动态规划的基本步骤:
1、找出最优解的性质，并刻画其结构的特征
2、递归地定义最优值
3、以自底向上的方式计算最优值
4、根据计算最优值时得到的信息，构造最优解
a 步骤 1-3 是动态规划算法的基本步骤。如果只需要求出最优值的情形，步骤 4 可以省略。
B 若需要求出问题的一个最优解，则必须执行步骤 4，步骤 3 中记录的信息是构造最优解的基础。

1、找到最优二叉查找树的性质，并刻画其结构的特征

a、找出最优二叉查找树的性质

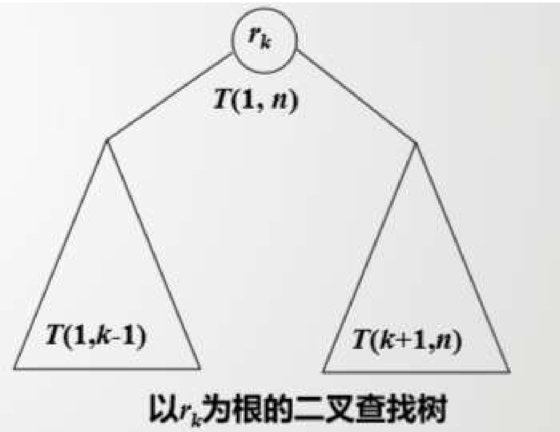
设$\{r_1, r_2, \cdots, r_n\}$ 是 n 个记录的集合，其查找概率分别是 $\{p_1, p_2, \cdots, p_n\}$，最优二叉查找树是以这 n 个记录构成的二叉查找树中具有最少平均比较次数的二叉查找树，即

$$\sum_{i=1}^{n} p_i \times c_i \quad \text{最小}$$

其中 pi 是记录 r_i 的查找概率，c_i 是在二叉查找树中查找 r_i 的比较次数

b、刻画最优二叉查找树的特征

将由 $\{r_1, r_2, \cdots, r_n\}$ 构成的二叉查找树即为 T(1, n)，其中 r_k（1<= k <= n）是 T(1, n) 的根结点，则其左子树 T(1, k-1) 由 $\{r_1, r_2, \cdots, r_{k-1}\}$ 构成，其右子树 T（k+1 , n）由$\{r_{k+1}, \cdots, r_n\}$ 构成。



若 T(1, n) 是最优二叉查找树，其左子树 T(1, k-1) 和右子树 T(k+1 , n) 也是最优二叉查找树，如若不然，假设 T`(1 , k-1 ) 是比 T( 1, k-1) 更优的二叉查找树，则 T`(1 , k-1) 的平均查找次数小于 T(1 , k-1)， 从而有 T`(1 , k-1 ) 、$r_k$、T( k+1 , n ) 构成的二叉查找树 T`(1 , n) 的平均查找次数小于 T(1 , n) 的平均查找次数，这与 T(1 , n) 是最优的二叉查找树的假设相矛盾。(反证法)

2、递归定义最优值

设 T(i , j)是由{ri , ⋯⋯ , rj}（1<= i <= j <= n）构成的二叉查找树，C(i , j)是这棵二叉查找树的平均比较次数。虽然最后的结果是 C(1 , n) ，但遵循动态规划法的求解方法，需要求出所有较小子问题 C(i , j) 的值。考虑从 {ri , ⋯⋯ , rj} 中选择一个记录

r_k作为二叉查找树的根结点，可以得到如下关系：

$$
\begin{aligned}
C(i,j) &= \min_{i\le k\le j}\{p_k\times 1+\sum_{s=i}^{k-1}p_s\times(r_s\text{在}T(i,k-1)\text{中的层数}+1)+\sum_{s=k+1}^{j}p_s\times(r_s\text{在}T(k+1,n)\text{中的层数}+1)\} \\
&= \min_{i\le k\le j}\{p_k+\sum_{s=i}^{k-1}p_s\times r_s\text{在}T(i,k-1)\text{中的层数}+\sum_{s=i}^{k-1}p_s+\sum_{s=k+1}^{j}p_s\times r_s\text{在}T(k+1,n)\text{中的层数}+\sum_{s=k+1}^{j}p_s\} \\
&= \min_{i\le k\le j}\{\sum_{s=i}^{k-1}p_s\times r_s\text{在}T(i,k-1)\text{中的层数}+\sum_{s=k+1}^{j}p_s\times r_s\text{在}T(k+1,n)\text{中的层数}+\sum_{s=i}^{j}p_s\} \\
&= \min_{i\le k\le j}\{C(i,k-1)+C(k+1,j)+\sum_{s=i}^{j}p_s\}
\end{aligned}
$$

