《算法设计与分析》

第五章 动态规划

马丙鹏

2023年10月23日

中国科学院大学
University of Chinese Academy of Sciences

第五章 动态规划

- 5.1 一般方法
- 5.2 多段图问题
- 5.3 每对结点之间的最短路径
- 5.4 最优二分检索树
- 5.5 0/1背包问题
- 5.6 可靠性设计
- 5.7 货郎担问题
- 5.8 流水线调度问题

5.4 最优二分检索树

■ 1. 问题的描述

□二分检索树定义

- 二分检索树T是一棵二元树，它或者为空，或者其每个结点含有一个可以比较大小的数据元素，且有：

① T的左子树的所有元素比根结点中的元素小；

② T的右子树的所有元素比根结点中的元素大；

③ T的左子树和右子树也是二分检索树。

- 注：
  二分检索树要求树中所有结点的元素值互异。

5.4 最优二分检索树

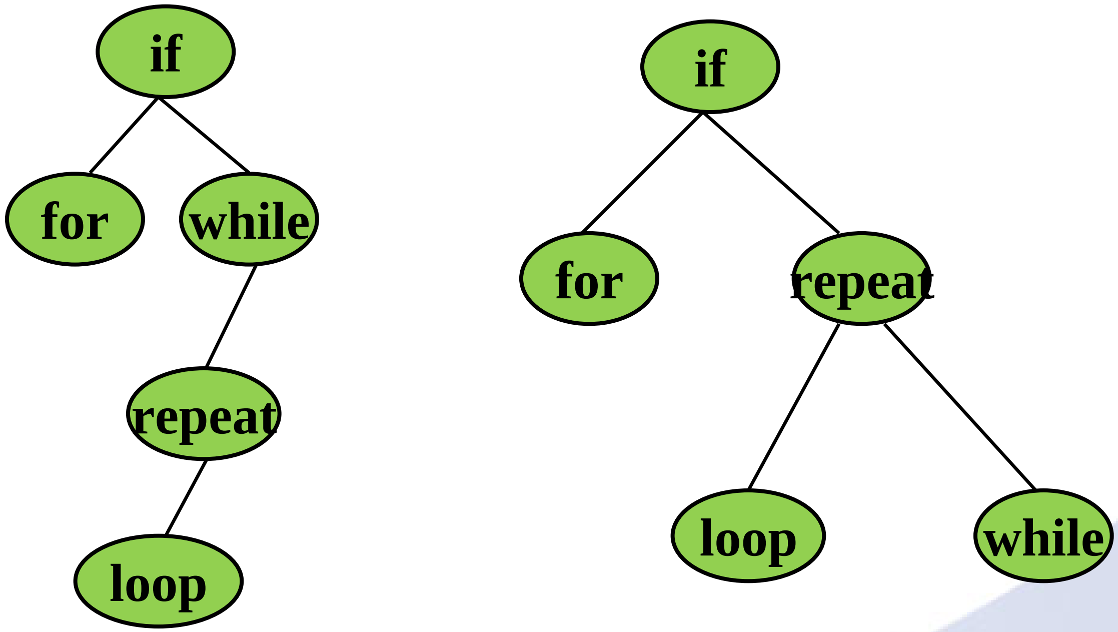
■ 1. 问题的描述

- 二分检索树定义
  - 对于一个给定的标识符集合，可能有若干棵不同的二分检索树。
  - 不同形态的二分检索树对标识符的检索性能是不同的。
  - 实例: 标志符集{a_1, a_2, a_3, a_4, a_5}=(for if loop repeat while)可能的二分检索树有如下：
    - ✓字符串的大小：以字典顺序定义

5.4 最优二分检索树

■ 1. 问题的描述

□二分检索树定义



Avg=(1+2+2+3+4)/5=2.4　　Avg=(1+2+2+3+3)/5=2.2

5.4 最优二分检索树

■ 1. 问题的描述

□二分检索树的检索

算法5.4 检索一棵二分检索树

```
procedure SEARCH(T, X, i)
//为X检索二分检索树T，如果X不在T中，则置i=0；否则i有IDENT(i)=X//
i←T
while i≠0 do
   case
     :X<IDENT(i): i←LCHILD(i) //若X小于IDENT(i)，在左子树中查找//
     :X=IDENT(i): return        //X等于IDENT(i)，返回//
     :X>IDENT(i): i←RCHILD(i) //若X大于IDENT(i)，在右子树中查找//
   endcase
repeat
end SEARCH
```

5.4 最优二分检索树

■ 1. 问题的描述

- 二分检索树的检索: 性能特征

① 例

➢ 图(a):
最坏情况下查找标识符**loop**需要做**4**次比较;
成功检索平均需要做**12/5**次比较;

➢ 图(b):
最坏情况下查找标识符**loop/while**需要做**3**次比较;
成功检索平均需要做**11/5**次比较。

5.4 最优二分检索树

■ 1. 问题的描述

- ❑二分检索树的检索: 性能特征
- ❑成功检索：
  - ➢X恰为标识符集合中的一个元素
  - ➢成功检索的情况共有n种，分别代表n个标识符
  - ➢每个标识符都有自己的检索频率

    如： $f_{for} > f_{while} > f_{loop}$

    将频率转换成概率，成功检索情况下检索概率为：

$$P_i, i=1, 2, \ldots, n$$

5.4 最优二分检索树

■ 1. 问题的描述

- □二分检索树的检索: 性能特征
- □不成功检索:
  - ➢X不是标识符集合中的任何元素
  - ➢不成功检索的情况有n+1种(类似于二分检索)
  - ➢不成功检索出现的频率也可以不同
  - ➢记不成功检索情况的检索概率为:
    Q_i, i=0, 1, …, n

中国科学院大学
University of Chinese Academy of Sciences

5.4 最优二分检索树

■ 1. 问题的描述

- □二分检索树的检索: 性能特征
- □所有成功检索和不成功检索是对同一标识符集合的所有检索需求的全集而言，即

$$\sum_{i=1}^{n} P_i + \sum_{i=0}^{n} Q_i = 1$$

$\sum_{i=1}^{n} P_i$： 表示成功检索的总概率

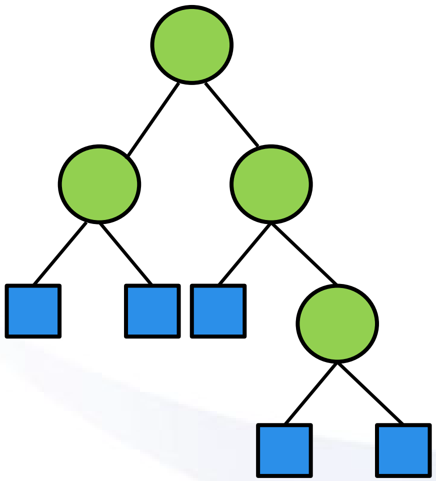
$\sum_{i=0}^{n} Q_i$： 表示不成功检索的总概率

中国科学院大学 University of Chinese Academy of Sciences

5.4 最优二分检索树

■ 1. 问题的描述

- ☐二分检索树的检索: 性能特征
- ☐标识符检索所需要的比较次数:
  - ➢成功检索：等于结点的级数，或结点到根的距离+1。
  - ➢不成功检索：在二分检索树中引入外部结点。

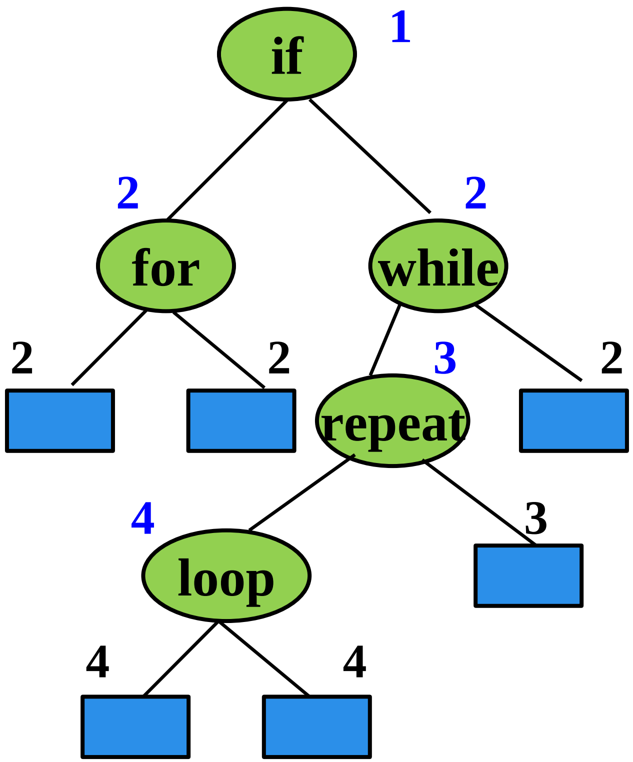


外结点作为原“内部结点树”的“叶子结点”

