

«Рекурсивные алгоритмы».

Разбор заданий №11 ЕГЭ по информатике и ИКТ



Исламов Ришат Габитович, учитель информатики
МБОУ Сургутский естественно-научный лицей
2016г.



В **2016** г. КИМ ЕГЭ сохранили значительную преемственность с КИМ **2015** г. Основные характеристики работы: количество заданий, сложность заданий, количество первичных баллов и алгоритм перевода первичных баллов в тестовые – остались неизменными. Отличия коснулись только содержания трех заданий первой части, в связи с полным отказом от заданий с выбором ответа. Также в связи с этим изменился порядок следования первых 5 заданий 1 части, остальные задания остались на своих местах.



Таким образом, сохранился реализованный в **2015 г.** подход укрупнения тематики заданий, сведения близких по тематике и сложности заданий в одну позицию. Такими укрупненными были в **2016 г.** позиции 4 (хранение информации в компьютере), 6 (формальное исполнение алгоритмов), 7 (технология вычислений и визуализации данных с помощью электронных таблиц) и 9 (скорость передачи звуковых и графических файлов).

В КИМ ЕГЭ, использовавшихся на экзамене, в части вариантов были задания по одной из указанных в спецификации тем, в другой части – по смежной теме. Это сильно повысило вариативность вариантов, добавив элемент неопределенности.



Общее количество участников экзамена в 2016 г. – 49380 чел, число снижается год от года (в 2015 г. – 50394 чел, 2014 – 53281 чел., 2013 – 54897 чел.), но доля от общего числа участников ЕГЭ более-менее неизменна: чуть выше 7%.

Данные о распределении участников по группам тестовых баллов приведены в табл. Числа, соответствующие диапазонам тестовых баллов, составляют доли в процентах.

Год	Средний тестовый балл	Диапазон тестовых баллов				
		0-20	21-40	41-60	61-80	81-100
2016	56,63	6,89	9	38,11	36,16	9,84
2015	53,99	8,75	11,85	38,57	32,63	8,21
2014	57,79	4,16	8,9	41,93	37,86	7,15

Динамика тестового балла по годам обучения

2013-2014			2014-2015			2015-2016		
РФ	Югра	Сургут	РФ	Югра	Сургут	РФ	Югра	Сургут
57,19	60,6	66,70	53,6	57,91	59,54	53,0	58,9	65,0



Спецификация КИМ ЕГЭ устанавливает три уровня сложности заданий: базовый, повышенный и высокий, при этом для заданий базового уровня примерный интервал выполнения задания предполагается 60-90%; для повышенного уровня результат выполнения должен быть в интервале 40-60%; с заданиями высокого уровня сложности должны справляться менее 40% участников экзамена.



В 2015 г. отмечался крайне низкий результат выполнения задания 11 (25,7% в 2015 г.) по теме «Рекурсивные алгоритмы».

В 2016 г. показатель выполнения этого задания возрос до 36%, но этого по-прежнему недостаточно.

Сегодня для вас я проведу занятие по разбору данного задания, так как формирование представления о рекурсивных вызовах процедур и функций относится к числу важных предметных результатов обучения информатике в средней школе, а само по себе понятие рекурсии является фундаментальным.



Рекурсия — определение, описание, изображение какого-либо объекта или процесса внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя.

Термин «рекурсия» используется в различных специальных областях знаний — от **лингвистики** до **логики**, но наиболее широкое применение находит в **математике** и **информатике**.

В математике **рекурсия** имеет отношение к методу определения функций и числовых рядов: рекурсивно заданная функция определяет своё значение через обращение к себе самой с другими аргументами.



Что нужно знать:

-Рекурсия — это такая организация выполнения работы функции, при которой данная функция вызывает саму себя. Рекурсия может быть прямой и косвенной.

-Рекурсия – это приём, позволяющий свести исходную задачу к одной или нескольким более простым задачам того же типа

-чтобы определить рекурсию, нужно задать:

-условие остановки рекурсии

-рекуррентную формулу

-любую рекурсивную процедуру можно запрограммировать с помощью цикла

-рекурсия позволяет заменить цикл и в некоторых сложных задачах делает решение более понятным, хотя часто менее эффективным



Рекурсия может быть прямой и косвенной.