因此，得到如下动态规划函数：

$$
\begin{aligned}
&C(i,i-1)=0 \quad (1\le i\le n+1) \\
&C(i,i)=p_i \quad (1\le i\le n) \\
&C(i,j)=\min\{C(i,k-1)+C(k+1,j)+\sum_{s=i}^{j}p_s\} \quad (1\le i\le j\le n,\ i\le k\le j)
\end{aligned}
$$

a、设一个二维表 C[n+1] [n+1]，其中 C[i][j]表示二叉查找树 T(i , j)的平均次数。

b、当 k=1 时，求 C[i][j] 需要用到 C[i][0]，当 k=n 时，求 C[i][j]需要用到 C[n+1][j]，所以，二维表 C[n+1][n+1] 行小标的范围为 1 ~ n+1，列下标的范围为 0 ~ n。

c、为了在求出由 $\{r_1, r_2, \cdots\cdots, r_n\}$ 构成的二叉查找树的平均比较次数的同时得到最优二叉查找树，设一个二维表 R[n+1][n+1]，其下标范围与二维表 C 相同，R[i][j]表示二叉查找树 T(i , j) 的根结点的序号。

3、以自底向上的方式计算最优值（其实包含了第 4 步）

例如，集合{A, B, C, D}的查找概率是{0.1, 0.2, 0.4, 0.3}，二维表C和R的初始情况如图所示。

	0	1	2	3	4
1	0	0.1			
2		0	0.2		
3			0	0.4	
4				0	0.3
5					0

	0	1	2	3	4
1		1			
2			2		
3				3	
4					4
5					

$$C(1,2)=\min\begin{cases} k=1: C(1,0)+C(2,2)+\sum_{s=1}^{2} p_s = 0+0.2+0.3=0.5 \\ k=2: C(1,1)+C(3,2)+\sum_{s=1}^{2} p_s = 0.1+0+0.3=0.4 \end{cases}=0.4$$

i \ j	0	1	2	3	4
1	0	0.1	0.4	1.1	1.7
2		0	0.2	0.8	1.4
3			0	0.4	1
4				0	0.3
5					0

i \ j	0	1	2	3	4
1		1	2	3	3
2			2	3	3
3				3	3
4					4
5					

① $C[i.i] = p_i \Rightarrow C[1.1] = p_1$ ......

② $C[1.2] = \min\begin{cases} k=1 & m[1.0]+m[2.2]+\sum_{z=1}^{2} p_z = 0.5 \\ k=2 & m[1.1]+m[3.2]+\sum_{z=1}^{2} p_z = 0.4 \end{cases} = 0.4 \quad k=2$

$C[2.3] = \min\begin{cases} k=2 & m[2.1]+m[3.3]+\sum_{z=2}^{3} p_z = 0.4+0.6=1 \\ k=3 & m[2.2]+m[4.3]+\sum_{z=2}^{3} p_z = 0.2+0.6=0.8 \end{cases} = 0.8 \quad k=3$

$m[3.4] = \min\begin{cases} k=3 & m[3.2]+m[4.4]+\sum_{z=3}^{4} p_z = 0.3+0.7=1 \\ k=4 & m[3.3]+m[5.4]+\sum_{z=3}^{4} p_z = 0.4+0.7=1.1 \end{cases} = 1 \quad k=3$

③ $C[1.3] = \min\begin{cases} k=1 & m[1.0]+m[2.3]+\sum_{z=1}^{3} p_z = 0.8+0.7=1.5 \\ k=2 & m[1.1]+m[3.3]+\sum_{z=1}^{3} p_z = 0.5+0.7=1.5 \\ k=3 & m[1.2]+m[4.3]+\sum_{z=1}^{3} p_z = 0.4+0.7=1.1 \end{cases}$

$= 1.1 \quad k=3$

$C[2.4] = \min\begin{cases} k=2 & m[2.1]+m[3.4]+\sum_{z=2}^{4} p_z = 1+0.9=1.9 \\ k=3 & m[2.2]+m[4.4]+\sum_{z=2}^{4} p_z = 0.5+0.9=1.4 \\ k=4 & m[2.3]+m[5.4]+\sum_{z=2}^{4} p_z = 0.8+0.9=1.7 \end{cases}$

$= 1.4 \quad k=3$

④ $m[1.4] = \begin{cases} k=1 & m[1.0]+m[2.4]+\sum_{z=1}^{4} p_z = 1.4+1=2.4 \\ k=2 & m[1.1]+m[3.4]+\sum_{z=1}^{4} p_z = 1.1+1=2.1 \\ k=3 & m[1.2]+m[4.4]+\sum_{z=1}^{4} p_z = 0.4+0.3+1=1.7 \\ k=4 & m[1.3]+m[5.4]+\sum_{z=1}^{4} p_z = 1.1+1=2.1 \end{cases}$

$= 1.7 \quad k=3$

由上述过程，可以得到最优二叉查找树的动态规划算法。