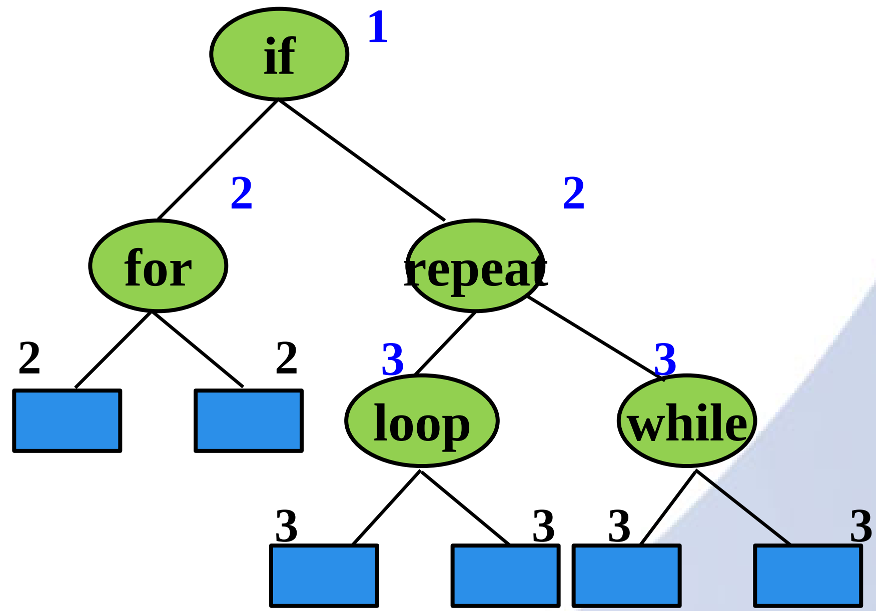
每个外部结点代表一种不成功检索的情况。

每个外部结点所表示的不成功检索需要的比较次数等于该外部结点到树根的距离，或该结点的级数-1。

5.4 最优二分检索树

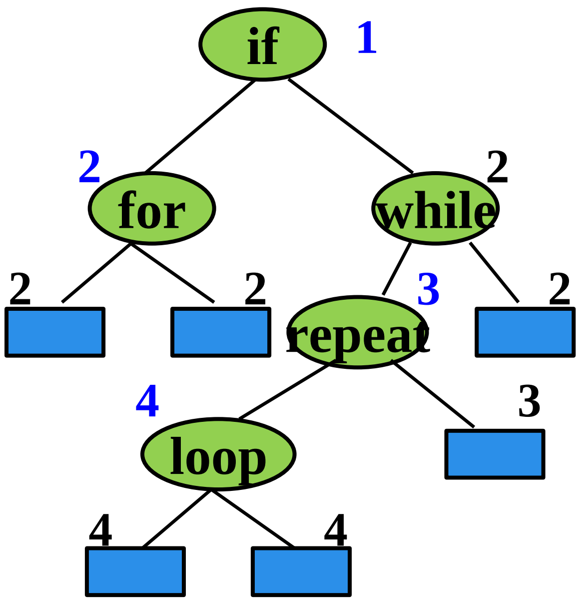


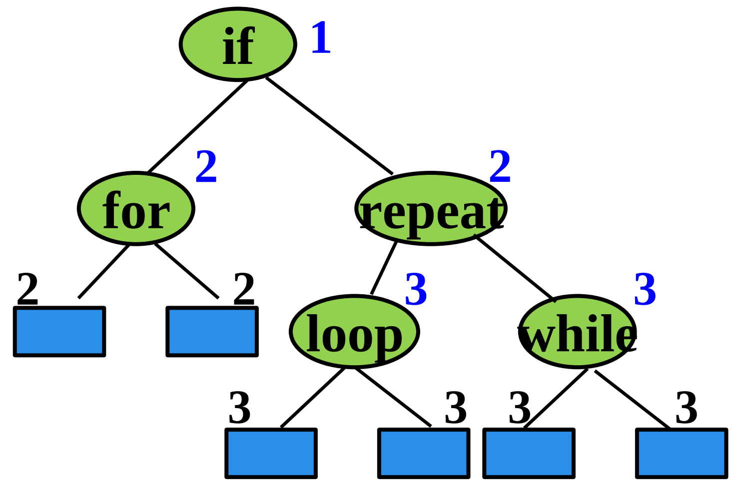
if
1
for
2
while
2
repeat
3
loop
4
2
2
2
3
4
4



if
1
for
2
repeat
2
loop
3
while
3
2
2
3
3
3
3

分析不同形态的二分检索树的检索性能





最坏：标识符loop需要做4次比较

平均：

$\text{Avg}_{成功}=(1+2+2+3+4)/5=2.4$

$\text{Avg}_{不成功}=(2+2+2+3+4+4)/6=2.83$

$\text{Avg}=[(1+2+2+3+4)+(2+2+2+3+4+4)]/11=2.64$

最坏：标识符loop/while需要做3次比较

平均：

$\text{Avg}_{成功}=(1+2+2+3+3)/5=2.2$

$\text{Avg}_{不成功}=(2+2+3+3+3+3)/6=2.67$

$\text{Avg}=[(1+2+2+3+3)+(2+2+3+3+3+3)]/11=2.45$

5.4 最优二分检索树

- 1. 问题的描述
  - 二分检索树的检索：性能特征
  - 通过以上分析看出：对同一标识符集合而言，不同形态的二分检索树的性能是不同的。
    - 什么样的二分检索树性能最好？
    - 如何衡量二分检索树的性能？
    - 何谓最优二分检索树？

5.4 最优二分检索树

■ 1. 问题的描述

- 二分检索树的检索: 最优二分检索树问题
  - 设给定的标识符集合是$\{a_1, a_2, \ldots, a_n\}$，并假定$a_1<a_2< \ldots < a_n$。
  - 设P(i)是对a_i检索的概率，即成功检索概率。
    ($1\le i\le n$，内部结点检索概率)
  - Q(i)是一种不成功检索情况下的概率，正被检索的标识符X恰好满足：$a_i < X<a_{i+1}$，$0\le i\le n$ (设$a_0=-\infty$, $a_{n+1}=+\infty$)时的概率。
    ($0\le i\le n$，外部结点检索概率)

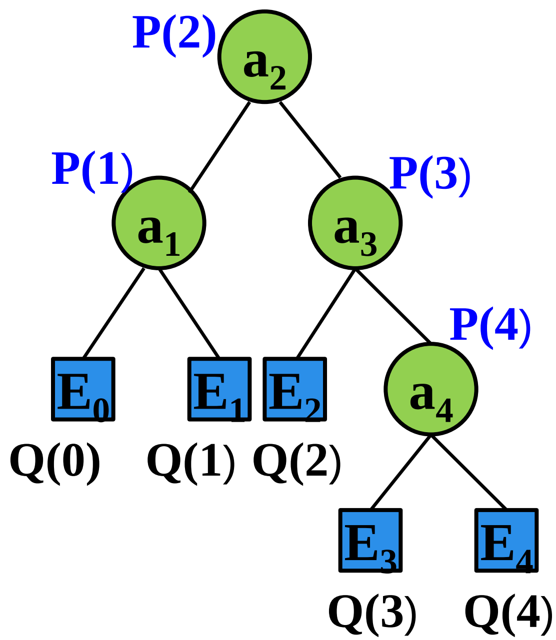
中国科学院大学
University of Chinese Academy of Sciences

5.4 最优二分检索树

■ 1. 问题的描述

□二分检索树的检索: 最优二分检索树问题

➢ 例



$a_0(-\infty)<a_1<a_2<\ldots<a_n<a_{n+1}(+\infty)$

5.4 最优二分检索树

■ 1. 问题的描述

- □二分检索树的检索: 最优二分检索树问题
  - ➢则 $\sum_{1\le i\le n} P(i)$表示所有成功检索的概率
  - ➢ $\sum_{0\le i\le n} Q(i)$ 表示所有不成功检索的概率
  - ➢考虑所有可能的成功和不成功检索情况，共2n+1种可能的情况，有

$$\sum_{1\le i\le n} P(i) + \sum_{0\le i\le n} Q(i) = 1$$

5.4 最优二分检索树

- **1. 问题的描述**
  - 二分检索树的检索: 二分检索树的预期成本
    - 根据期望值计算方法，算法对于所有可能的成功检索情况和不成功检索情况，“加权”平均比较次数，即预期成本为：
    - 平均检索成本 = Σ每种情况出现的概率×该情况下所需的比较次数
      平均检索成本的构成：成功检索成本＋不成功检索成本

