Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```

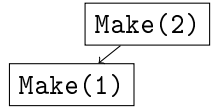
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```

Make(2)

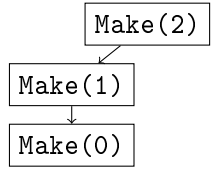
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



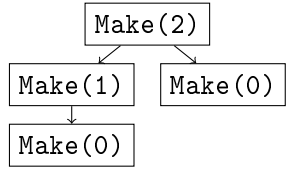
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



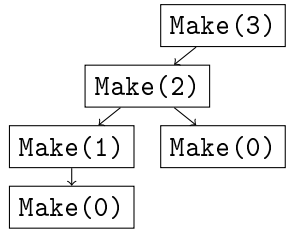
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



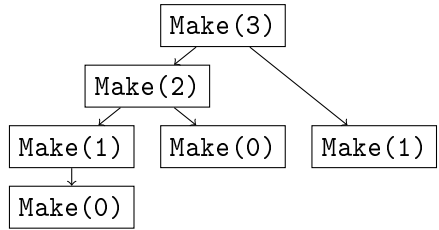
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



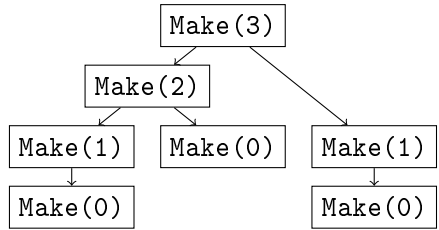
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



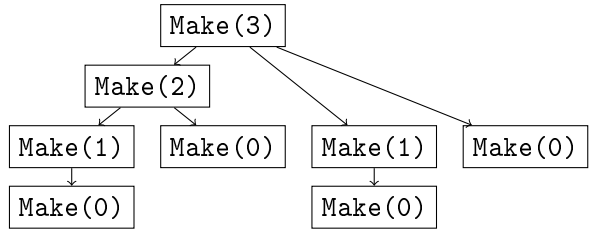
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



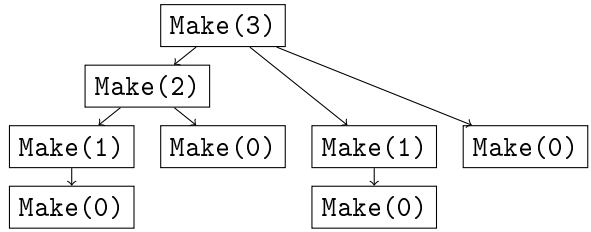
Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



Дерево рекурсивных вызовов

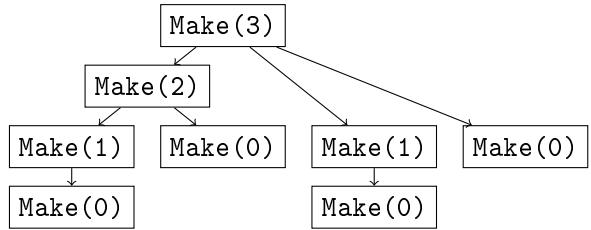
```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



$p(0) = 1$

Дерево рекурсивных вызовов