В случае прямой рекурсии вызов функцией самой себя делается непосредственно в этой же функции

```
procedure F(n: integer);
```

```
begin
```

```
  writeln(n);
```

```
  if n > 1 then begin
```

```
    F(n-1);
```

```
    F(n-3)
```

```
end
```

```
end;
```

```
end;
```



Косвенная рекурсия создаётся за счёт вызова данной функции из какой-либо другой функции, которая сама вызывалась из данной функции.

```
function F(n: integer): integer;
```

```
begin
```

```
  if n > 2 then
```

```
    F := F(n - 1) + G(n - 2)
```

```
  else
```

```
    F := 1;
```

```
end;
```

```
function G(n: integer): integer;
```

```
begin
```

```
  if n > 2 then
```

```
    G := G(n - 1) + F(n - 2)
```

```
  else
```

```
    G := 1;
```

```
end;
```



Существует решение этого задания методом формального исполнения (трассировки) алгоритма, хотя более простым для реализации является решение методом записи рекуррентных соотношений и построения таблицы значений. Низкий показатель выполнения этого задания говорит о том, что понятие рекурсии многими учащимися в процессе обучения так и не было освоено.



Ниже записана рекурсивная функция (процедура) F. Что выведет программа при вызове F(9)? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

(КИМ ЕГЭ 2015 г.)

```
procedure F(n: integer);  
begin  
write(n);  
if n >= 7 then  
begin  
F (n - 3); F (n - 1)  
end end;
```

Сначала необходимо изучить текст программы на одном из языков программирования и понять, что выполняет данная функция. Функция получает на вход одно число n , выводит его на экран, затем при условии, что $n \geq 7$ осуществляет два последовательных вызова $F(n - 3)$ и $F(n - 1)$, что приведет к печати меньших значений n и дальнейшим рекурсивным вызовам.



Выпишем рекуррентное соотношение для общего случая:

$F(n) = n, F(n - 3), F(n - 1)$, при $n \geq 7$;

$F(n) = n$, при $n < 7$.

Далее заполним таблицу, что выведет функция при вызове для разных значений n :

n	Рекуррентное соотношение для $F(n)$	Результат вызова функции $F(n)$
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7, $F(4)$, $F(6)$	746
8	8, $F(5)$, $F(7)$	85746
9	9, $F(6)$, $F(8)$	9685746

Дан рекурсивный алгоритм: (Статград 2016 г)

```
procedure F(n: integer);  
begin  
  writeln(n);  
  if n < 6 then  
    begin F(n+2); F(n*3) end  
end;
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

1. сначала определим рекуррентную формулу; обозначим через F(n) сумму чисел, которая выводится при вызове F(n)

2. при $n \geq 6$ процедура выводит число n и заканчивает работу без рекурсивных вызовов: **$F(n) = n$ при $n \geq 6$**

3. при $n < 6$ процедура выводит число n и дважды вызывает сама себя:

$F(n) = n + F(n+2) + F(3n)$ при $n < 6$

4. в результате вызова F(1) получаем

$$F(1) = 1 + F(3) + F(3); \quad F(3) = 3 + F(5) + F(9) = 3 + F(5) + 9$$

$$F(5) = 5 + F(7) + F(15) = 5 + 7 + 15 = 27$$

используем обратную подстановку:

$$F(3) = 3 + F(5) + 9 = 3 + 27 + 9 = 39 \quad F(1) = 1 + F(3) + F(3)$$

Ответ: 79.



Ниже записан рекурсивный алгоритм F. (Статград 28.09.15)

```
procedure F(n: integer);  
begin  
if n > 0 then  
begin  
F(n - 4); writeln(n); F(n div 3)  
end  
end;
```

Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова **F(9)**?

F(n) = n + F(n-4) + F(n div 3) при n > 0

$$F(1) = 1$$

$$F(2) = 2 + F(2-4) + F(2 \text{ div } 3) = 2$$

$$F(3) = 3 + F(3-4) + F(1) = 4$$

$$F(4) = 4 + F(1) = 5$$

$$F(5) = 5 + F(1) + F(1) = 7$$

$$F(6) = 6 + F(2) + F(2) = 10$$

$$F(7) = 7 + F(3) + F(2) = 13$$

$$F(8) = 8 + F(4) + F(2) = 15$$

$$F(9) = 9 + F(5) + F(3) = 20$$



Ниже записаны две рекурсивные функции (процедуры): F и G.

