给定一棵树，其中每个顶点的代价记为，树中一个顶点集合的代价为。记树的顶点集合为，的切面定义如下：

一个顶点集合称为树的一个切面，如果满足以下两个条件：

(1) ;

(2) 从的根顶点到任意叶子顶点的路径上有且只有一个顶点属于。

请设计算法，求解的一个切面使得最小，并分析算法的时间开销。

算法设计思路：

记以顶点为根的子树为，求解中代价最小的切面，其代价记为，即为树中切面的最小代价。记为中代价最小的切面。分治算法主要步骤如下：

**边界条件处理：**如果顶点是一个叶子，则，。

**Divide:** 如果顶点不是一个叶子，记为的儿子顶点集合，将划分为顶点和个子树，其中。

**Conquer:** 对于所有顶点，递归地求解和。

**Merge:** 比较和。如果，则，。如果，则，。

伪代码：

MinSection(n)

1. S(n) is a new linking list containing a single element n;

2. if n is a leaf

3. return c(n), S(n);

4. c’=0;

5. for m in CH(n)

6. cost(m), S(m) = MinSection(n);

7. c’←c’+cost(m);

8. if c(n) c’

9. return c(n), S(n);

10. S(n) is a new linking list with a single element #;

11. for m in CH(n)

12. S(n).tail.next←S(m).head;

13. S(n).tail←S(m).tail;

14. remove the first element # from S(n);

15. return c’, S(n);

运行MinSection(T.root)计算T的切面.

算法时间复杂性分析：

求解中最小切面代价记为。

边界条件处理代价；

Divide代价: ；

Conquer代价: ;

Merge代价: ；

总体代价: ，。

空间代价: 任何时刻，每个顶点只需要保存在1个顶点对应的切面中（用链表保存切面中的顶点），因此空间代价为。