РЕШЕНИЕ ЗАДАЧ С ПОМОЩЬЮ РЕКУРСИВНОГО АЛГОРИТМА В КОМПИЛЯТОРЕ TURBO PASCAL

Кузнецова Екатерина Александровна

ГУО «Козенская средняя школа Мозырского района»

Республика Беларусь, Гомельская область, учитель информатики

Аннотация: В данной статье продемонстрированы решения головоломки «Ханойская башня» и задачи «Числа Фибоначчи» при помощи рекурсивного алгоритма, суть которого обращение к самому себе.

Ключевые слова: рекурсивный алгоритм, Turbo Pascal

«Ханойская башня». Правила головоломки «Ханойская башня» таковы. Имеется доска с тремя колышками. На первом из них нанизано несколько дисков убывающего диаметра (самый большой находится внизу – рис. 1).

Требуется расположить диски в том же порядке на третьем колышке, причём диски разрешается перекладывать только по одному, а класть большой диск на меньший нельзя. Один колышек используется в качестве вспомогательного. Ответим на вопрос – сколько перемещений дисков следует выполнить?

Алгоритм решения головоломки следующий:

1. Переместить верхние n-1 дисков на второй колышек.
2. Нижний диск с первого колышка переместить на третий.
3. Переместить n-1 дисков на третий колышек, используя первый в качестве вспомогательного.
4. Повторять, пока на третьем колышке не будет сформирована новая пирамида.

Исходная задача сводится, таким образом, к двум задачам о переносе n-1 диска и одной задаче о переносе одного диска. Для n-1 требуется одно перемещение. Исходный текст программы для вычисления количества ходов приведён ниже. Количество ходов вычислим с применением рекурсии (функция hanoi)

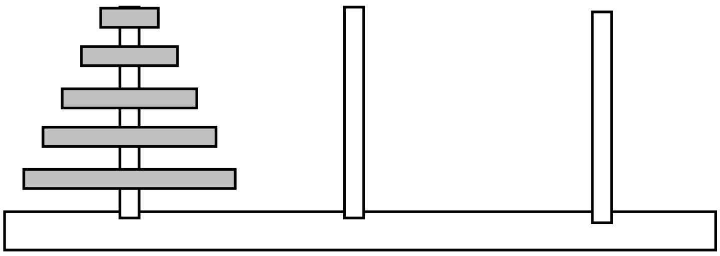

Рисунок 1. Головоломка «Ханойская башня»

Программа «Ханойская башня»:

```
{$S+}
program hanoi_moves;
function hanoi1(n: Word): LongInt;
begin
  if n=1 then hanoi1:=1
  else hanoi1:=2*hanoi1(n-1)+1;
end;
function hanoi2(n: Word): LongInt;
var j: LongInt; k: Word;
begin
  if n=1 then hanoi2:=1
  else
    begin
      j:=1;
      for k:=2 to n do j:=2*j+1;
      hanoi2:=j;
    end;
  end;
  writeln(hanoi1(20));
  writeln(hanoi2(20));
end.
```

«Числа Фибоначчи». Рассмотрим ещё один пример использования рекурсии – вычисление N-ого по счёту числа Фибоначчи. Числа Фибоначчи составляют последовательность, очередной элемент которой вычисляется по двум предыдущим значениям: $F_n=F_{n-1} + F_{n-2}$.

Нулевое и первое значения должны быть заданы, их значения равны единице. Последовательности такого рода применяются, например, в программах генераторах случайных чисел. Вычисление 20-ого числа Фибоначчи реализовано в программе Fibonacci. Впрочем, номер числа можно изменить, задав в описании константы другое значение.

Программа «Числа Фибоначчи»:

```
program Fibonacci;
uses Crt;
const n=20;
function F(n: word): longint;
begin
  if keypressed then
    halt;
  if (n=0) or (n=1) then
    F:=1
  else F:=F(n-1)+F(n-2); {рекурсивный вызов}
end;
function G(n: word): longint;
var x,y,t: longint; k: word;
begin
  x:=1;
  y:=1;
  for k:=2 to n do
  begin
    t:=y;
    y:=x+y;
```

```
    x:=t;
  end;
  G:=y;
end;
begin
  clrscr;
  textcolor(lightgray +128);
  write('Считаю...');
  textcolor(lightgray);
  writeln('—Ждите ответа--');
  writeln;
  writeln('Рекурсивный алгоритм : F(', n, ')= ', F(n));
  writeln;
  writeln('Итерационный алгоритм : F(', n, ')= ', G(n));
  gotoxy(1,1);
  clreol;
  gotoxy(1,7);
  write('Нажмите <Enter>');
end.
```

В этой программе реализованы два метода решения задачи вычисления числа Фибоначчи. Один назовём итерационным методом – он заключается в прямом программировании итерационной формулы для чисел Фибоначчи. В функции G для этого используются три вспомогательные переменные типа LongInt.

Решение с использованием рекурсивных вызовов запрограммировано с помощью функции F. Оператор, вычисляющий её значение, два раза вызывает саму эту функцию. Текст рекурсивной функции короче, лаконичнее итерационной функции

Литература

1. Мальцев А.И. Алгоритмы и рекурсивные функции. – 1965.

2. Ставровский А.Б. Турбо Паскаль 7.0. Учебник. – 2000.