Maximum Tree Merge

最大二叉树合并

问题：

拥有个节点的二叉树，按照中序遍历将所有节点标记为，如图：



节点拥有价值，将子树进行合并，产生的代价的计算方法是，即其左子树的合并代价乘以右子树的合并代价，再加根节点自身的价值，特别的我们规定空子树的合并代价为1。合并顺序的不同会使最终整个树的合并代价不同，求该二叉树的最大合并代价。

本问题的原型为“加分二叉树”。

解法：

将二叉树中的所有节点按照中序遍历依次编号为，根据中序遍历的性质，可知连续节点刚好属于1个子树，且在中选取节点作为根节点（），则其左子树为，右子树范围为。例如上图中，属于子树2（以2为根节点的子树），属于子树6。设为以节点组成的子树的最大合并代价，其中且，其转移方程如下：

1. 将所有可能情况都初始化为最小的合并代价，即1；
2. 对于只有一个节点的子树来说，其合并代价为自身根节点的价值加1，即，因为左右子树都是空子树，其合并代价为1；
3. 将分为和左右两个子树，则，其中。在范围内遍历所有情况，选取最大的即可；

即为二叉树的最大合并价值。该算法的时间复杂度是。

加分二叉树：

<http://codevs.cn/problem/1090/>