5.4 最优二分检索树

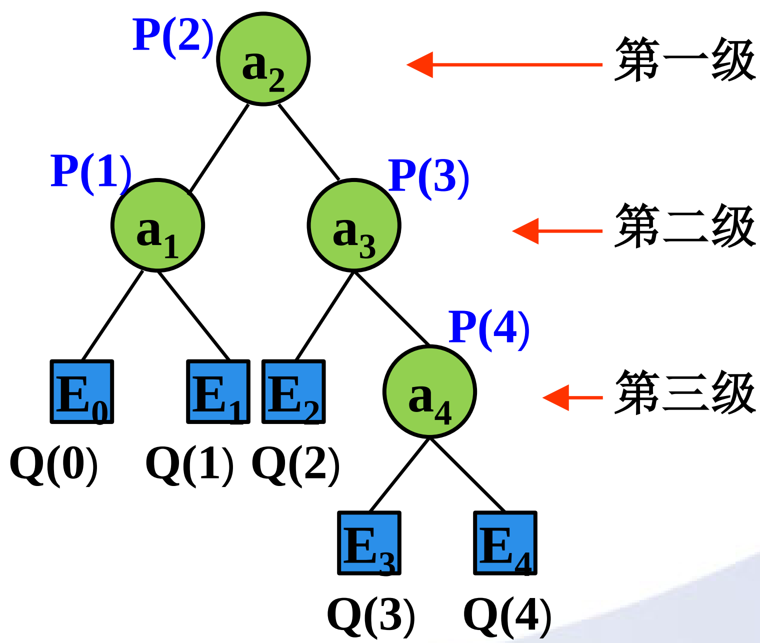
- **1. 问题的描述**
  - 二分检索树的检索: 二分检索树的预期成本
  - 成功检索:
    - 在m级内结点终止的成功检索，需要做m次比较运算(基于二分检索树结构)。
    - P(i)为a_i的检索概率，level(a_i) = 结点a_i 的级数 (1≤i≤n)。
    - 任意结点a_i的成功检索的成本为:

$$P(i)*level(a_i);$$

5.4 最优二分检索树

■ 1. 问题的描述

- 二分检索树的检索：二分检索树的预期成本
- 成功检索：



5.4 最优二分检索树

■ 1. 问题的描述

- 二分检索树的检索：二分检索树的预期成本
- 不成功检索：
  - ➢不成功检索的等价类：每种不成功检索情况定义了一个等价类，共有n+1个等价类$E_i(0\le i\le n)$。其中，

    $E_0=\{X|\ X<a_1\}$

    $E_i=\{X|\ a_i<X<a_{i+1}(1\le i<n)\}$

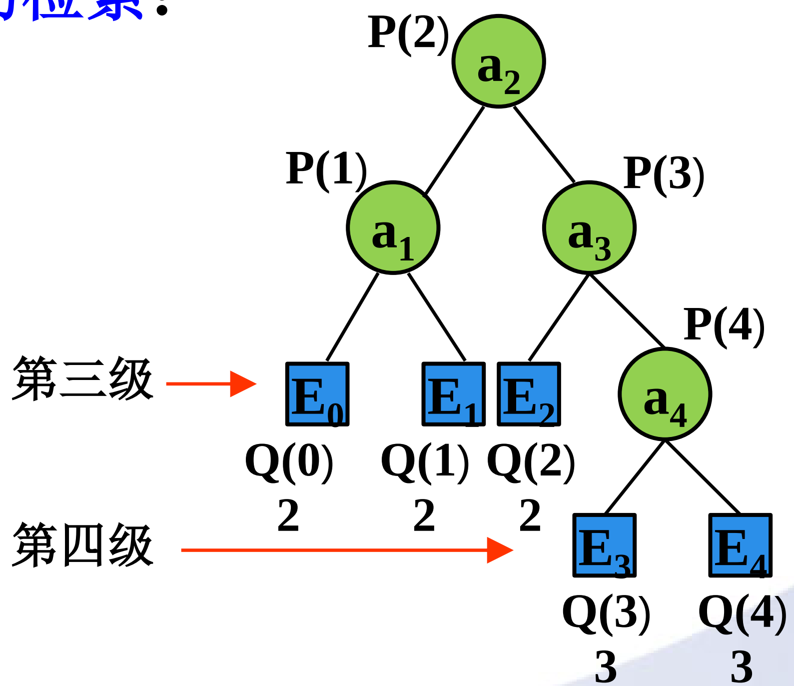
    $E_n=\{X|\ X>a_n\}$
  - ➢若E_i处于m级，则算法需作m-1次迭代。Q(i)为E_i的检索概率，level(E_i) = 结点E_i 的级数(0≤i≤n)，则，外部结点E_i的不成功检索的成本为：

    Q(i)*(level(E_i)-1)

5.4 最优二分检索树

■ 1. 问题的描述

- □二分检索树的检索：二分检索树的预期成本
- □不成功检索：



5.4 最优二分检索树

■ 1. 问题的描述

- 二分检索树的检索：二分检索树的预期成本
- 预期总的成本公式表示如下：

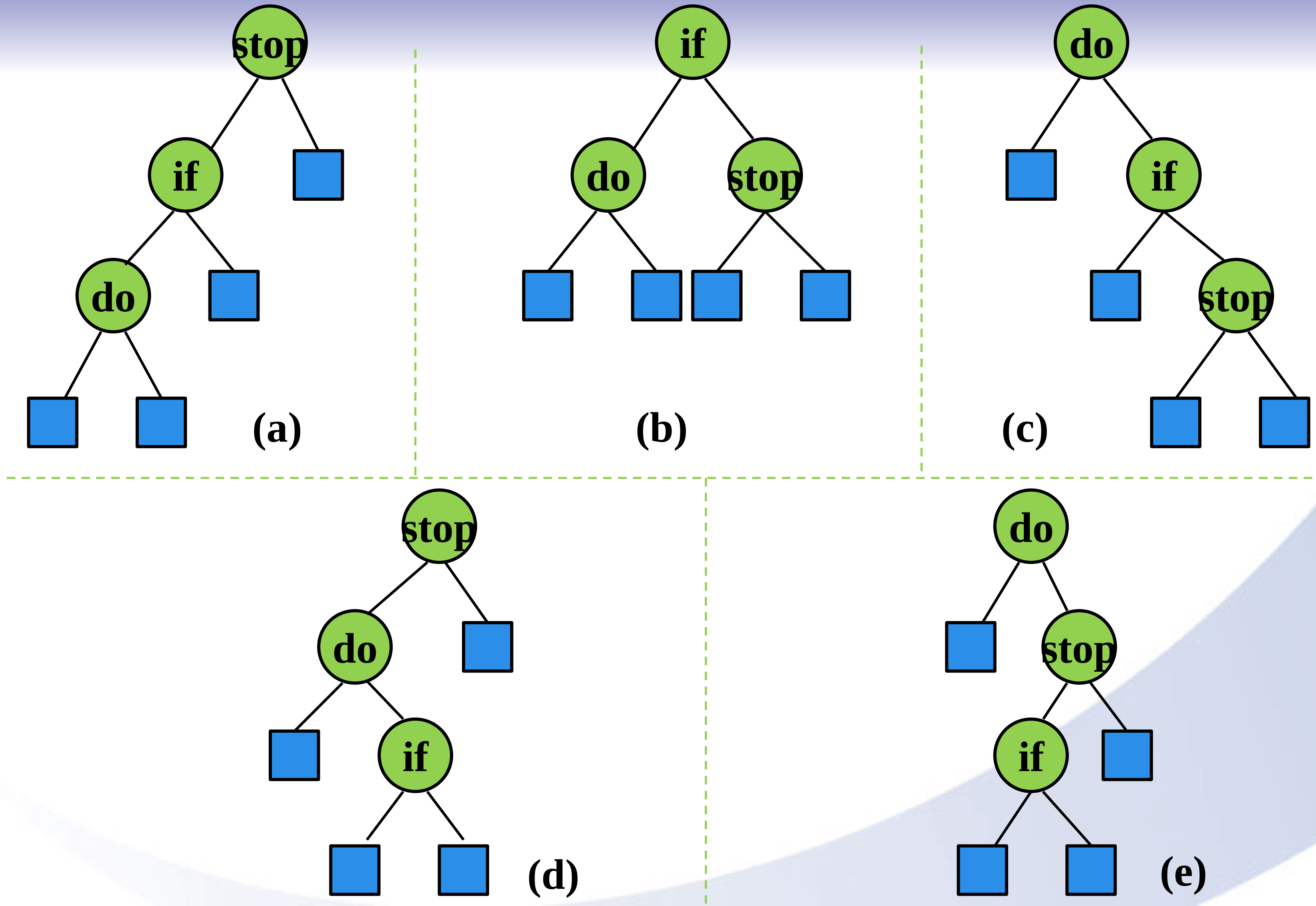
$$\mathrm{COST}(T)=\sum_{1\le i\le n}\mathrm{P(i)}*level(a_i)+\sum_{0\le i\le n}\mathrm{Q(i)}*(level(E_i)-1)$$