```
procedure F(n: integer); forward;  
procedure G(n: integer); forward;  
procedure F(n: integer);  
begin  
if n > 0 then G(n - 1);  
end;  
procedure G(n: integer);  
begin  
writeln('*'); if n > 1 then F(n - 2);  
end;
```

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?



В данной задаче используется так называемая косвенная рекурсия, когда функция F вызывает функцию G , а функция G вызывает функцию F .

Изучив текст программы, заметим, что функция $G(n)$ осуществляет вызов $F(n - 2)$, а вызов функции $F(n)$ приводит к вызову $G(n - 1)$.

Таким образом, если функция $G(n)$ вызывает $F(n - 2)$, то она в свою очередь вызывает $G(n - 3)$.

Обозначим через $g(n)$ количество звездочек, которое будет напечатано на экране, если вызвать функцию $G(n)$ из данного задания. Всегда будет напечатана хотя бы одна звездочка, так как функция G начинается с команды вывода одной звездочки, но если $n > 3$, то произойдет рекурсивный вызов $G(n) \wedge F(n - 2) \wedge G(n - 3)$.

При $n < 3$ рекурсивного вызова не будет, например, если $n = 2$, то $G(2)$ вызовет $F(0)$, а та просто завершит работу, не вызывая ничего.



Выпишем рекуррентное соотношение для $g(n)$:

$$g(n) = 1 + g(n - 3), \text{ при } n \geq 3,$$

$$g(n) = 1, \text{ при } n < 3.$$

Заполним таблицу значений функции $g(n)$:

n	Рекуррентное соотношение для $g(n)$	Значение функции $g(n)$
0	1	1
1	1	1
2	1	1
3	$1 + g(3-3)=1+1$	2
4	$1 + g(4-3)=1+1$	2
5	$1 + g(2)=1+1$	2
6	$1 + g(3)=1+2$	3
7	$1 + g(4)$	3
8	$1 + g(5)$	3
9	$1 + g(6)$	4
10	$1 + g(7)$	4

Статград 30.09.2016

Ниже записаны рекурсивные функции F и G.

```
function F(n: integer): integer;  
begin  
  if n > 2 then  
    F := F(n-1) + G(n-2)  
  else  
    F := n;  
  end;  
function G(n: integer): integer;  
begin  
  if n > 2 then  
    G := G(n-1) + F(n-2)  
  else  
    G := 3-n;  
  end;
```

**Чему будет равно значение, вычисленное при выполнении
вызова G(6)?**



$F(n)$ - if $n > 2$ then $F := F(n-1) + G(n-2)$ Else $F := n$;
 $G(N)$ - if $n > 2$ then $G := G(n-1) + F(n-2)$ else $G := 3-n$;

Рекуррентное
соотношение

Значение функции

$$F(1) := 1$$

1

$$G(1) := 3-1=2$$

2

$$F(2) := 2$$

2

$$G(2) := 3-2=1$$

1

$$F(3) := F(2) + G(1)$$

4

$$G(3) := G(2) + F(1)$$

2

$$F(4) := F(3) + G(2)$$

5

$$G(4) := G(3) + F(2)$$

4

$$F(5) := F(4) + G(3)$$

7

$$G(5) := G(4) + F(3)$$

8

$$G(6) := G(5) + F(4)$$

13

Дан рекурсивный алгоритм:

```
procedure F(n: integer);
```

```
begin
```

```
  writeln(n);
```

```
  if n < 6 then begin
```

```
    F(n+2); F(n*3)
```

```
  end
```

```
end;
```

Найдите сумму чисел, которые будут выведены при вызове F(1).

1. Заполним таблицу F(n) при n ≥ 6 (где F(n) = n)

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G(n)						6	7	8	9	10	11	12	13	14	15

2. Заполним таблицу с n = 5 справа налево, используя формулу

F(n) = n + F(n+2) + F(3n); F(5) = 5 + F(7) + F(15) = 5 + 7 + 15 = 27

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G(n)	79	30	39	22	27	6	7	8	9	10	11	12	13	14	15



Дан рекурсивный алгоритм:

