最优二叉搜索树

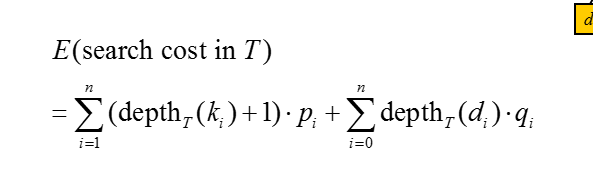
**参考书上和网上的博客，我发现一个问题，希望老师能解答一下。**

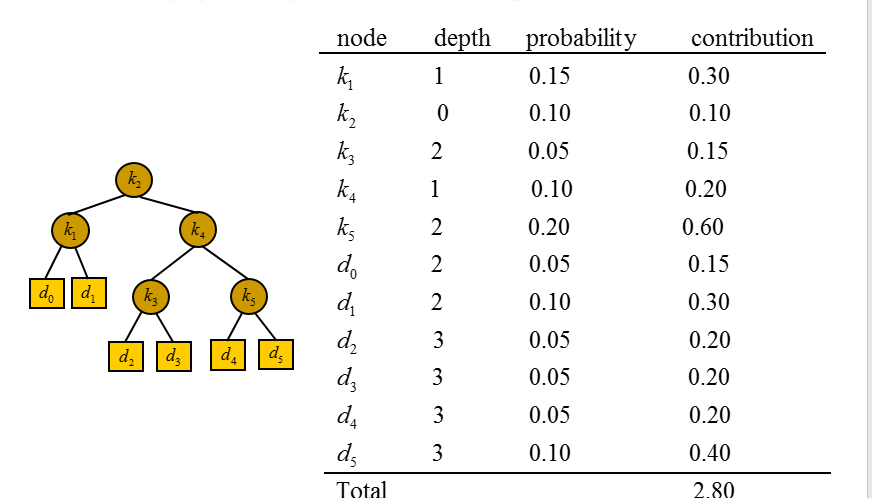
假设e[i][j]代表i到j的最少代价，

书上是e[i][i-1]=0;

而大多数博客是e[i][i-1]=q[i-1]；

本质问题就是,在ppt上您这样子写





但是ppt上计算代价时，每一个e(d0)=(2+1)\*0.05,这里又把depth+1再乘以概率。

所以当j=i-1时，按照第一幅计算d时不加1,那么e[i][j-1]=0;

但是按照第二幅图，e[i][i-1]=w[i][i-1]=q[i-1];

最后，我还是按照e[i][i-1]=q[i-1]来做了。

#include<iostream>

using namespace std;

double p[8]={0,0.04,0.06,0.08,0.02,0.1,0.12,0.14};

double q[8]={0.06,0.06,0.06,0.06,0.05,0.05,0.05,0.05};

int root[8][8];//记录根节点

double w[9][9];//从i到j概率之和

double e[9][9];//从i到j期望

void bst(double\* p,double\* q)

{

for(int i=1;i<=8;i++)

{

w[i][i-1]=q[i-1];

e[i][i-1]=q[i-1];//\*\*\*\*\*\*\*e[i][i-1]=0还是等于q[i-1]？？？？;

}

for(int len=1;len<=7;len++)

for(int i=1;i<=8-len;i++)

{

int j=i+len-1;

w[i][j]=w[i][j-1]+p[j]+q[j];

e[i][j]=0x7ffffffff;

for(int k=i;k<=j;k++)

{

double tmp=e[i][k-1]+e[k+1][j]+w[i][j];

if(tmp<e[i][j])

{

e[i][j]=tmp;

root[i][j]=k;

}

}

}

}

void printtree(int i,int j,int r)

{

int rootchild=root[i][j];

if(rootchild==root[1][7])

{

cout<<"k"<<rootchild<<"是根"<<endl;

printtree(i,rootchild-1,rootchild);

printtree(rootchild+1,j,rootchild);

return ;

}

if(j<i-1)

return;

else if(j==i-1)

{

if(j<r)

cout<<"d"<<j<<"是k"<<r<<"的左孩子"<<endl;

else

cout<<"d"<<j<<"是k"<<r<<"的右孩子"<<endl;

return ;

}

else

{

if(rootchild<r)

cout<<"k"<<rootchild<<"是k"<<r<<"的左孩子"<<endl;

else

cout<<"k"<<rootchild<<"是k"<<r<<"的左孩子"<<endl;

}

printtree(i,rootchild-1,rootchild);

printtree(rootchild+1,j,rootchild);

}

int main()

{

bst(p,q);

//printtree(1,7,-1);

cout<<e[1][7]<<endl;

system("pause");

return 0;

}

