

問10 最適二分探索木に関する次の記述を読んで、設問 1～3 に答えよ。

(平成 11 年 PE 午後 II 問 1)

識別子がキーワードか否かを判定するプログラムを二分探索木を使って実現したい。判定しようとする識別子について探索木をたどり、一致するものが見つければ、この識別子をキーワードと判定し、一致するものがなければキーワードではないと判定する。同じキーワードの集合に対しても、どのキーワードに対応する節を根にするかは一意に決まらないので、探索木の作り方は多数ある。キーワードとキーワード以外の識別子の出現確率が分かっているときには、その確率で識別子を検索したときに比較回数を最少にするような二分探索木が考えられる。このような木を最適二分探索木と呼ぶ。

ここでは、より一般的に木のコストを定義し、コストを最小にすることによって最適二分探索木を作るプログラムについて考察する。コストの計算で使う重みとして、識別子の出現確率を使えば、コストの値は、検索時に行う識別子の比較回数となる。

キーワード  $K_r$  ( $r=1, 2, \dots, n$ ) と各キーワードの重み  $\alpha_r$  ( $r=1, 2, \dots, n$ )、キーワード以外の識別子の重み  $\beta_r$  ( $r=0, 1, 2, \dots, n$ ) が与えられたとき、キーワードを検索する二分探索木のコストを次のように定義する。

$$\text{コスト} = \sum_{r=1}^n \alpha_r \times p_r + \sum_{r=0}^n \beta_r \times q_r$$

ここで、 $p_r$  は根からキーワード  $K_r$  が格納されている節までの経路長、 $q_r$  はキーワード以外を表す葉までの経路長を表し、経路長は“根から節又は葉に至るときに通る枝の数+1”で定義する。キーワード以外の識別子の重み  $\beta_r$  は次のように与える。 $\beta_r$  ( $r=1, 2, \dots, n-1$ ) は、 $K_r$  より大きく  $K_{r+1}$  より小さいすべての識別子の重みの総和を、 $\beta_0$  は  $K_1$  より小さいすべての識別子の重みの総和を、 $\beta_n$  は  $K_n$  より大きいすべての識別子の重みの総和を与える。2 個のキーワードの例を図 1 に示す。図で○は節を、□は葉を表す。この例では、 $p_1=1, p_2=2, q_0=2, q_1=3, q_2=3$  となる。

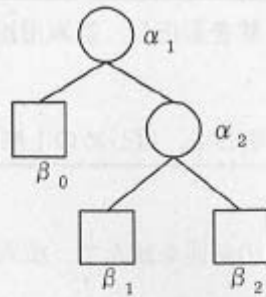


図1 二つの節からなる木

簡単な例で実際に最適二分探索木を求めてみる。二つのキーワード begin と end を含む識別子が、表1に示す出現頻度で現れるとする。このとき、 $\alpha_r$  ( $r=1, 2$ )と $\beta_r$  ( $r=0, 1, 2$ )の値は表2のようになる。 $\beta_1$ は識別子cとdの出現確率の和、 $\beta_2$ は識別子xとyの出現確率の和となる。

表1 単語の出現頻度

識別子	出現頻度	計	出現確率
a	20	20	0.1
begin	40	40	0.2
c	20	60	0.3
d	40		
end	60	60	0.3
x	10	20	0.1
y	10		

