 2 日

树的直径

树的直径定义为一个树中最长的一条链。求树的直径有两种做法。

树的直径

树的直径定义为一个树中最长的一条链。求树的直径有两种做法。

一种做法比较显然，我们可以大力 DP，维护出到一个节点时向下的最长链和次长链，然后用二者加和来更新答案，同时更新父亲节点的最长链和次长链。

树的直径

树的直径定义为一个树中最长的一条链。求树的直径有两种做法。
一种做法比较显然，我们可以大力 DP，维护出到一个节点时向下的最长链和次长链，然后用二者加和来更新答案，同时更新父亲节点的最长链和次长链。
另外一种做法则可以这样：随便选择一个点，然后找到距离这个点最远的点 A, 再以 A 为源点，找到距离 A 最远的一个点 B, AB 路径上的点就是直径。

树的直径

对于第一种做法比较适合拓展到其他形式的 DP 上去，更新时维护次小的想法也比较适合推荐。同时可以求出任意一个子树的直径。
第二种方法我们可以拿出一些非常有用的性质。这些性质有利于解题。

树的直径

对于第一种做法比较适合拓展到其他形式的 DP 上去，更新时维护次小的想法也比较适合推荐。同时可以求出任意一个子树的直径。

第二种方法我们可以拿出一些非常有用的性质。这些性质有利于解题。

第二种方法不支持负权，这需要注意下。

求树的重心

对于一棵 n 个节点的无根树，找到一个点 A，使得把树变成以该点为根的有根树时，最大子树的结点树最小。A 叫做重心。

求树的重心

对于一棵 n 个节点的无根树，找到一个点 A，使得把树变成以该点为根的有根树时，最大子树的结点树最小。A 叫做重心。
求法很简单，求 $size$ 即可。

求树的重心

对于一棵 n 个节点的无根树，找到一个点 A，使得把树变成以该点为根的有根树时，最大子树的结点树最小。A 叫做重心。
求法很简单，求 $size$ 即可。
容易发现重心的各个儿子的 $size \leq \frac{n}{2}$

支配集与独立集

求一个最大点集使得其中点的个数尽可能多且该集合中不存在两个点有边相连。

支配集与独立集

求一个最大点集使得其中点的个数尽可能多且该集合中不存在两个点有边相连。
求一个最小点集使得树上的每个点都存在一个集合中的点与其相连。

支配集与独立集

求一个最大点集使得其中点的个数尽可能多且该集合中不存在两个点有边相连。

求一个最小点集使得树上的每个点都存在一个集合中的点与其相连。

两个问题很有代表性，这里讲一下解法。

第一个问题

第一个问题比较好解决，考虑令 $f(x)$ 表示 x 点在集合中，以 x 为根的子树能选取的最多点数，$g(x)$ 表示 x 点不在集合中，以 x 为根的子树能选取的最多点数。
考虑按照题意的合法性转移即可。

$$f(x) = \sum g(x)$$

$$g(x) = \sum \max\{g(son), f(son)\}$$

第二个问题

这个问题相对复杂，我们称选择的点能够“覆盖”与其相连的点，那么考虑一个点的合法状态有 3 种，分别设选改点的状态为 $f(x)$，这个点被儿子覆盖为 $g(x)$，这个点被父亲覆盖为 $h(x)$

f, h 函数的转移都很简单，对于 g，分类讨论即可。

DP 的两种处理方法

前面默认我们都是使用了 *DFS* 来递归处理子树，然后回溯更新节点，但是实际操作中会存在问题。
Windows 下默认栈空间大小为 $4Mb$, *Linux* 下为 $8Mb$, 大量递归会堆栈溢出。
考虑这个转移的过程只需要所有的儿子都被更新完。我们 *BFS* 这颗树，然后倒过来处理 *DP* 是可以得到同样的效果的。
至此我们解决了这个问题。