- 最优二分检索树问题：
  - ➢求一棵预期成本最小的二分检索树

例5.9 标识符集合(a_1, a_2, a_3) = (do, if, stop)。可能的二分检索树如下所示。

●成功检索：3种

●不成功情况：4种


stop
if
do
(a)
if
do
stop
(b)
do
if
stop
(c)
stop
do
if
(d)
do
stop
if
(e)

5.4 最优二分检索树

■ 1. 问题的描述

- ◘二分检索树的检索: 最优二分检索树问题
  - ➢求一棵使得预期成本最小的二分检索树

1) 等概率：P(i)=Q(i)=1/7

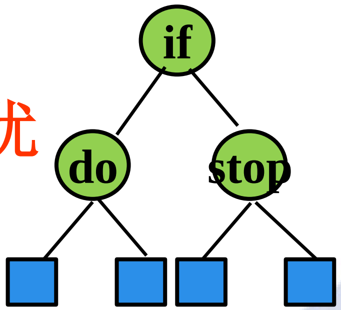
cost(a) = 15/7

cost(b) = 13/7 ⟶ 最优

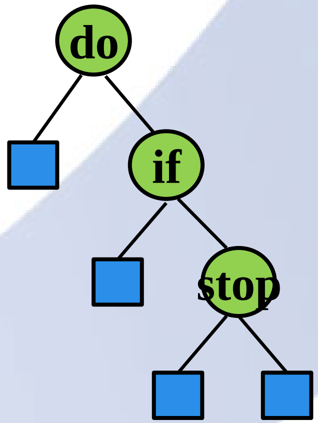
cost(c) = 15/7

cost(d) = 15/7

cost(e) = 15/7



(b)



(c)

5.4 最优二分检索树

■ 1. 问题的描述

□二分检索树的检索：最优二分检索树问题

➢求一棵使得预期成本最小的二分检索树

2)不等概率：

P(1)=0.5 P(2)=0.1 P(3)=0.05

Q(0)=0.15 Q(1)=0.1 Q(2)=0.05 Q(3)=0.05

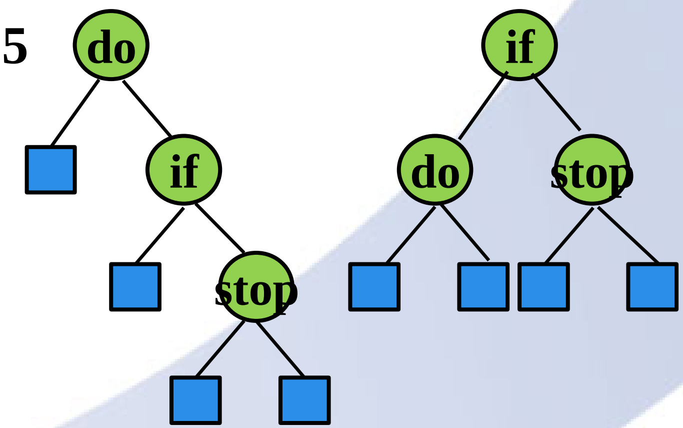
cost(a) = 2.65

cost(b) = 1.9

cost(c) = 1.5 → 最优

cost(d) = 2.15

cost(e) = 1.6



(c) (b)

cost(c) = 0.5*1+0.1*2+0.05*3+
0.15*1+0.1*2+0.05*3+0.05*3=1.5

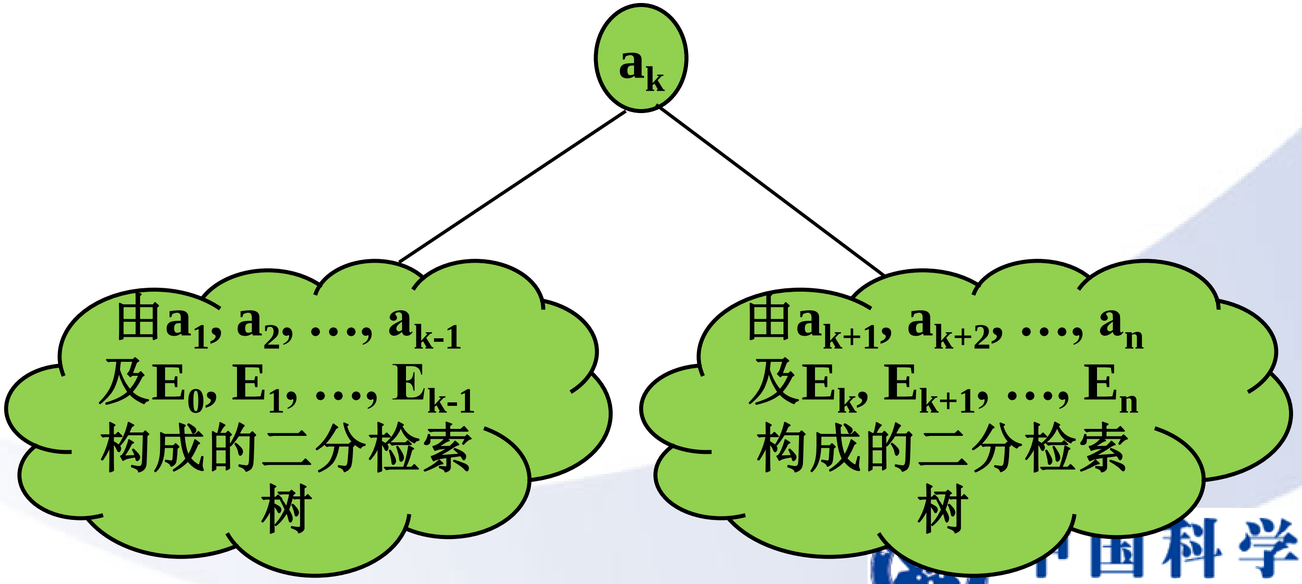
cost(b) = 0.5*2+0.1*1+0.05*2+
0.15*2+0.1*2+0.05*2+0.05*2=1.9

5.4 最优二分检索树

- **1. 问题的描述**
  - 二分检索树的检索：最优二分检索树问题
    - 对于给定的标识符集合，已知成功与不成功检索各种情况的检索概率，什么形态的二分检索树才是最优二分检索树？
    - 枚举法：列举各种形态的二分检索树，分析比较，找出最优的。
    - 动态规划策略：把构造二分检索树的过程看成一系列决策的结果。
      - ✓决策什么？

5.4 最优二分检索树

- 2. 多阶段决策过程
  - 把构造二分检索树的过程看成一系列决策的结果。
  - 决策的策略：决策树根，如果$\{a_1, a_2, \ldots, a_n\}$存在一棵二分检索树，$a_k$是该树的根，则内结点$a_1, a_2, \ldots, a_{k-1}$和外部结点$E_0, E_1, \ldots, E_{k-1}$将位于根$a_k$的左子树中，而其余的结点将位于右子树中。



