Добавил(а) Гісь Ігор Володимирович
26.12.12 10:13 -

Дерево (рекурсивне опрацювання дерев)

Те, що зображено на малюнку, називається деревом вибору усіх можливих варіантів двійкових комбінацій, що починаються на 1.

Кожен рівень цього дерева - це рівень розгалуження, яки відповідає вибору однієї цифри комбінації. Усього рівнів N (у нашому прикладі- 3), по кількості цифр.

Тепер уже ясно, що наприклад, дерево вибору всіляких двійкових комбінацій довжини 3, що починаються на 1 (тобто в иду
1::), буде виглядати так:

□□□□□

□ 1 □..... 1-ая цифра

□□□□□

□□□□□□□□□□□□

□□□□□ □□□□□

□ 0 □.....□ 1 □..... 2-ая цифра

□□□□□ □□□□□

□□□□□□□□□□□□□□□

□□□□□□□□□□ □□□□□□□□□□

□ 0 □□ 1 □.□ 0 □□ 1 □..... 3-я цифра

□□□□□□□□□□ □□□□□□□□□□

100 101 110 111

Верхній квадрат, що містить цифру 1, називається **коренем** цього дерева, а всі квадрати найнищого рівня, що далі вже не розгалужуються, називаються **листками** дерева. Для повної аналогії з деревом варто було б перевернути наш малюнок, але так вже повелося,

що
математичні дерева малюють униз,
- просто це зручніше.

Уведемо ще два терміни: **гілка** повного дерева - це шлях, що з'єднує його корінь і який-небудь з листів, **вузол** - це довільний елемент на гілці

Давайте порахуємо число листів. Воно збігається з числом усіх двійкових комбінацій. Як виходить кожна
з комбінацій?
Потрібно просто перебрати (обійти)
усі гілки
повного дерева праворуч.

Цей процес і називається повним **обходом дерева**.

Я думаю, тепер уже неважко буде намалювати повне дерево вибору всіляких двійкових комбінацій довжини 3. Воно має
пустий корінь, уведений просто для зручності.
Ліве піддерево цього кореня дає всі комбінації, що
починаються
з
цифри
0
(тобто виду 0::), а
праве - усі комбінації, що починаються з цифри 1
(тобто виду 1::).

□□□□□

□ □

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ □□□□□

□ 0 □.....□ 1 □..... 1

□□□□□ □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ □□□□□ □□□□□ □□□□□

□ 0 □.....□ 1 □.....□ 0 □.....□ 1 □..... 2

□□□□□ □□□□□ □□□□□ □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□ □□□□□□□□□□□ □□□□□□□□□□□ □□□□□□□□□□□

□ 0 □□ 1 □.□ 0 □□ 1 □.□ 0 □□ 1 □.□ 0 □□ 1 □..... 3

□□□□□□□□□□□ □□□□□□□□□□□ □□□□□□□□□□□ □□□□□□□□□□□

000 001 010 011 100 101 110 111

Видно, що число листів цього дерева дорівнює 8 чи 2 , тобто збігається з числом усіляких двійкових комбінацій.

Задача 1

Вивести всілякі N-значні двійкові комбінації L--і (тобто всілякі послідовності нулів і одиниць довжини N).

Отже, ми повинні вивести всі двійкові комбінації, кожна довжиною N. Наприклад, для N = 4 ми повинні вивести наступні 16 комбінацій:

0000 0001 0010 0011 0100 0101 0110 0111

1000 1001 1010 1011 1100 1101 1110 1111

Я думаю, уже ясна загальна схема алгоритму обходу дерева вибору:

ЯКЩО знаходимося в листі

ТО вивести всі цифри гілки, від кореня до даного листа

ІНАКШЕ ввійти в піддерево, що починається з 0;

ввійти в піддерево, що починається з 1

ВСЕ

Приведений нижче алгоритм запам'ятовує обрані цифри в таблиці c[1:N]. На кожному рівні дерева
ми вибираємо одну цифру c[i],
тоді у момент, коли ми досягнемо листа (це рівень N), всі елементи

таблиці будуть заповнені цифрами однієї комбінації (чи цифрами на
гілці, від кореня до листа).

Процедура Розгалуження (ЦІЛЕ і, іс) має два параметри.

Перший, і, визначає розмірність піддерева, у яке ми входимо (вона дорівнює N-i+1). Друг
ий, іс
,
- це цифра, що стоїть у корені
цього піддерева.