树形 DP 为什么相较其他 DP 来比有难度

树形 DP 为什么相较其他 DP 来比有难度

1. 在树上进行，相较于序列，更新的方式更多，对思维难度和代码实现难度要求都更高。

树形 DP 为什么相较其他 DP 来比有难度

1. 在树上进行，相较于序列，更新的方式更多，对思维难度和代码实现难度要求都更高。

2. 对 DP 优化的考察更为明显，如何通过更优秀的状态表示将一个复杂度更高的动态规划降维。

树形 DP 为什么相较其他 DP 来比有难度
1. 在树上进行，相较于序列，更新的方式更多，对思维难度和代码实现难度要求都更高。
2. 对 DP 优化的考察更为明显，如何通过更优秀的状态表示将一个复杂度更高的动态规划降维。
3. 背包问题的拓展以及树上背包。
接下来我们将通过习题来解决这些问题。

POJ 1463

给定一棵树，被选定的点可以覆盖所有和他向连的边，求覆盖所有的边需要最少多少个点？$n \leq 10^5$。

POJ 1463

给定一棵树，被选定的点可以覆盖所有和他向连的边，求覆盖所有的边需要最少多少个点？$n \leq 10^5$。
$f(x)$ 表示选定 x, 保证子树合法，子树内所需要的最小点数，
$g(x)$ 表示不选择 x, 保证子树合法所需要的最小点数

$$f(x) = \sum \min\{f(son[x]), g(son[x])\}$$

$$g(x) = \sum f(son[x])$$

POJ 1155

一个树形网络，编号为 1 的是广播站，叶子节点为广播接收者，到达叶子节点有收益，但是在节点间传播需要花费该边边权的价值。
问在保证广播站收益不亏本的情况下最多能选择多少叶子结点？数据范围支持 n^3

POJ 1155

一个树形网络，编号为 1 的是广播站，叶子节点为广播接收者，到达叶子节点有收益，但是在节点间传播需要花费该边边权的价值。
问在保证广播站收益不亏本的情况下最多能选择多少叶子结点？数据范围支持 n^3
如果根节点直接连接所有叶子，那这个问题就是个 01 背包。

POJ 1155

一个树形网络，编号为 1 的是广播站，叶子节点为广播接收者，到达叶子节点有收益，但是在节点间传播需要花费该边边权的价值。
问在保证广播站收益不亏本的情况下最多能选择多少叶子结点？数据范围支持 n^3
如果根节点直接连接所有叶子，那这个问题就是个 01 背包。
考虑到树上怎么做，$f(i,j)$ 表示 i 节点下方有 j 个叶子的收益。

$$f(i,j) = \max\{f(i,j) + f(son, j-k) - val_{i \to son}\}$$

注意对最开始不合法情况的初始化。

POJ 1947

将一棵 n 个节点的有根树，删掉一些边变成恰有 m 个节点的新树。求最少需要去掉几条边。数据范围支持 n^2m

POJ 1947

将一棵 n 个节点的有根树，删掉一些边变成恰有 m 个节点的新树。求最少需要去掉几条边。数据范围支持 n^2m

令 $f(i,j)$ 保留 i, 在 i 的子树中选出一个有 j 个节点的新树最少需要去掉几条边。

POJ 1947

将一棵 n 个节点的有根树，删掉一些边变成恰有 m 个节点的新树。求最少需要去掉几条边。数据范围支持 n^2m

令 $f(i,j)$ 保留 i, 在 i 的子树中选出一个有 j 个节点的新树最少需要去掉几条边。

更新的时候便利每一个儿子，同时每次需要首先将 $f(i,j)$ 更新为 $f(i,j)+1$ 即可以直接切掉和这个儿子的边，然后考虑如果不砍掉这条边该怎么做

$$f(i,j) = \min\{f(i,k) + f(son, j-k)\}$$

另外还有每个节点的初始化问题，请各位自己考虑。

POJ 1935

给你一个树，一些点和一个根，从根出发遍历给定的点（任意顺序），最后不必回根的最小路径. $n \leq 2 \times 10^5$

POJ 1935

给你一个树，一些点和一个根，从根出发遍历给定的点（任意顺序），最后不必回根的最小路径. $n \leq 2 \times 10^5$
如果你精通数据结构的话，这题会有一些高端做法，支持数据范围扩大若干倍。这里只考虑这个问题。
首先我们一定是停在一个距离根最远的点，答案就是往返-最远点到跟的距离。求每个点到跟的距离很简单，考虑如果求出过给定点的距离和。

POJ 1935

给你一个树，一些点和一个根，从根出发遍历给定的点（任意顺序），最后不必回根的最小路径. $n \leq 2 \times 10^5$

如果你精通数据结构的话，这题会有一些高端做法，支持数据范围扩大若干倍。这里只考虑这个问题。

首先我们一定是停在一个距离根最远的点，答案就是往返-最远点到跟的距离。求每个点到跟的距离很简单，考虑如果求出过给定点的距离和。

做法也很简单，如果一个点被标记，那就沿着这棵树向上爬，爬到最后一个没被标记的点，并更新答案，将沿途标记。

每个点最多被标记一次，所以复杂度 $O(n)$.

