动态规划的拓扑延拓

谢兴宇

计算机科学与技术系，清华大学

拓扑结构及符号约定

良基结构

- 序列：$\{a_n\}$
- 环
- 树：T
- 环套树

从经典问题出发

序列上的经典动态规划问题

- 最大带权独立集
- 最大子段和
- 最长上升子序列

最大带权独立集

$ans = \max_{1 \le i_1, i_2, \cdots, i_k \le n, \forall 1 \le j < k, i_{j+1} - i_j > 1} \sum_{j=1}^{k} a_{i_j}$

扩展：有多少独立集其权值和是最大的？

最大带权独立集：解法

一些可行的DP状态：

- f_i表示前i个数中最大带权独立集的权值
- f_i表示前i个数中，取到i的最大带权独立集的权值
- $f_{i,0/1}$表示前i个数，第i个数取/不取的最大带权独立集的权值

最大子段和

$ans = \max_{1 \le l \le r \le n} \sum_{i=l}^{r} a_i$

$n \le 10^7$

最大子段和：解法一

f_i表示以i为结尾的最大子段和（可能为空），可写出状态转移方程：

$f_i = \max(f_{i-1} + a_i, 0)$。

设计状态时的细节

对状态做细微的修改，状态转移方程和最终的答案会发生什么样的变化？

- 以i为结尾的可能为空的最大子段和
- 以i为结尾的不能为空的最大子段和
- 前i个数的可能为空的最大子段和
- 前i个数的不能为空的最大子段和

God is in detail.

最大子段和：解法二

记$I = \{[l, r] | 1 \le l \le r \le n\}$.

记$\alpha \in I$为最大子段和对应的区间，$\beta \in I$为最短负前缀。

则或者α与β无交，或者α为β的前缀。

```
x = ans = 0;
for (int i = 1; i <= n; ++i)
        ans = max(ans, x = max(x + a[i], 0));
```

同一份代码，不同的解读。

最大子段和：解法三

记$s_i = \sum_{j=1}^{i} a_j$

$$ans = \max_{1\le l\le r\le n} \sum_{i=l}^{r} a_i$$
$$= \max_{0\le l<r\le n}(s_r - s_l)$$
$$= \max_{1\le r\le n}(s_r - \min_{0\le l<r} s_l)$$

从区间和到前缀和。

最大子段和：总结

- 线性序列上的DP
- 对子串关系的分析
- 序列问题的杀器：前缀和

扩展

最大子段和的方案数？即，有多少区间，其中元素的和是最大的？

支持单点修改、区间查询的动态最大子段和？

最长上升子序列

$P = (a_{i_1}, a_{i_2}, \cdots, a_{i_k}), i_1 < i_2 < \cdots < i_k, a_{i_1} < a_{i_2} < \cdots < a_{i_k}$

$ans = \max |P|$

$n \leq 10^3$

$n \leq 10^6$

最长上升子序列：解法

以f_i表示以i结尾的最长上升子序列的长度。

$f_i = \max_{1 \le j < i, a_j < a_i} f_j + 1$

如何求$\max$？

最长上升子序列：解法一

从数据结构的角度来看，需要支持：

- 单点修改
- 求前缀$\max$

可行的方案：

- 枚举：$O(n^2)$
- 线段树：$O(n\log n)$
- 树状数组：$O(n\log n)$

最长上升子序列：解法二

按$f_1, f_2, \cdots, f_n$的顺序求f_i，并维护数组g，$g_k = \min_{f_i=k} a_i$.

最长上升子序列：几何直观

从一维序列到二维平面。

解法回顾：另一个视角。

扩展：最长上升子序列方案数

有多少个上升子序列同为最长的？

$n \leq 1000$

$n \leq 10^6$

最长上升子序列的方案数：解法

- 暴力：$O(n^2)$
- 树状数组：不再可行
- 线段树：依然可行
- g数组：需要改成数组套`vector`，`vector`中存a_i和cnt_i。

环形DP

两种方法：

- 将环倍长
- 破环为链

环上最大带权独立集

在环$\{a_n\}$中选择若干个不相邻的数，使其和最大。

$n \leq 10^7$

平凡扩展：求最大带权独立集的数量。

环上最大带权独立集：解法

破坏为链：枚举第一个数选或者不选。

环形最大子段和

$ans = \max_{0 \le i < n, 0 \le m \le n} \sum_{j=0}^{m-1} a_{1+(i+j)\%n}$

$n \le 10^7$

平凡扩展：求环形最大子段和的数量。

环形最大子段和：解法一

倍长。

$ans = \max_{0 \le l < r \le 2n, r-l \le n}(s_r - s_l)$

怎么求$\max$？

- 线段树/树状数组：$O(n \log n)$
- ST表：$O(n \log n)$
- 笛卡尔树：$O(n)$
- 单调队列：$O(n)$
- 拼接前后缀$\max$：$O(n)$

环形最大子段和：解法二

破环为链。

若a_1和a_n中有一个没选，便是序列上的情况。

若a_1和a_n都选了，则 $ans = \max_{1 \le i < j \le n} presum_i + suffixsum_j$。

环形LIS

要求LIS首尾不能相交。

$n \le 10^4$

环形LIS：解法

记$A_k = \{i|f_i = k\}$，使用一个数组套链表来维护A_k。支持：

- 在末尾添加一个新元素
- 在开头删除一个新元素

树形DP

一般在LCA处统计信息：链，连通集，子树。

连通集

$LCA(a, b)$满足交换律、结合律。

因此，我们可以定义点集的$LCA(A), A \subset V$

Question: 若A为连通点集，是否有$LCA(A) \in A$?