А тепер весь алгоритм цілком.

```
ПОЧ   'розгалуження порожнього кореня:

Розгалуження( 1, 0) ' входимо в ліве піддерево з коренем 0

Розгалуження( 1, 1) ' входимо в праве піддерево з коренем 1

КІН
```

```
'Процедура розгалуження при виборі цифри двійкової комбінації 'з номером i+1

 АЛГ Розгалуження (ЦІЛЕ i, ic )

ПОЧ

c[i] := ic

 ЯКЩО i = N 'досягли листа дерева?

 ТО ВИСНОВОК( c[1:N] ) ' -так, виведення комбінації

ІНАКШЕ ' -ні, продовжуємо розгалуження

: Розгалуження( i+1, 0) ' входимо в ліве піддерево з коренем

: Розгалуження( i+1, 1) ' входимо в праве піддерево з коренем

LBCE

 КІН
```

Дерево вибору, що ми будували, аналізуючи алгоритм одержання комбінацій з 0 і 1, називається двійковим, чи бінарним, тому що на кожному
рівні відбувалося розгалуження на дві гілки.
У довільному дереві
таких гілок може бути більше.

```
uses crt;

const n=5;

var c:array[1..n] of integer;

procedure pr(inn:integer;ic:integer);

var i:integer;

begin

c[inn]:=ic;

if inn=n then begin for i:=1 to n do write(c[i]);
```

```
writeln;



end



else begin pr(inn+1,0);pr(inn+1,1);



end;



end;









begin



clrscr;



pr(1,0);



pr(1,1);



end.
```

Задача 2

Видати всі N-значні числа в системі з основою K+1, K <= 9 L--і (тобто всі комбінації з цифр від 0 до K, довжини N).

Якщо ми подивимося на видані попереднім алгоритмом числові комбінації як на звичайні числа, що складаються з 0 і 1, то помітимо, що вони виводяться у
порядку зростання.
Якщо ж ми дзеркально розгорнемо дерево (ніби
порядку зростання будемо перебирати гілки ліворуч), то тим самим розв’яжемо дану задачу:

```
uses crt;

const n=3;

 k=2;

var c:array[1..n] of integer;

i:integer;
```

```
procedure pr(inn:integer;ic:integer);

var j,ii:integer;

begin

c[inn]:=ic;

if inn=n then begin for j:=1 to n do write(c[j]);

write('---');

end

else for ii:=0 to k do pr(inn+1,ii);

end;

begin
```

```
clrscr;


for i:=0 to k do pr(1,i);


end.
```

Задача 3

Вивести всілякі N-значні двійкові комбінації L--і в порядку спадання утворених ними чисел.

Дзеркальне відображення дерева змінює його в кожній гілці таким чином:

□□□□□ □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ □□□□□ -> □□□□□ □□□□□

□ 0 □ □ 1 □ □ 1 □ □ 0 □

L-T-i L-T-i L-T-i L-T-i

Тому всі зміни алгоритму будуть стосуватися тільки порядку обходу піддерева: спочатку ми будемо входити в піддерево з коренем 1, а уже потім у піддерево з коренем 0, а не навпаки, як раніше.

В алгоритмі це приведе до перестановки місцями двох наступних

(одна за одною) команд виклику процедури Розгалуження().

Тепер наш алгоритм, наприклад, для N=4, видасть комбінації

у наступній послідовності:

1111 1110 1101 1100 1011 1010 1001 1000

0111 0110 0101 0100 0011 0010 0001 0000

Задача 3

Видати всілякі двійкові комбінації в такому порядку: комбінації, які починаються з 0, повинні виводитись в порядку зрост
ання утворених ними чисел, а починаються з 1 - у порядку
їхнього спадання.

Для N=4 послідовність видачі комбінацій повинна бути така:

0000 0001 0010 0011 0100 0101 0110 0111 <- зростає

1111 1110 1101 1100 1011 1010 1001 1000 <- спадає

Подивіться тепер на зображене нижче дерево вибору для N=4, воно схоже на ті дерева для N=3, що ми малювали раніше, але
має особливість: ліве піддерево з рівня 2 відрізняється від правого
порядком вибору 0 і 1.
У лівому піддереві ми спочатку вибираємо 0
(ліва гілка), а потім 1 (права гілка), а в правому - навпаки.

□□□

□ □

□□□

□□

□□□ □□□

□0□ ....□1□ 1

□□□ □□□

□□□□□□□□П□□□□□□□□□□□□□□□□□□□О□□□□□□□□□□□

□□□ □□□ □□□ □□□

□0□ ....□1□ ....□1□ ....□0□ 2

□□□ □□□ □□□ □□□

П П О О

0 ..... 1 ..... 0 ..... 1 ...... 1 ..... 0 ..... 1 ..... 0 3

П П П П О О О О

0 . 1 . 0 . 1 . 0 . 1 . 0 . 1 .. 1 . 0 . 1 . 0 . 1 . 0 . 1 . 0 .. 4

ліве піддерево праве піддерево

Будемо робити обхід цього дерева, перебираючи його гілки праворуч.

При цьому ми й одержимо необхідну послідовність комбінацій:

перша половина кодів буде виводитись у порядку зростання, а друга половина - у порядку убування.

Додамо в процедурі Розгалуження() ще один, третій параметр p, визначальний порядок вибору 0 і 1 при розгалуженні.
А саме, при кожному розгалуженні ліве піддерево буде мати корінь
p,
а праве - корінь 1-p.