小结

通过上面的题，我们发现，这些题还是比较简单的，状态表示常常与他给的限制和一些必要的元素组成，往往还会与背包相联系。

然而这些都是之前非常熟练的东西，我们只需要简单的处理一些小的细节。

贪吃的九头龙

N 个节点的一棵树被 M 个脑袋吃，每个脑袋至少吃一个。最大的头必须恰好吃 K 个且必须包括 1 号节点。
如果一条树边的两边都是被同一个脑袋吃掉的，则这段树枝的权值将被计算进答案中，要求使答案最小。
$N, M, K \leq 100$

首先无解很容易判断，只需要考虑 $N-K$ 与 $M-1$ 的情况。这个题和前面的题的区别就在于这题有 M 个人在进行背包，显然弄一个 M 维背包维护 M 个人不现实。

首先无解很容易判断，只需要考虑 $N-K$ 与 $M-1$ 的情况。
这个题和前面的题的区别就在于这题有 M 个人在进行背包，显然弄一个 M 维背包维护 M 个人不现实。
但是 $M>2$ 时总存在一种方案不让这 M 个吃树枝，道理显然。
剩下的就和前面一样了，$f(i,j,k)$ 表示 i 节点子树，大头吃了 j 个，k=0 表示是大头吃的，k=1 是小头吃的。背包即可。

树网的核

在一个 n 个节点的树的直径上找一条 $\leq s$ 的路径，使得树中的点到路径的最远点最近。

对于 30% 的数据 $n \leq 300$

对于 100% 的数据 $n \leq 5000000$

30pts 做法

这个是原题的数据范围，欢迎大家随意 YY

100pts

先找出直径，把他“伸直”，其他连在直径上的点变为若干个子树挂在这个直径下面。
我们在直径上进行决策，就是在直径上选出了一段区间 $\leq s$，这段区间要怎么决策呢？

100pts

先找出直径，把他“伸直”，其他连在直径上的点变为若干个子树挂在这个直径下面。

我们在直径上进行决策，就是在直径上选出了一段区间 $\leq s$，这段区间要怎么决策呢？

求出每一个小树到直径上对应根的距离 dis_{MAX}。在决策的时候，可以发现，决策点只有区间两边的端点到直径两端点的距离，以及区间上最大的 dis_{MAX}。

为什么没有区间外的 dis_{MAX}, 显然，如果他的 dis_{MAX} 比直径的端点到区间左端点还大，那么现在的直径就不是直径了
然后我们只需要对一个长度 $\leq s$ 的一段区间，维护一个单调队列，每次决策就行了时间复杂度 $O(n)$

BZOJ 2907

给定一棵边权都是 1 的树，可以删掉 k 条边，求删掉后森林中所有树直径的最大值的最小值。$n \le 2 \times 10^5$

BZOJ 2907

给定一棵边权都是 1 的树，可以删掉 k 条边，求删掉后森林中所有树直径的最大值的最小值。$n \leq 2 \times 10^5$
最大值的最小值问题，我们容易想到是二分问题，但是怎么验证？

BZOJ 2907

给定一棵边权都是 1 的树，可以删掉 k 条边，求删掉后森林中所有树直径的最大值的最小值。$n \le 2 \times 10^5$
最大值的最小值问题，我们容易想到是二分问题，但是怎么验证？
把一个节点所有向下的链都拿出来，如果最大加次大比二分的答案就删掉这个和这个子树相连的边。
时间复杂度 $O(n \log^2(n))$, 贪心的正确性证明请各位自行解决。

BZOJ 1131

给出一个 N 个点的树, 找出一个点来, 以这个点为根的树时, 所有点的深度之和最大 $N \leq 1000000$.