```
static void Make(int n)
{
    for (int i=n-1; i>=0; i--)
    {
        Console.Write(i + " ");
        Make(i);
    }
}
```



$$p(0) = 1$$

$$p(n) = 1 + \sum_{i=0}^{n-1} p(i)$$

Расчет количества вызовов

$$p(0) = 1$$

Расчет количества вызовов

$$p(0) = 1$$

$$p(n) = 1 + \sum_{i=0}^{n-1} p(i)$$

Расчет количества вызовов

$$p(0) = 1$$

$$p(n) = 1 + \sum_{i=0}^{n-1} p(i)$$

$$p(0) = 1,\ p(1) = 1+1 = 2,\ p(2) = 1+2+1 = 4,\ p(3) = 1+4+2+1 = 8,\ p(4) = 16$$

Расчет количества вызовов

$$p(0) = 1$$

$$p(n) = 1 + \sum_{i=0}^{n-1} p(i)$$

$$p(0) = 1,\ p(1) = 1+1 = 2,\ p(2) = 1+2+1 = 4,\ p(3) = 1+4+2+1 = 8,\ p(4) = 16$$

$$p(n) = 2^n?$$

Расчет количества вызовов

$$p(0) = 1$$

$$p(n) = 1 + \sum_{i=0}^{n-1} p(i)$$

$$p(0) = 1,\ p(1) = 1+1 = 2,\ p(2) = 1+2+1 = 4,\ p(3) = 1+4+2+1 = 8,\ p(4) = 16$$

$$p(n) = 2^n?$$

$$1 + \sum_{i=0}^{n-1} 2^i = 1 + 2^n - 1 = 2^n$$

Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```

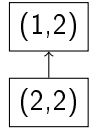
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```

(2,2)

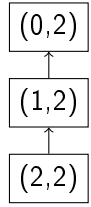
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



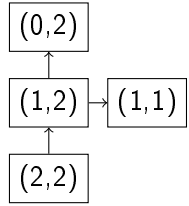
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



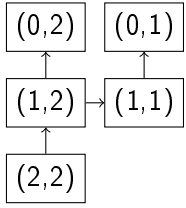
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



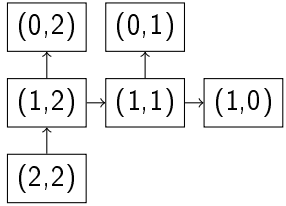
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



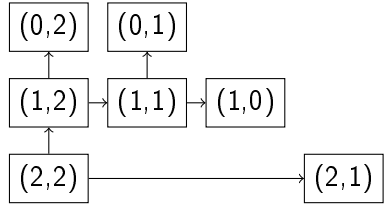
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



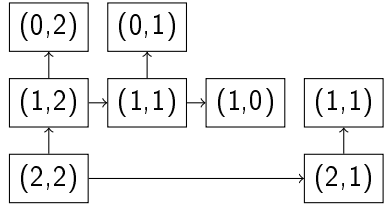
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



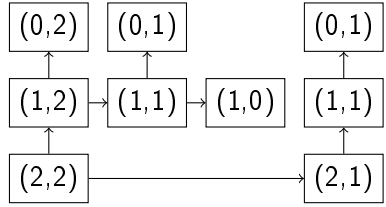
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



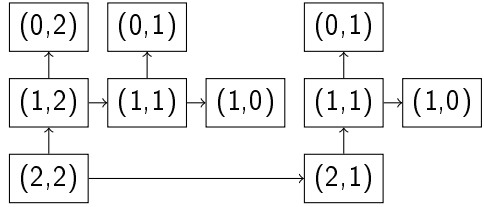
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



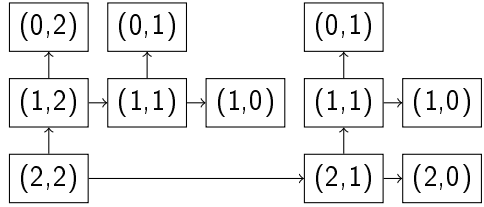
Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



Дерево рекурсивных вызовов

```
static int Make(int x, int y)
{
if (x<=0 || y<=0) return 1;
return Make(x-1,y)+Make(x,y-1);
}
```



Расчет количества итераций

$$p(0, y) = 1$$

$$p(x, 0) = 1$$

$$p(x, y) = 1 + p(x - 1, y) + p(x, y - 1)$$

Расчет количества итераций

$$p(0, y) = 1$$
$$p(x, 0) = 1$$

$$p(x, y) = 1 + p(x - 1, y) + p(x, y - 1)$$

	0	1	2	3	4
0	1	1	1	1	1
1	1	3	5	7	9
2	1	5	11	18	28
3	1	7	18	37	66
4	1	9	28	66	133

Расчет количества итераций

$$p(0, y) = 1$$

$$p(x, 0) = 1$$

$$p(x, y) = 1 + p(x - 1, y) + p(x, y - 1)$$

$$p(x, y) \leq 3^{x+y}$$

Расчет количества итераций

$$p(0, y) = 1$$

$$p(x, 0) = 1$$

$$p(x, y) = 1 + p(x - 1, y) + p(x, y - 1)$$

$$p(x, y) \leq 3^{x+y}$$

$$p(0, 0) = 1 \leq 3^0$$

Расчет количества итераций

$$p(0, y) = 1$$

$$p(x, 0) = 1$$

$$p(x, y) = 1 + p(x - 1, y) + p(x, y - 1)$$

$$p(x, y) \leq 3^{x+y}$$

$$p(0, 0) = 1 \leq 3^0$$

$$p(x, y) = 1 + p(x - 1, y) + p(x, y - 1) \leq 3^{x+y-1} + 3^{x+y-1} + 3^{x+y-1} = 3^{x+y}$$

Расчет количества итераций

$$p'(0, y) = 1$$

$$p'(x, 0) = 1$$

$$p'(x, y) = p'(x - 1, y) + p'(x, y - 1) < p(x, y)$$

	0	1	2	3	4
0	1	1	1	1	1
1	1	2	3	4	5
2	1	3	6	10	15
3	1	4	10	20	35
4	1	5	15	35	70

Расчет количества итераций

$$p'(0, y) = 1$$

$$p'(x, 0) = 1$$

$$p'(x, y) = p'(x - 1, y) + p'(x, y - 1) < p(x, y)$$

	0	1	2	3	4
0	1	1	1	1	1
1	1	2	3	4	5
2	1	3	6	10	15
3	1	4	10	20	35
4	1	5	15	35	70

$$p'(x, y) = C^x_{x+y} = \frac{(x + y)!}{x!y!}$$