Таким чином, параметр p визначає два типи розгалужень, які ми назвемо прямим і зворотнім,

при p = 0: при p = 1:

□□□ □□□

□□П□□ □□З□□

□□□ □□□□□□ □□□

□0□ □1□ □1□ □0□

□□□ □□□□□□□ □□□

(буква в крапці розгалуження вказує на тип розгалуження:

П-прямий, З-зворотний).

Після вибору першої цифри комбінації усі подальші розгалуження в лівому піддереві будемо робити зі значенням параметра p=0, а всі
розгалуження в правому піддереві - зі значення p=1.

```
ЦІЛИЙ N  'довжина двійкової комбінації

ЦІЛИЙ ТАБ c[1:N] 'цифри однієї комбінації

АЛГ Двійкові_комбінації( )
```

ПОЧ 'розгалуження порожнього кореня:

Розгалуження(1, 0, 0) ' входимо в ліве піддерево з коренем 0

 ' типом розгалуження П

Розгалуження(1, 1, 1) ' входимо в праве піддерево з коренем 1

 ' типом розгалуження ПРО

КІН

' Процедура розгалуження при виборі цифри двійкової комбінації з номером i+1

АЛГ Розгалуження(ЦІЛЕ ic, i, p)

ПОЧ

c[i] := ic

```
 ЯКЩО i = N  'досягли листка дерева?

 ТО ВИВЕДЕННЯ( c[1:N] )  ' -так, виведення комбінації

 ІНАКШЕ  ' -ні, продовжуємо розгалуження

  : Розгалуження( i+1, p, p )  ' входимо в ліве піддерево

 : Розгалуження( i+1, 1-p, p )  ' входимо в праве піддерево

 ВСЕ

 КІН
```

```
uses crt;

const n=4;

var c:array[1..n] of integer;

i:integer;
```

```
procedure pr(inn,ic,p:integer);

var j:integer;

begin

c[inn]:=ic;

if inn=n then begin for j:=1 to n do write(c[j]);

writeln;

end

else begin pr(inn+1,p,0);pr(inn+1,1-p,1);end;

end;

begin

clrscr;
```

```
pr(1,0,0);


pr(1,1,1);


end.
```

Коди Грея

Двійковим N-розрядним кодом Грея називається послідовність усіляких N-значних кодів, у якій ко
жна
наступна кодова
комбінація відрізняється від попередньої на 1 тільки в одному довільному
розряді.
Видати N-розрядні коди Грея.

Наприклад, для N=4, кодом Грея є:

0000 0001 0011 0010 0110 0111 0101 0100

1100 1101 1111 1110 1010 1011 1001 1000 .

НОТАТКИ

Особливістю всіх задач, що ми розв’язували дотепер, було незмінність варіантів вибору. Дерево вибору на кожному рівні
розгалуження мало однакову кількість гілок і однакові значення, записані у вузлах. Ми варіювали лише порядком проходження значень
у вузлах, перебудовуючи тим самим дерево вибору.

Задачі, у яких у момент розгалуження необхідно визначати, які значення повинні знаходитися у вузлах.

1. Вивести всілякі N-значні двійкові комбінації,

N

сума цифр яких дорівнює K (K < 2)

2. Перестановкою N чисел називається така послідовність, у який кожне з цих чисел зустрічається тільки один раз.

Видати всі перестановки натуральних чисел чисел від 1 до N (N <= 9).

3. Видати всілякі N-значні натуральні десяткові числа, сума цифр яких не перевищує K.

4. Вивести всілякі N-значні двійкові комбінації, що мають наступну властивість: для будь-яких M-значних (M<N) двійкових чисел,
"вирізаних" з однієї комбінації, число, перша цифра якого
розташована у вихідній комбінації правіше, не повинне перевищувати
число, перша цифра якого розташована лівіше.

Наприклад, для N=5, M=3, допустимими є комбінації

11111 111 111 111 * 1111 111 111 *

11110 111 111 110 * 1110 111 110 *

11101 111 110 101 * 1101 110 101 *

11100 111 110 100 * 1100 110 100 *

11011 110 101 011 * 1011 101 011 *

11010 110 101 010 * 1010 101 010 *

11001 110 100 001 * 1001 100 001 *

11000 110 100 000 * 1000 100 000 *

10111 101 011 111 0111 011 111

10110 101 011 110 0110 011 110

10101 101 010 101 0101 010 101

10100 101 010 100 0100 010 100

10011 100 001 011 0011 001 011

10010 100 001 010 0010 001 010

10001 100 000 001 0001 000 001

10000 100 000 000 * 0000 000 000 *

01111 011 111 111

01110 011 111 110

01101 011 110 101

01100 011 110 100

01011 010 101 011

01010 010 101 010

01001 010 100 001

01000 010 100 000

00111 001 011 111

00110 001 011 110

00101 001 010 101

00100 001 010 100

00011 000 001 011

00010 000 001 010

00001 000 000 001

00000 000 000 000 *