BZOJ 1131

给出一个 N 个点的树, 找出一个点来, 以这个点为根的树时, 所有点的深度之和最大 $N \leq 1000000$.
一种比较显然的做法是对每个点为跟跑一个树形 DP, 然而复杂度不允许。
考虑一个点为根转变成其儿子变成根的差量即可。

BZOJ 1060

给定一棵树，你每一次可以使一条边的边权 +1 求想让根节点到所有的叶子节点的距离相等最小需要几次

BZOJ 1060

给定一棵树，你每一次可以使一条边的边权 +1 求想让根节点到所有的叶子节点的距离相等最小需要几次
我们从底层开始算起，显然在根上修改比较省时，然而这个的前提是下面都相等，进行多棵子树的调节递归做下去, 先把下面改成相等，然后向上一层层做就行了

BZOJ 1304

给一棵 m 个结点的无根树，你可以选择一个度数大于 1 的结点作为根，然后给一些结点（根、内部结点和叶子均可）着以黑色或白色。
你的着色方案应该保证根结点到每个叶子的简单路径上都至少包含一个有色结点（哪怕是这个叶子本身）。
对于每个叶结点 u，定义 c_u 为从根结点从 u 的简单路径上最后一个有色结点的颜色。给出每个 c_u 的值，设计着色方案，使得着色结点的个数尽量少。

容易发现可以随便选择一个点当根。

容易发现可以随便选择一个点当根。
用 $f(x, opt)$ 表示 x 的子树中，最后一个点想要得到一个白色/黑色的祖先，的最小代价
对于同一个节点及其子树，并不会在他的儿子上同时存在两个矛盾。因为如果同时存在，不管前面放哪个颜色，都会使另一个不合法.

容易发现可以随便选择一个点当根。
用 $f(x, opt)$ 表示 x 的子树中，最后一个点想要得到一个白色/黑色的祖先，的最小代价
对于同一个节点及其子树，并不会在他的儿子上同时存在两个矛盾。因为如果同时存在，不管前面放哪个颜色，都会使另一个不合法.
用 f_i 表示 i 节点及其子树全部合法的最小染色数
用 g_i 表示 i 节点及其子树只因为白色的叶子而不合法的最小染色数
用 h_i 表示 i 节点及其子树只因为黑色的叶子而不合法的最小染色数

令

$$x = \sum f(v)$$

$$y = \sum \min\{f(v), g(v)\}$$

$$z = \sum \min\{f(v), h(v)\}$$

然后就能转移了……

$$f(i) = \min\{x, y + 1, z + 1\}$$

$$g(i) = \min\{y, z + 1\}$$

$$h(i) = \min\{y + 1, z\}$$

答案就是 $f(root)$

BZOJ 2657

给定一个三角剖分之后的凸多边形，求连接凸多边形的两个顶点的线段能经过的最多的三角形数

BZOJ 2657

给定一个三角剖分之后的凸多边形，求连接凸多边形的两个顶点的线段能经过的最多的三角形数

平面图转对偶图，剩下的问题发现就是求直径……

BZOJ 4033

有一棵点数为 N 的树，树边有边权。给定一个数 K ，你要在这棵树中选择 K 个点，将其染成黑色，并将其他的 N-K 个点染成白色。将所有点染色后，你会获得黑点两两之间的距离加上白点两两之间的距离的和的受益。问受益最大值是多少。
$N \leq 5000$

上套路！$f(i,j)$ 表示以 i 为根的子树选了 j 个黑点的答案

上套路！$f(i, j)$ 表示以 i 为根的子树选了 j 个黑点的答案
直接算答案的贡献不好算，我们把他拆开，分到每条边上，
如果一条边左边 a 个点右面 b 个点，那么他对答案的贡献就是 $ab \times val$
剩下的就是 n^2 DP 了，注意枚举范围要枚举的正好，全枚举就成 n^3 的了。

一棵 n 个点的树有边权的树，第 i 个结点居住着 a_i 个人。假设在 i 结点举行会议，所有人都从原住址沿着最短路径来到 i 结点，行走的总路程为 b_i。输出所有 b_i。” 吉丽已经造好了数据，但熊孩子把输入文件中所有 a_i 给删掉了。你能帮他恢复吗？