5.4 最优二分检索树

■ 2. 多阶段决策过程

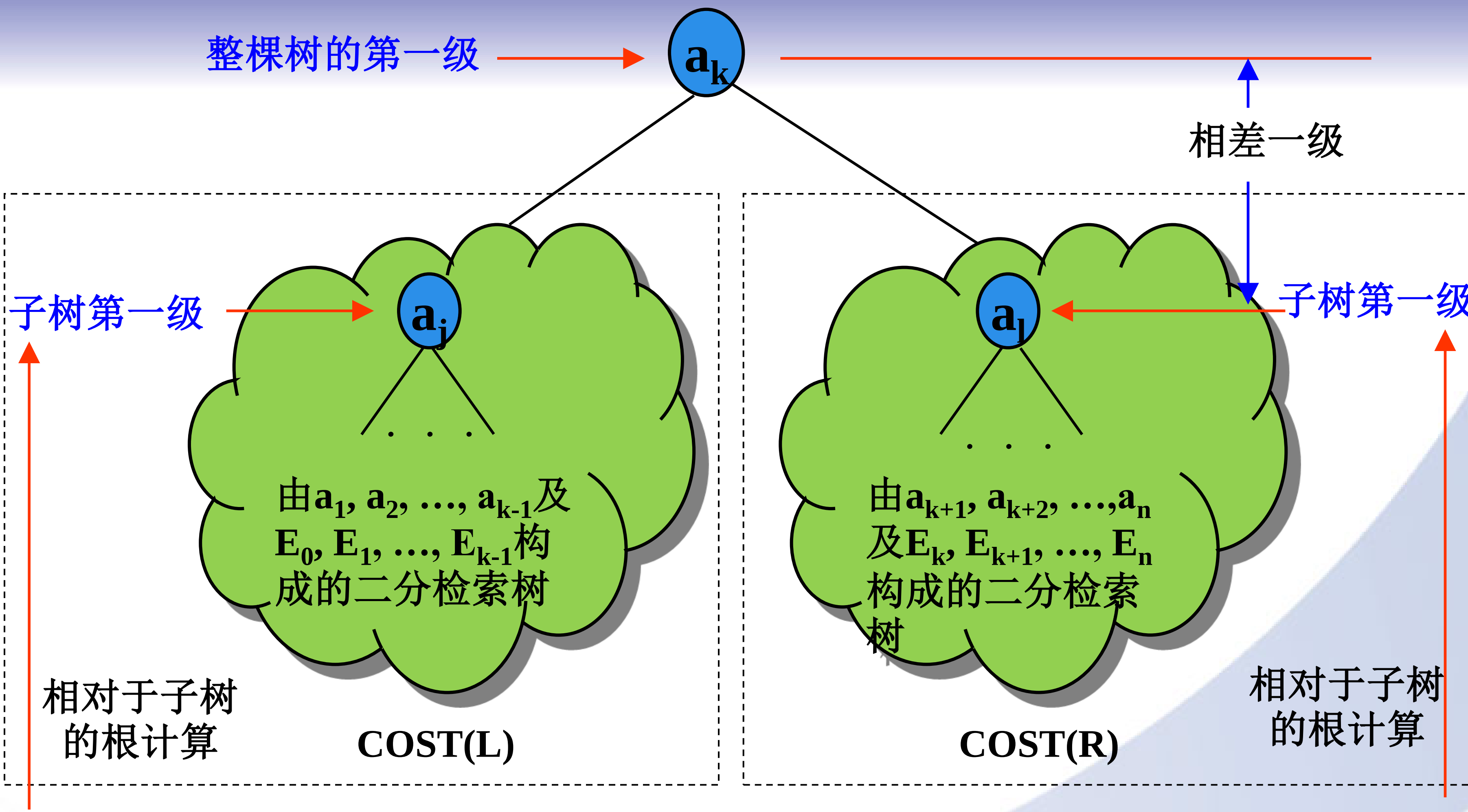
□定义：

$$COST(L)=\sum_{1\le i<k}P(i)*level(a_i)+\sum_{0\le i<k}Q(i)*(level(E_i)-1)$$

$$COST(R)=\sum_{k<i\le n}P(i)*level(a_i)+\sum_{k\le i\le n}Q(i)*(level(E_i)-1)$$

□含义：

- 左、右子树的预期成本——左、右子树的根处于第一级
- 左、右子树中所有结点的级数相对于子树的根测定，而相对于原树的根少1



$$COST(L)=\sum_{1\le i<k}P(i)*level(a_i)+\sum_{0\le i<k}Q(i)*(level(E_i)-1)\qquad COST(R)=\sum_{k<i\le n}P(i)*level(a_i)+\sum_{k\le i\le n}Q(i)*(level(E_i)-1)$$

若：$\text{level}_T(X)$为结点X在树T中的级数，level(X)为X在T的左/右子树中的级数，则：$\text{level}_T(X) = \text{level}(X)+1$

$$\text{COST}(T) = \sum_{1\le i\le n} \text{P(i)} * level_T(a_i) + \sum_{0\le i\le n} \text{Q(i)} * (level_T(E_i) - 1)$$

$$= P(k) + \sum_{1\le i\le k-1} \text{P(i)} * (level(a_i) + 1) + \sum_{0\le i\le k-1} \text{Q(i)} * ((level\ (E_i) + 1) - 1)$$

$$+ \sum_{k+1\le i\le n} \text{P(i)} * (level(a_i) + 1) + \sum_{k\le i\le n} \text{Q(i)} * ((level\ (E_i) + 1) - 1)$$

$$= \text{P(k)} + \sum_{1\le i\le k-1} \text{P(i)} * level(a_i) + \sum_{0\le i\le k-1} \text{Q(i)} * (level(E_i) - 1)$$

$$+ \sum_{1\le i\le k-1} \text{P(i)} + \sum_{0\le i\le k-1} \text{Q(i)} \quad \Rightarrow \quad \text{Q(0)} + \sum_{1\le i\le k-1} (\text{P(i)} + \text{Q(i)})$$

$$+ \sum_{k+1\le i\le n} \text{P(i)} * level(a_i) + \sum_{k\le i\le n} \text{Q(i)} * (level(E_i) - 1)$$

$$+ \sum_{k+1\le i\le n} \text{P(i)} + \sum_{k\le i\le n} \text{Q(i)} \quad \Rightarrow \quad \text{Q(k)} + \sum_{k+1\le i\le n} (\text{P(i)} + \text{Q(i)})$$

COST(L)

COST(R)

记：$W(i,j)=Q(i)+\sum_{i+1}^{j}(Q(l)+P(l))$

则有：

左子树“成本差额”

$$W(0,k-1)=\sum_{1\le i\le k-1}P(i)+\sum_{0\le i\le k-1}Q(i)=Q(0)+\sum_{1\le i\le k-1}(P(i)+Q(i))$$

$$W(k,n)=\sum_{k+1\le i\le n}P(i)+\sum_{k\le i\le n}Q(i)=Q(k)+\sum_{k+1\le i\le n}(P(i)+Q(i))$$

右子树“成本差额”

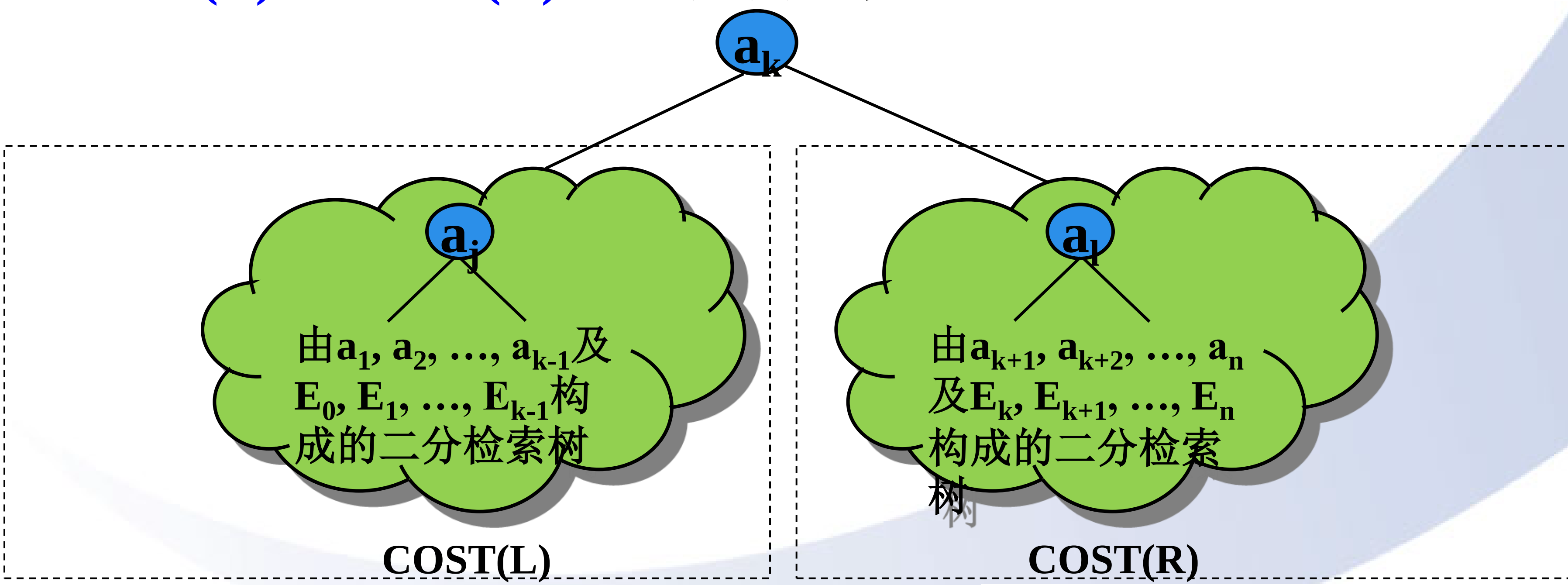
则，原二分检索树的预期成本可表示为：

$$\begin{aligned}\mathrm{COST}(T)&=P(k)+\mathrm{COST}(L)+W(0,k-1)\\&\quad+\mathrm{COST}(R)+W(k,n)\\&=\mathrm{COST}(L)+\mathrm{COST}(R)+W(0,n)\end{aligned}$$