树的直径

求有边权树T的直径（距离最远的点对的距离）?

$|T| \leq 4 * 10^6$

可以考虑一下两种情况：

- 边权非负
- 边权可以为负数

树的直径：解法一

若边权非负：贪心。

从任意一点x出发，找到距离x最远的点u，再找到距离u最远的点v，(u,v)即为直径。

定理：树T中任一点x，y是距离x最远的点，则必有点u，使得(u,y)为树T的一条直径。

定理：树T中任一点x和直径(u,v)，u或v是距离x最远的点。

树的直径：解法二

若边权可为负数：DP。

记$f_i = \max_{j \in subtree_i} d_{i,j}$。

转移方程：$f_i = \max_{j \in child_i}(f_j + w_{i,j})$

$ans = \max_i \max(\max_{j \neq k \in child_i}(f_j + w_{i,j} + f_k + w_{i,k}), f_i)$

树上最远距离

求树中距离每个点最远的点的距离。

$n \le 4 * 10^6$

考虑以下两种情况：

- 边权非负
- 边权可为负

树上最远距离：解法

先处理出上一题的f。

以g表示i往上的最远距离。

$$g_i = \max(g_{parent_i}, \max_{j \in sibling_i}(f_j + w_{j,parent_i})) + w_{i,parent_i}$$

$$ans_i = \max(f_i, g_i)$$

树上LIS

在树上选出一条路径，使得这条路径的LIS最长，求这个最长的LIS的长度。

$n \le 5000$

$n \le 10^5$

树上LIS：解法一

记f_i为从i向下走的最长上升子序列的长度，记g_i为从i向下走的最长下降子序列的长度。

$ans = \max_{i \notin subtree_j, j \notin subtree_i, a_i < a_j} \{g_i + f_j\}$

时间复杂度：$O(n^2)$

树上LIS：解法二

点分治，配合g数组的LIS解法。

时间复杂度：$O(n\log^2 n)$

树上最大独立集

在树T树上选出一些点，使它们两两不相邻，这样的点集被称为独立集。

$|T| \le 4 * 10^6$

可以考虑以下两种情况：

- 点权均为1（无点权）
- 有点权

树上最大独立集：解法一

若无点权：贪心。

不断选择一个叶节点，删掉与之相邻的节点。

树上最大独立集：解法二

若有点权：DP

$f_{i,0/1}$表示在以i为根的子树中，i不选/选时，最大带权独立集的权值。

$f_{i,1} = \sum_{j \in child_i} f_{j,0}$

$f_{i,0} = \sum_{j \in child_i} \max(f_{j,0}, f_{j,1})$

$ans = \max(f_{root,0}, f_{root,1})$

树形背包

问树中大小为$1, 2, \cdots, n$的连通块有多少个?

$n \leq 500$

$n \leq 5000$

树形背包：解法

记$f_{i,j}$为以i为LCA有多少大小为j的连通块。

$f_{i,*}$可由$f_{j,*}(j \in child_i)$做背包合并而来（当然，也可以理解为多项式卷积）。

时间复杂度似乎是$O(n^3)$的。

如果我们限制j的范围为$size_i$，时间复杂度降为$O(n^2)$。

树形背包：扩展

树中大小为$\boldsymbol{k}$的连通块有多少个？

$n \leq 10^4, k \leq 100$

有依赖的树形DP

一类对形如“若x满足，则$parent_x$也满足”的条件的处理方法。

$f_{i,j}$表示dfs序在i及之后的所有节点中选出j个点的方案数。

则$f_{i,j} = f_{i+1,j-1} + f_{i+size_i,j}$。

$f_{1,k}$即为包含dfs序为1的节点的大小为k的连通块的数量，求出$f_{1,k}$的时间代价为$O(nk)$。

配合点分治便可以做到$O(nk\log n)$。

环套树的直径

有边权的无向环套树上两点间距离的最大值，两点之间的距离指的是两点之间的一条简单路径的长度。

$n \leq 3 * 10^6$

环套树的直径：解法

不经过环上一边的两点间的距离即为每棵树的直径的最大值。

考虑处理经过环上一边的两点间的距离，处理出每棵树根节点向下走的最远距离f_i，记环上连接i和$i-1$的边权为a_i，则这部分对答案的贡献为：$\max_{r-l\le m,0<l<r\le 2m}(s_r-s_l+f_l+f_r)$。

用区间RMQ的线性解法/单调队列/拼接前后缀$\max$的方法可以做到$O(n)$。

环套树上的最大独立集

考虑两种情况：

- 有点权
- 无点权

$n \le 10^6$

环套树上的最大独立集：解法

若无点权，类似树的做法，同样可以贪心。

若有点权，也可以先做树形DP，然后再在环上DP。一个更简便的写法时，直接枚举环上一点选/不选，这样以此为根做树形DP。

环套树上的LIS

$n \le 100$

更大的n?

留待思考……

参考资料

- 刘汝佳. 算法竞赛入门经典. 清华大学出版社.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms(Third Edition). MIT Press.
- https://en.wikipedia.org/
- https://oi-wiki.org/