首先，如果是给你 a_i，让你求 b_i 还是非常简单的，刚刚讲完。我们可以求出一个点及其子树的 a 的和，设为 sum 那么从一个点的父亲转移过来就是

$$b[x] = b[fa[x]] + sum[1] - 2 * sum[x]$$

移项得

$$2sum[x] = b[fa[x]] - b[x] + sum[1]$$

所以我们可以遍历整棵树求出 sum_x 与 sum_1 的函数关系

首先，如果是给你 a_i，让你求 b_i 还是非常简单的，刚刚讲完。我们可以求出一个点及其子树的 a 的和，设为 sum 那么从一个点的父亲转移过来就是

$$b[x] = b[fa[x]] + sum[1] - 2*sum[x]$$

移项得

$$2sum[x] = b[fa[x]] - b[x] + sum[1]$$

所以我们可以遍历整棵树求出 sum_x 与 sum_1 的函数关系

而由原始的定义可知

$$b[1] = \sum a_i \times dis_i$$

我们可以扫一遍，求出 sum_1，然后再扫一遍，用 sum_1，求出 $a_2, a_3, a_4 \ldots, a_n$
最后就得到了 a_1

BZOJ 3037

给出一个每个点都有一个依赖节点的图，选择一个节点必须选择这个节点的依赖节点，才会得到这个节点的权值。
每个点有一个空间，给出总空间限制，问最多可以获得多少权值。

BZOJ 3037

给出一个每个点都有一个依赖节点的图，选择一个节点必须选择这个节点的依赖节点，才会得到这个节点的权值。
每个点有一个空间，给出总空间限制，问最多可以获得多少权值。
这个题非常简单……我们 *Tarjan* 缩点之后跑背包就好了……

BZOJ 1040

有一个 n 个人的军团，每个人都有一个自己不喜欢的人，以及自己的战斗力。
现在请选出一些人，使得他们的战斗力最大，且对于每个在这个集合里的人，他不喜欢的人都不在这个集合里。
$n \leq 1000000$

先考虑建模，容易发现如果 A 不喜欢 B，那么 A, B 就不可能在一个集合里。这时我们在 A, B 中间连一条边，那么现在就是在求这个图的最大独立集了。
N 个人 N 条边，一定是一个环和若干个树。我们可以把环当成根，然后其他接在换上的部分作为这个新根的子树。这个结构其实叫基环树。
考虑把环从任意一处断开，并记两边的节点分别为 u, v, 那么我们可以以 u 为根跑一个一定不选 u 的独立集，再以 v 为根跑一个一定不选 v 的独立集，二者取 $\max$ 即可。

BZOJ 3037

给定一张有向图，每个点有且仅有一条出边，要求如果选择了一个点 x，那么至少存在一个未选择的点 y，y 的出边指向 x，求最多选择去多少个点。

BZOJ 3037

给定一张有向图，每个点有且仅有一条出边，要求如果选择了一个点 x，那么至少存在一个未选择的点 y，y 的出边指向 x，求最多选择去多少个点。
正着做比较难，因为限制一个元素的元素可能很多。

BZOJ 3037

给定一张有向图，每个点有且仅有一条出边，要求如果选择了一个点 x，那么至少存在一个未选择的点 y，y 的出边指向 x，求最多选择去多少个点。
正着做比较难，因为限制一个元素的元素可能很多。
考虑反向求最少留下多少点，发现每个点要么留下来，要么被限制他的点覆盖。现在就变成了最小支配集。

设 $f(x)$ 为以 x 为根的子树选择 x 的最小支配集 $g(x)$ 为不选择 x 的最小支配集
由于是基环树林所以我们选择一个环上的点拆掉它的出边设这个点为 x, 出边指向的点为 y 讨论:
1. 若 x 选择则 y 一开始就是被支配状态 $g(y)$ 初值为 0 求一遍最小支配集
2. 若 x 不选正常求最小支配集即可
两种情况取最小值计入 ans 最后输出 n-ans 即可

BZOJ 1791

求一个基环树森林的直径。

BZOJ 1791

求一个基环树森林的直径。
先把非环部分的最长链用 DP 求出来，维护出非环部分的直径。

BZOJ 1791

求一个基环树森林的直径。
先把非环部分的最长链用 DP 求出来，维护出非环部分的直径。
再把最长链记为 M_i，在环上跑一个 $dis_i + M_i$ 的单调队列即可。