算法设计与分析实验报告

实验名称：最优二分检索树(动态规划)

一、问题陈述，相关背景、应用及研究现状的综述分析

1.问题陈述：

设 **S=(x_1,x_2,…,x_n)**是有序集，且 **$x_1<x_2<\cdots<x_n$**，已知键值和区间的存取概率分布为**(a_0,b_1,a_1,b_2,…,b_n,a_n)**，其中 **a_i** 表示相应区间的搜索概率，**b_i** 表示相应键值的搜索概率。在所有表示有序集的二叉树中找出一棵具有最小平均路长的二叉搜索树；

二、模型拟制、算法设计和正确性证明

1. 算法设计：

从最优子结构性质出发，利用归一化因子分析子树的归一化期望成本：

最优子结构性质

T_l和T_r中结点的深度是它们在T_{ij}中的对应深度level(x) / level(E)减1，所以当左右子树被视为独立的树时，其归一化期望成本分别为:

$$p'_l = \sum_{k=i}^{m-1} \bar{b}'_k * level(x_k) + \sum_{k=i-1}^{m-1} \bar{a}'_k * (level(E_k) - 1)$$

$$p'_r = \sum_{k=m+1}^{j} \bar{b}'_k * level(x_k) + \sum_{k=m}^{j} \bar{a}'_k * (level(E_k) - 1)$$

p'_l，p'_r的归一化因子分别为：$w_{i,\ m-1}$ ，$w_{m+1,\ j}$

从而推导整棵树的归一化期望成本的递推公式：

$$w_{i,j}\,p_{i,j} = w_{i,j} + \min_{i \le k \le j}\{w_{i,k-1}\,p_{i,k-1} + w_{k+1,j}\,p_{k+1,j}\}$$

定义$w_{i,j}\,p_{i,j}$为最优函数，记为m(i, j)，则m(1,n) = $w_{1,n}\,p_{1,n} = p_{1,n}$为所求的最优值。其递归计算式为：

$$m(i,j) = w_{i,j} + \min_{i \le k \le j}\{m(i,k-1) + m(k+1,j)\}, \quad i \le j$$

$$m(i,i-1) = 0, \quad 1 \le i \le n$$

$w_{i,i-1} = a_{i-1}$，$w_{i,j} = a_{i-1} + b_i + \ldots + b_j + a_j$

三、时间和空间复杂性分析

时间复杂度：

(1)子序列长度 r 从 0 递增到 n-1；

(2)对每一个 r，有(n－r)个 m(i,j)要计算；

(3)计算每个 m(i,j)要比较 r 次，找出最小值；

所以计算复杂度为：$\sum_{r=1}^{n} r(n-r) = O(n^3)$

空间复杂度：

如果按照没有使用额外空间的角度考虑

S（n）=O(1)

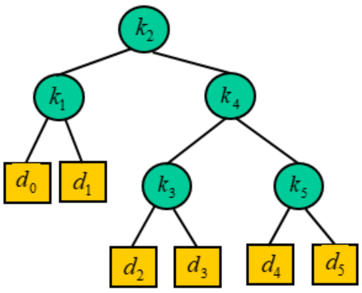
不过基础的数据存储需要二维数组

四、程序实现和实验测试过程

源程序见 best_binary_search_tree.cpp

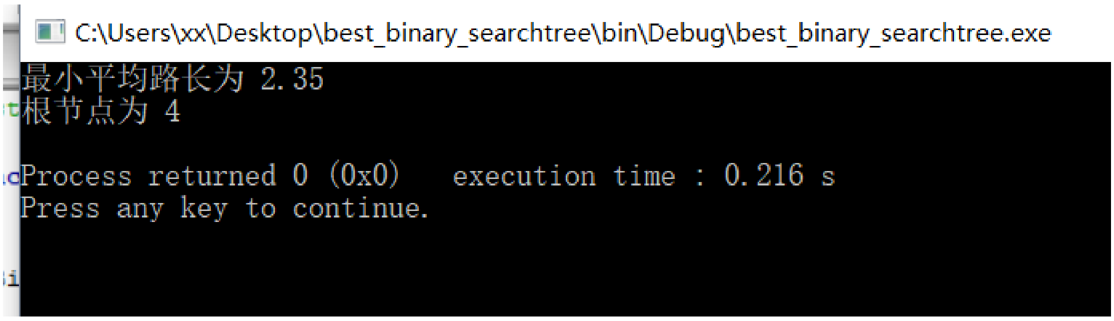
测试过程：

结点概率数据



node	depth	probability	contribution
k_1	1	0.15	0.30
k_2	0	0.10	0.10
k_3	2	0.05	0.15
k_4	1	0.10	0.20
k_5	2	0.20	0.60
d_0	2	0.05	0.10
d_1	2	0.10	0.20
d_2	3	0.05	0.15
d_3	3	0.05	0.15
d_4	3	0.05	0.15
d_5	3	0.10	0.30
Total		1	2.40

最优二叉搜索树：



```
C:\Users\xx\Desktop\best_binary_searchtree\bin\Debug\best_binary_searchtree.exe
最小平均路长为 2.35
根节点为 4

Process returned 0 (0x0)   execution time : 0.216 s
Press any key to continue.
```

五、总结

数组申请空间时要注意 Wij 和 Mij 都是从 i=1 到 i=n+1，j=0 到 j=n，因此至少要 Float M[N+2][N+2]