最优二分检索树：COST(T)有最小值

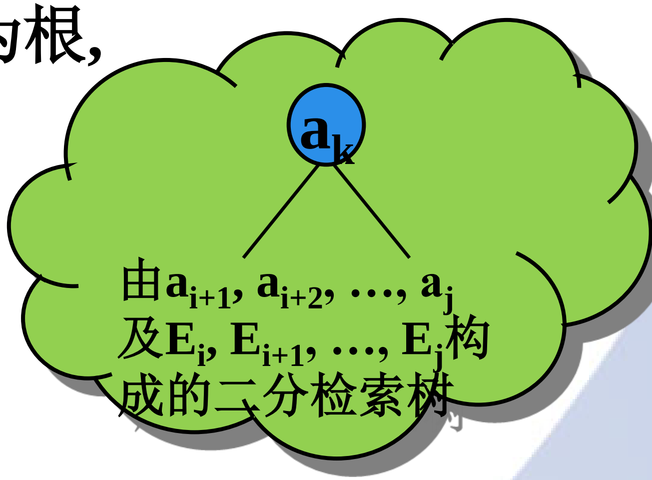
最优性原理成立

若以a_k为根的二分检索树T是关于集合$\{a_1, a_2, \ldots, a_n\}$的一棵最优二分检索树，L、R分别是$a_k$的左子树和右子树，L包含结点$a_1, a_2, \ldots, a_{k-1}$和$E_0, E_1, \ldots, E_{k-1}$、R包含结点及$a_{k+1}, a_{k+2}, \ldots, a_n$和$E_k, E_{k+1}, \ldots, E_n$，则L和R必是相应集合的最优二分检索树。**COST(L)**和**COST(R)**是其预期成本。



中国科学院大学
University of Chinese Academy of Sciences

记由$a_{i+1}, a_{i+2}, \ldots, a_j$和$E_i, E_{i+1}, \ldots, E_j$构成的最优二分检索树的预期成本为**C(i, j)**，设二分检索树T以a_k为根,



则有

$$COST(L) = C(0, k-1)$$

$$COST(R) = C(k, n)$$

而，

$$C(0,n) = \min_{1\le k\le n}\{C(0,k-1)+C(k,n)+\underbrace{P(k)+W(0,k-1)+W(k,n)}_{W(0,n)}\}$$

对任何C(i, j)有，

$$C(i,j) = \min_{i<k\le j}\{C(i,k-1)+C(k,j)+\underline{P(k)+W(i,k-1)+W(k,j)}\}$$

$$= \min_{i<k\le j}\{C(i,k-1)+C(k,j)\}+\underline{W(i,j)}$$

5.4 最优二分检索树

- **2. 多阶段决策过程**
  - 向前递推过程：
    - 首先计算所有$j-i = 1$的$C(i, j)$，
    - 然后依次计算$j-i = 2, 3, \ldots, n$的$C(i, j)$，
    - $C(0, n)$ = 最优二分检索树的成本。

初始值

$$\begin{cases} C(i, i) = 0 \\ W(i, i) = Q(i), \quad 0 \le i \le n \end{cases}$$

5.4 最优二分检索树

- **2. 多阶段决策过程**
  - 最优二分检索树的构造
    - 在计算C(i, j)的过程中，记下使之取得最小值的k值，即树T_{ij}的根，记为R(i, j)。
    - 依据R(0, n), …, 推导树的形态
    - 例5.10

例　设n=4, 且(a_1, a_2, a_3, a_4)=(do　if　read　while)。

设P[1:4] = (3, 3, 1, 1),

Q[0:4] = (2, 3, 1, 1, 1)　(概率值“扩大”了16倍)

构造最优二分检索树

$$W(i,j) = Q(i) + \sum_{i<s\le j}(P(s)+Q(s)) = P(j)+Q(j)+W(i,j\text{-}1)$$

$$C(i,j) = \min_{i<k\le j}\{C(i,k-1)+C(k,j)+W(i,j)\}$$

初始　W(i, i)=Q(i)

C(i, i)=0

R(i, i)=0

且有，W(i, j)=P(j)+Q(j)+W(i, j-1)

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1					
2					
3					
4					

P[1:4] = (3, 3, 1, 1),

Q[0:4] = (2, 3, 1, 1, 1)

W(i, j), C(i, j), R(i, j)

j-i=0，第0行时，为空树

W(0,0)=Q(0)=2
C(0,0)=0
R(0,0)=0

W(1,1)=Q(1)=3
C(1,1)=0
R(1,1)=0

W(2,2)=Q(2)=1
C(2,2)=0
R(2,2)=0

W(3,3)=Q(3)=1
C(3,3)=0
R(3,3)=0

W(4,4)=Q(4)=1
C(4,4)=0
R(4,4)=0

W(i, j)=P(j)+Q(j)+W(i, j-1)

$$C(i,j)=\min_{i<k\le j}\{C(i,k-1)+C(k,j)+W(i,j)\}$$

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2					
3					
4					

P[1:4] = (3, 3, 1, 1),

Q[0:4] = (2, 3, 1, 1, 1)

W(i, j), C(i, j), R(i, j)

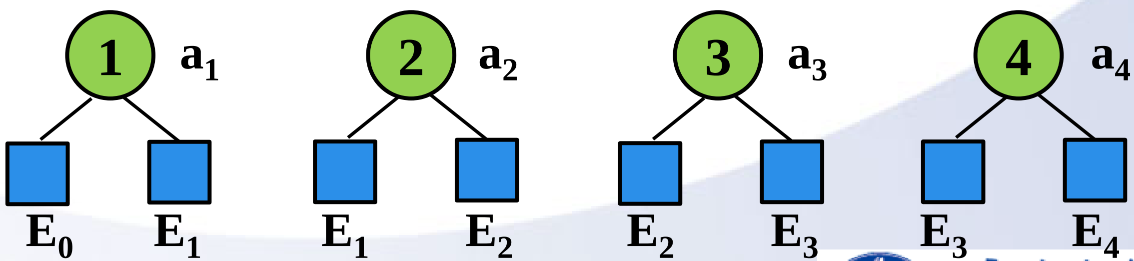
j-i=1，第1行时

W(0,1)= P(1)+Q(1)+W(0,0) = 3+3+2 = 8
C(0,1)= W(0,1)+min{C(0,0)+C(1,1)} = 8
R(0,1)=1 k=1

W(1,2)=P(2)+Q(2)+W(1,1) = 7
C(1,2)=W(1,2)+min{C(1,1)+C(2,2)} = 7
R(1,2)=2 k=2

W(2,3)=P(3)+Q(3)+W(2,2) = 3
C(2,3)=W(2,3)+min{C(2,2)+C(3,3)} = 3
R(2,3)=3 k=3

W(3,4)=P(4)+Q(4)+W(3,3) = 3
C(3,4)=W(3,4)+min{C(3,3)+C(4,4)} = 3
R(3,4) = 4 k=4



W(i, j)=P(j)+Q(j)+W(i, j-1)

$$C(i,j)=\min_{i<k\le j}\{C(i,k-1)+C(k,j)+W(i,j)\}$$

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2	12,19,1	9,12,2	5,8,3		
3					
4					

P[1:4] = (3, 3, 1, 1),

Q[0:4] = (2, 3, 1, 1, 1)

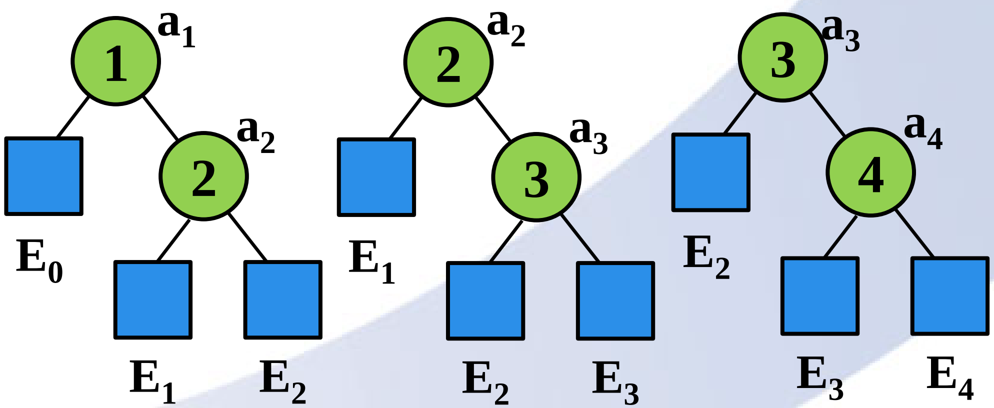
W(i, j), C(i, j), R(i, j)

j-i=2，第2行时

W(0,2)= P(2)+Q(2)+W(0,1) = 3+1+8 = 12
C(0,2)=W(0,2)+min{C(0,0)+C(1,2),C(0,1)
+C(2,2) }=12+min{0+7,8+0}=19
R(0,2)=1

