

一、题目说明

题目96. Unique Binary Search Trees, 求1-n节点组成的二叉排序树的个数。

二、我的解答

首先, 我枚举了 $G(1)=1, G(2)=2, G(3)=5, G(4)=14$, 在枚举的过程中, 我们知道: 1-n的二叉搜索树, 包括以1,2,...n为根的所有二叉树的总数。以i为根, 左边为i-1个数, 右边n-i个数, 故:

$$G(n) = G(0)*G(n-1)+G(1)*(n-2)+\dots+G(n-1)*G(0)$$

有了这个有可以用动态规划做了:

```
class Solution{
public:
    int numTrees(int n){
        vector<int> dp;
        dp.push_back(1); //以0为根1个
        dp.push_back(1); //以1为根1个
        for(int i = 2; i <= n; i ++){
            int sum = 0;
            for(int j = 1; j <= i; j++){
                sum += dp[j-1] * dp[i-j];
            }
            dp.push_back(sum);
        }
        return dp[n];
    }
};
```

性能如下:

```
Runtime: 0 ms, faster than 100.00% of C++ online submissions for Unique Binary Search Trees.
Memory Usage: 8.3 MB, less than 59.09% of C++ online submissions for Unique Binary Search Trees.
```

三、优化措施

用dfs试试, 其实理解了原理, 做起来还是简单的, numTrees(18)=477638700, 本地运行1.3s, 提交后**Time Limit Exceeded**:

```
class Solution{
public:
    int dfs(int start,int end){
        if(start>end) return 1; //空
        int ans = 0;
        for(int i=start;i<=end;i++){
            int left = dfs(start,i-1);
            int right = dfs(i+1,end);
            ans += left*right;
        }
    }
};
```

```
        return ans;
    }
    int numTrees(int n){
        if(n==0) return 1;
        return dfs(1,n);
    }
};
```