表2  $\alpha_r$ と $\beta_r$ の値

r	$K_r$	$\alpha_r$	$\beta_r$
0	-	-	0.1
1	begin	0.2	0.3
2	end	0.3	0.1

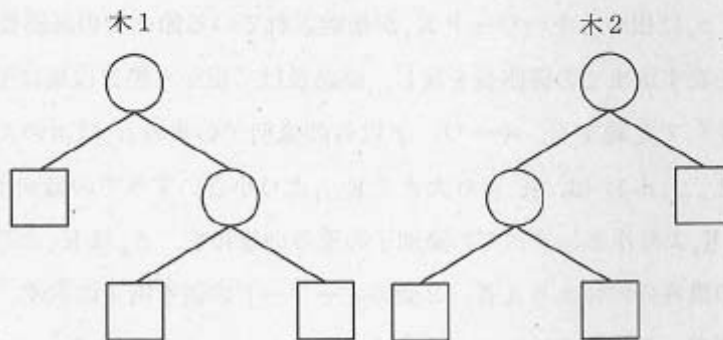


図2 キーワードが二つのときに作ることのできる木

節が二つの木は図2の2通りしか存在しない。表2の重みを使って、二つの木のコストを計算してみる。

$$\text{木1のコスト} = (0.2 \times 1 + 0.3 \times 2) + (0.1 \times 2 + 0.3 \times 3 + 0.1 \times 3) = 2.2$$

$$\text{木2のコスト} = (0.2 \times 2 + 0.3 \times 1) + (0.1 \times 3 + 0.3 \times 3 + 0.1 \times 2) = 2.1$$

この場合は木2のコストが最小となり、木2が最適二分探索木となる。

最適二分探索木を作るには“最適二分探索木の任意の部分木は最適二分探索木である”という事実に着目する。この事実は次のようにして示すことができる。仮に最適二分探索木で最適でない部分木をもつものがあつたと仮定する。最適でない部分木を再構成して最適にすれば、元の木よりコストが小さい木が作れることになる。これは元の木が最適二分探索木であるという仮定に反することになる。

この事実を利用して最適二分探索木を求めることができる。n個のキーワードが与えられたとき、n個のキーワードの一つを木の根として、左右の部分木の最適二分探索木のコストを同様な方法で求め、全体のコストを求める。これをすべてのキーワードに対して計算し、最小値をとる木が、求める最適二分探索木になる。

キーワード  $K_i, \dots, K_j$  で構成される最適二分探索木のコストを求める処理は、次のように書くことができる。

$i \leq j$  のとき

全重みの和  $\times n \rightarrow \min$

k を i から j まで繰り返す

$K_i, \dots, K_{k-1}$  の最適二分探索木のコスト  $\rightarrow LC$

$K_{k+1}, \dots, K_j$  の最適二分探索木のコスト  $\rightarrow RC$

LC, RC を利用して計算した、 $K_k$  を根とする木のコスト  $\rightarrow \text{cost}$

if (cost < min) cost  $\rightarrow$  min

min が最小コスト

$i > j$  のとき

$\beta_j$  が最小コスト

LCの計算では範囲を  $i, \dots, k-1$  とし, RCの計算では範囲を  $k+1, \dots, j$  として上の処理を再帰的に繰り返す。処理を繰り返すたびに, 部分木の節数は減少し, 最終的には範囲を表す  $i$  と  $j$  の大小関係が逆転する。これは部分木の節数が0の場合に相当し, コストは直ちに求められる。少し考察すると, 節数0の木に対しては, その位置に対応する葉の重みを使えばよいことが分かる。

このまま再帰関数で定義して, トップダウンに求めることもできるが, 必ずしも効率はよくない。そこで, 上の考えを基に, 部分木の最適コストを求める部分を工夫することで, ボトムアップに計算する方法を考える。

$K_i, \dots, K_j$  からなる最適二分探索木のコストを  $C[i-1, j]$  にしておくことにする。 $K_k$  を根としたときの最小コストは, 左右の部分木を最適二分探索木としたときに求まる。左部分木を最適二分探索木としたときのコスト  $C[i-1, k-1]$ , 右部分木を最適二分探索木としたときのコスト  $C[k, j]$  が求まっていれば, これを使って  $K_k$  を根としたときの木の最小コストを求めることができる。節数の少ない木から順に最小コストを計算すれば, 部分木は元の木より節数が少ないので, 左右の部分木の最適二分探索木のコストは計算が終わっており, 元の木の利用ができる。