W(1,3)= P(3)+Q(3)+W(1,2) = 1+1+7 = 9
C(1,3)=W(1,3)+min{C(1,1)+C(2,3),C(1,2)
+C(3,3) }=9+min{0+3,7+0}=12
R(1,3)=2

W(2,4)= P(4)+Q(4)+W(2,3) = 1+1+3 = 5
C(2,4)=W(2,4)+min{C(2,2)+C(3,4),C(2,3)
+C(4,4) }=5+min{0+3,3+0}=8
R(2,4)=3



W(i, j)=P(j)+Q(j)+W(i, j-1)

$$C(i,j)=\min_{i<k\le j}\{C(i,k-1)+C(k,j)+W(i,j)\}$$

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2	12,19,1	9,12,2	5,8,3		
3	14,25,2	11,19,2			
4					

P[1:4] = (3, 3, 1, 1),

Q[0:4] = (2, 3, 1, 1, 1)

W(i, j), C(i, j), R(i, j)

j-i=3，第3行时

W(0,3)=P(3)+Q(3)+W(0,2) = 14

C(0,3)=W(0,3)+min{C(0,0)+C(1,3),C(0,1)+C(2,3), C(0,2)+C(3,3)}

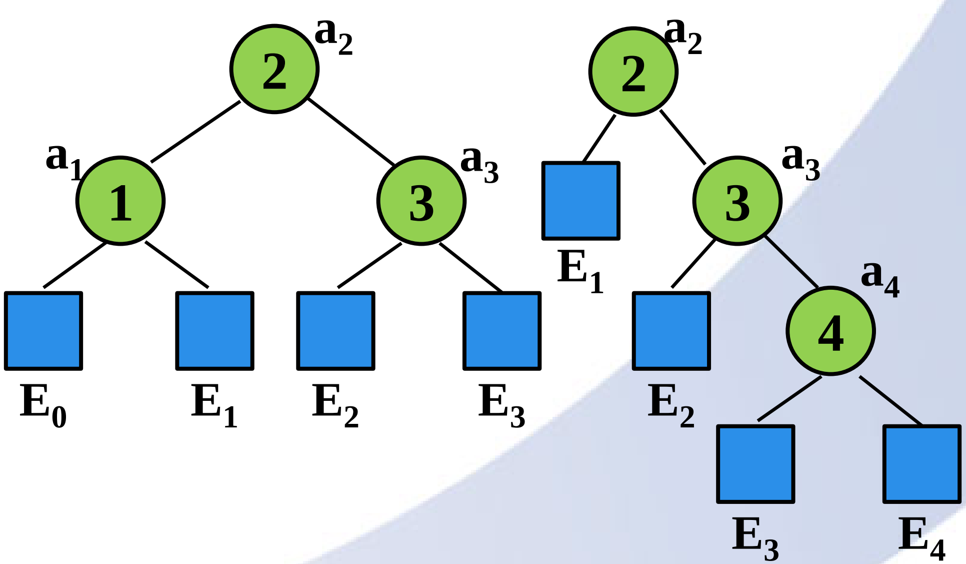
=14 +min{0+12, 8+3,19+0}=25

R(0,3) = 2

W(1,4)=P(4)+Q(4)+W(1,3) = 11

C(1,4)=W(1,4)+min{C(1,1)+C(2,4), C(1,2)+ C(3,4), C(1,3)+C(4,4)} = 19

R(1,4) = 2



W(i, j)=P(j)+Q(j)+W(i, j-1)

$$C(i,j)=\min_{i<k\le j}\{C(i,k-1)+C(k,j)+W(i,j)\}$$

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2	12,19,1	9,12,2	5,8,3		
3	14,25,2	11,19,2			
4	16,32,2				

P[1:4] = (3, 3, 1, 1),

Q[0:4] = (2, 3, 1, 1, 1)

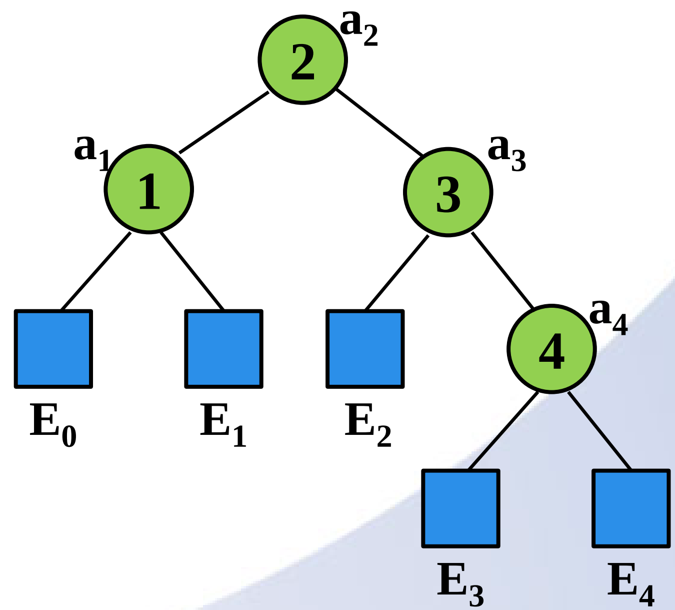
W(i, j), C(i, j), R(i, j)

j-i=4，第4行时

W(0,4)= P(4)+Q(4)+W(0,3) = 1+1+14 = 16

C(0,4)=W(0,4)+min{C(0,0)+C(1,4),C(0,1)+C(2,4),C(0,2)+C(3,4),C(0,3)+C(4,4) }=16+min{0+19, 8+8,19+3, 25+0}=32

R(0,4)=2



W(i, j)=P(j)+Q(j)+W(i, j-1)

$$C(i,j)=\min_{i<k\le j}\{C(i,k-1)+C(k,j)+W(i,j)\}$$

C(i, j), W(i,j), R(i, j)计算结果

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2	12,19,1	9,12,2	5,8,3		
3	14,25,2	11,19,2			
4	16,32,2				

W(i, j), C(i, j), R(i, j)

则，C(0, 4)=32

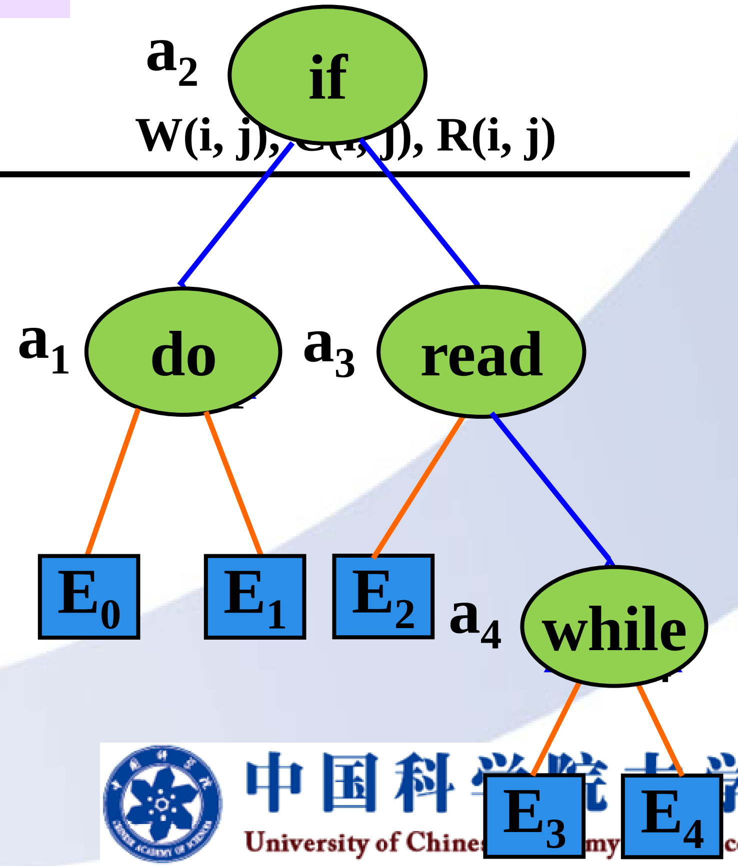
二分检索树：

T_{04}=2 =>T_{01}(左子树)，T_{24}(右子树)