```
procedure F(n: integer);
```

```
begin
```

```
  writeln(n);
```

```
  if n < 5 then
```

```
    Begin  F(n + 1);F(n + 3)  end
```

```
end;
```

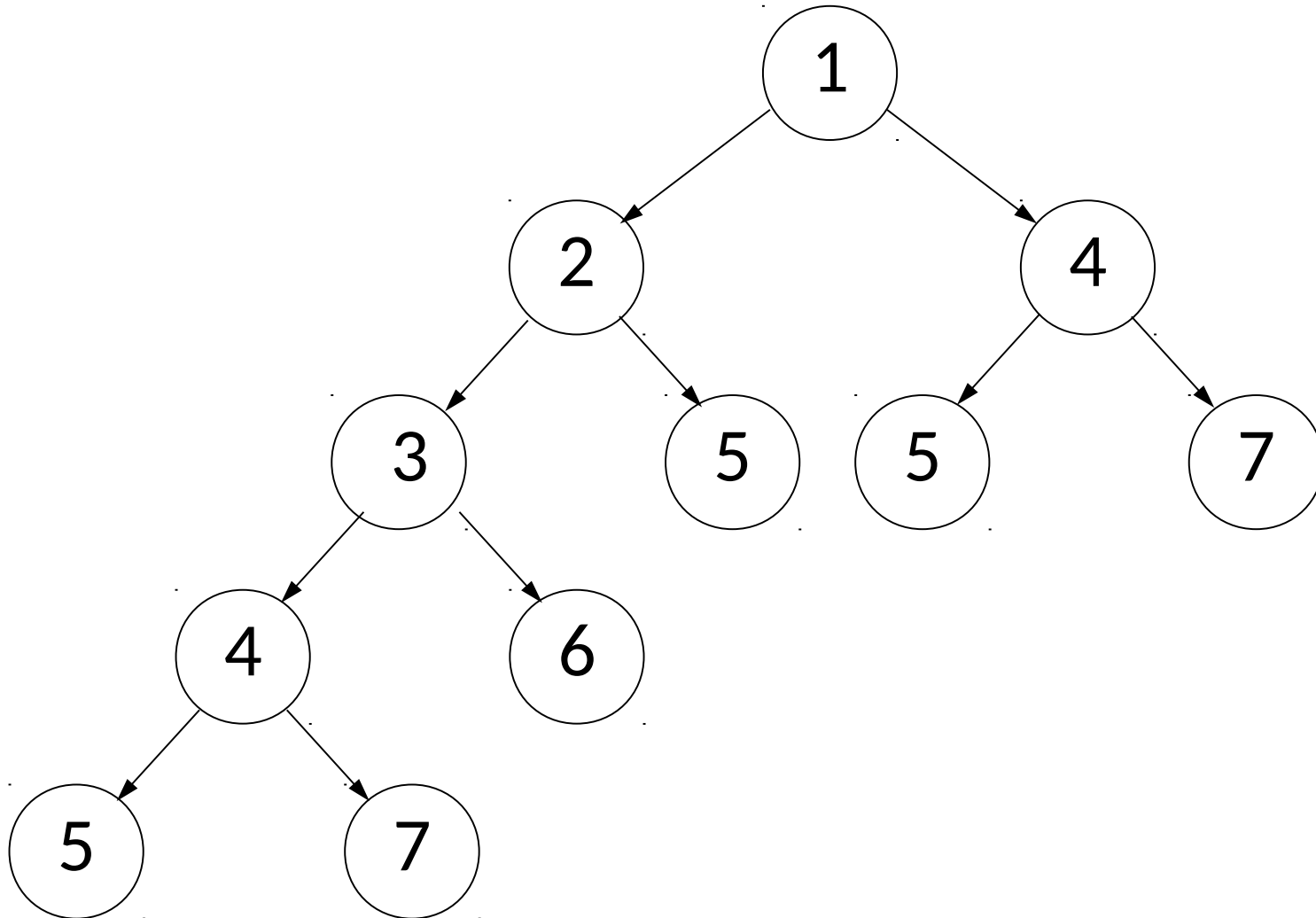
Найдите сумму чисел, которые будут выведены при вызове F(1).

1. Поскольку в начале каждого вызова на экран выводится значение единственного параметра функции, достаточно определить порядок рекурсивных вызовов и сложить значения параметров

2. Поскольку при $n < 5$ выполняется два рекурсивных вызова, решать такую задачу «на бумажке» удобно в виде двоичного дерева (в узлах записаны значения параметров при вызове функции):



n ; $F(n + 1)$; $F(n + 3)$



складывая все эти числа, получаем 49



Спасибо за внимание!

Жду вопросов по электронной почте

Islamov.rishat86@mail.ru

**Успехов при сдаче ЕГЭ по
информатике и ИКТ**