**設問1** キーワードが3個のとき, 作られる可能性のある二分探索木をすべて, 図2 にならって図示せよ。

**設問2** キーワードが3個の場合を考える。 $\alpha_r$  と  $\beta_r$  が表3のように与えられているとき, 最適二分探索木のコストを求め, そのときの木の形を図示せよ。同じコストの木が二つ以上あるときは, すべてを示すこと。

表3 重み

r	$\alpha_r$	$\beta_r$
0	—	0.2
1	0.2	0.1
2	0.1	0.1
3	0.2	0.1

設問3 キーワード  $K_i, K_{i+1}, \dots, K_j$  からなる木で、 $K_k$  を根とする木のコストを最小にするには、この木の左右の部分木を最適二分探索木とすればよい。左右の部分木を最適二分探索木としたときのコストを利用して、全体のコストを求める方法を考える。

左右の部分木を最適二分探索木としたときに、部分木における各節、葉への経路長をそれぞれ  $p'_r (r=i, i+1, \dots, j)$ ,  $q'_r (r=i-1, i, i+1, \dots, j)$  とする。また、木全体の重みの和を  $W$ , 左右の最適二分探索木のコストをそれぞれ  $LC, RC$  とする。式を簡単にするために  $p'_k=0$  とおく。

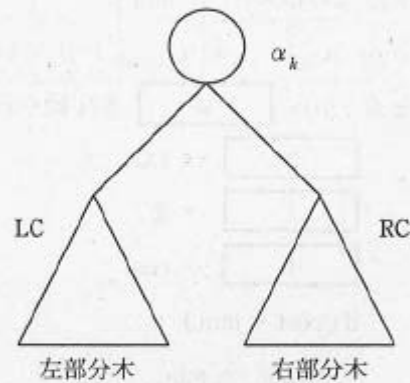


図3  $K_k$  を根とする木

図3を参考にするに、 $K_k$  を根とする木の最小のコストは次の式で計算できることが分かる。次の式中の  $\boxed{a} \sim \boxed{e}$  を埋めよ。

$$\begin{aligned}
 \text{木のコスト} &= \sum_{r=i}^j \alpha_r \times \boxed{a} + \sum_{r=i-1}^j \beta_r \times \boxed{b} \\
 &= \sum_{r=i}^j \alpha_r \times \boxed{c} + \sum_{r=i-1}^j \beta_r \times \boxed{d} + \left( \sum_{r=i}^j \alpha_r + \sum_{r=i-1}^j \beta_r \right) \\
 &= \boxed{e}
 \end{aligned}$$

設問4 キーワード  $K_1, K_2, \dots, K_n$  に対する最適二分探索木を求める次のプログラム中の  $f$  ~  $k$  に入れる適切な式を答えよ。

$r$  を 0 から  $n$  まで繰り返す

$\beta_r \rightarrow C[r, r]$

$m$  を 1 から  $n$  まで繰り返す

$i$  を 1 から  $f$  まで繰り返す

全重みの和  $\times n \rightarrow \min$

$\alpha_i + \alpha_{i+1} + \dots + \alpha_{i+m-1} + \beta_{i-1} + \beta_i + \dots + \beta_{i+m-1} \rightarrow W$

$k$  を  $i$  から  $g$  まで繰り返す

$h \rightarrow LC$

$i \rightarrow RC$

$j \rightarrow \text{cost}$

if (cost < min)

cost  $\rightarrow$  min

$k \rightarrow C[i-1, i+m-1]$

$C$  を印刷

設問5 設問4のプログラムで完成した最適二分探索木のコストは配列  $C$  のどこに求まるか答えよ。

設問6 表3のデータに対し、設問4のプログラムに従って計算したとき、最終的に  $C$  はどのような値を取るか。配列の空欄を埋めよ。

$C[i, j]$ :

$i \backslash j$	0	1	2	3
0				
1	-			
2	-	-		
3	-	-	-	