T_{01}=1 =>T_{00}(左子树)，T_{11}(右子树)

T_{24}=3 =>T_{22}(左子树)，T_{34}(右子树)

T_{34}=4 =>T_{33}(左子树)，T_{44}(右子树)



j-i=0阶段的计算：

$$\begin{cases} W(0, 0) = Q(0) = 2 \\ C(0, 0) = 0 \\ R(0, 0) = 0 \end{cases}$$

$$\begin{cases} W(1, 1)=Q(1) = 3 \\ C(1, 1) = 0 \\ R(1, 1) = 0 \end{cases}$$

$$\begin{cases} W(2, 2)=Q(2) = 1 \\ C(2, 2)= 0 \\ R(2, 2) = 0 \end{cases}$$

$$\begin{cases} W(3, 3)= Q(3) = 1 \\ C(3, 3)= 0 \\ R(3, 3) = 0 \end{cases}$$

$$\begin{cases} W(4, 4)= Q(4) = 1 \\ C(4, 4)= 0 \\ R(4, 5) = 0 \end{cases}$$

$$W(i,j) = P(j) + Q(j) + W(i, j\text{-}1)$$
$$C(i,j) = \min_{i<k\le j}\{C(i,k-1) + C(k,j)\} + W(i,j)$$

j-i=1阶段的计算:

$$\left\{\begin{array}{l} W(0,1)=P(1)+Q(1)+W(0,0) = 3+3+2 = 8 \\ C(0,1)=W(0,1)+\min\{C(0,0)+C(1,1)\} = 8 \\ R(0,1) = 1 \end{array}\right.$$

$$\left\{\begin{array}{l} W(1,2)=P(2)+Q(2)+W(1,1) = 7 \\ C(1,2)=W(1,2)+\min\{C(1,1)+C(2,2)\} = 7 \\ R(1,2) = 2 \end{array}\right.$$

$$\left\{\begin{array}{l} W(2,3)=P(3)+Q(3)+W(2,2) = 3 \\ C(2,3)=W(2,3)+\min\{C(2,2)+C(3,3)\} = 3 \\ R(2,3) = 3 \end{array}\right.$$

$$\left\{\begin{array}{l} W(3,4)=P(4)+Q(4)+W(3,3) = 3 \\ C(3,4)=W(3,4)+\min\{C(3,3)+C(4,4)\} = 3 \\ R(3,4) = 4 \end{array}\right.$$

仅有一个内结点

$$W(i,j) = P(j) + Q(j) + W(i, j-1)$$
$$C(i,j) = \min_{i<k\le j}\{C(i,k-1) + C(k,j)\} + W(i,j)$$

j-i=2阶段的计算：

W(0, 2) = P(2)+Q(2)+W(0, 1) = 12

C(0, 2) = W(0, 2)+min{C(0, 0)+C(1, 2), C(0, 1)+C(2, 2)}

= 12 + min{0+7, 8+0}=19

R(0, 2) = 1

W(1, 3) = P(3)+Q(3)+W(1, 2) = 9

C(1, 3) = W(1, 3)+min{C(1, 1)+C(2, 3), C(1, 2)+C(3, 3)} = 12

R(1, 3) = 2

W(2, 4) = P(4)+Q(4)+W(2, 3) = 5

C(2, 4) = W(2, 4)+min{C(2, 2)+C(3, 4), C(2, 3)+C(4, 4)} = 8

R(2, 4) = 3

$$W(i,j) = P(j) + Q(j) + W(i, j-1)$$
$$C(i,j) = \min_{i<k\le j}\{C(i,k-1) + C(k,j)\} + W(i,j)$$

j-i=3阶段的计算:

W(0, 3) = P(3)+Q(3)+W(0, 2) = 14

C(0, 3) = W(0, 3)+min{C(0, 0)+C(1, 3), C(0, 1)+C(2, 3),

C(0, 2)+C(3, 3)}

= 14 + min{0+12, 8+3, 19+0} = 25

R(0, 2) = 2

W(1, 4) = P(4)+Q(4)+W(1, 3) = 11

C(1, 4) = W(1, 4)+min{C(1, 1)+C(2, 4), C(1, 2)+C(3, 4) ,

C(1, 3)+C(4, 4)}

= 19

R(1, 4) = 2

中国科学院大学
University of Chinese Academy of Sciences

$$W(i,j) = P(j) + Q(j) + W(i,j\text{-}1)$$

$$C(i,j) = \min_{i<k\le j}\{C(i,k-1)+C(k,j)\} + W(i,j)$$

j-i=4阶段的计算：

W(0, 4) = P(4)+Q(4)+W(0, 3) = 16

C(0, 4) = W(0, 4)+min{C(0, 0)+C(1, 4), C(0, 1)+C(2, 4),

C(0, 2)+C(3, 4), C(0, 3)+C(4, 4)}

= 16 + min{0+19, 8+8, 19+3, 25+0}

= 32

R(0,4) = 2

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2	12,19,1	9,12,2	5,8,3		
3	14,25,2	11,19,2			
4	16,32,2			W(i, j), C(i, j), R(i, j)	

计算时间

- 当j-i=1时,计算了n个C(i, j)

  j-i=2时,计算了n-1个C(i, j)

  ……

  j-i=m时,计算了n-m+1个C(i, j)

- C(i, j)的计算：O (m)

  每个C(i, j)要求找出m个量中的最小值。

- 则，n-m+1个C(i, j)的计算时间：O $(nm-m^2)$
- 对所有可能的m，总计算时间为：$\sum_{1\le m\le n}(nm-m^2)=O(n^3)$

j-i \ i	0	1	2	3	4
0	2,0,0	3,0,0	1,0,0	1,0,0	1,0,0
1	8,8,1	7,7,2	3,3,3	3,3,4	
2	12,19,1	9,12,2	5,8,3		
3	14,25,2	11,19,2			
4	16,32,2			W(i, j), C(i, j), R(i, j)	

计算时间

- **D.E. Knuth**得出这样的结论：使**C(i, j)**取最小值的**k**可以在区间$R(i,\ j-1) \le k \le R(i+1,\ j)$中求出，而不必在$i \le k \le j$这个相对较大的范围内求出。
- 一种改进措施：最优的**k**∈**[R(i, j-1), R(i+1, j)]**
- 改进后的算法为**OBST**算法
- **OBST**算法的计算时间:$O(n^2)$

4. 算法描述

//给定n个标识符$a_1<a_2<\ldots<a_n$。已知成功检索的概率P(i), 不成功检索概率Q(i)， $0\le i\le n$。算法对于标识符$a_{i+1}, \ldots, a_j$计算最优二分检索树T_{ij}的成本C(i, j)、根R(i, j)、权W(i, j) //

```
procedure OBST(P, Q, n)
   real P(1:n), Q(1:n), C(0:n, 0:n), W(0:n, 0:n); integer R(0:n,
   for i←0 to n-1 do
        (W(i, i), R(i, i), C(i, i))←(Q(i), 0, 0)  //置初值//
        (W(i, i+1), R(i, i+1), C(i, i+1))←(Q(i)+Q(i+1)+P(i+1), i+1, Q(i)+Q(i+1)+P(i+1))
   repeat                                  //含有一个结点的最优树//
   (W(n, n), R(n, n), C(n, n))←(Q(n), 0, 0)
   for m←2 to n do        //找有m个结点的最优树//
       for i←0 to n-m do
           j←i+m
           W(i, j) ←W(i, j-1)+P(j)+Q(j)
           k←区间[R(i,j-1), R(i+1,j)]中使{C(i,l-1)+C(l,j)}取最小值的l值 //Knuth结论
           C(i, j) ←W(i, j)+C(i, k-1)+C(k, j)
           R(i, j)←k
       repeat
   repeat
end OBST
```

初始化，
j–i=0, j–i=1

相邻m个元素的
最优二分检索树的成本

树根为a_k

作业-课后练习18

- 问题描述
  - 设n=4， 且(a_1, a_2, a_3, a_4)=(do, if, stop, then)， 设P(1:4) = (3, 3, 1, 1)， Q(0:4) = (1, 3, 2, 1, 1)) (概率值“扩大”了16倍)
  - 计算W(i, j)，R(i, j)和C(i, j)，并给出最优二分检索树
- 要求
  - 作业提交到课程网站上
  - Word文档即可

End