

# Experimentos em Complexidade

## Trabalho 1 – Relatório

Giovana Nascimento Raupp

Prof. Isabel Harb Manssour

Os algoritmos foram testados com os valores:  $n=10$ ;  $n<100$ ;  $n+=10$ .  
Como consta na Classe App:

```
public class App {
    public static void main(String[] args) {
        Funcoes f = new Funcoes();
        for(int n = 10; n<100; n+=10) { → valores para n
            int r = f.f1(n); → f1 - modificado a cada teste
            System.out.println(n+" "+r);
        }
    }
}
```

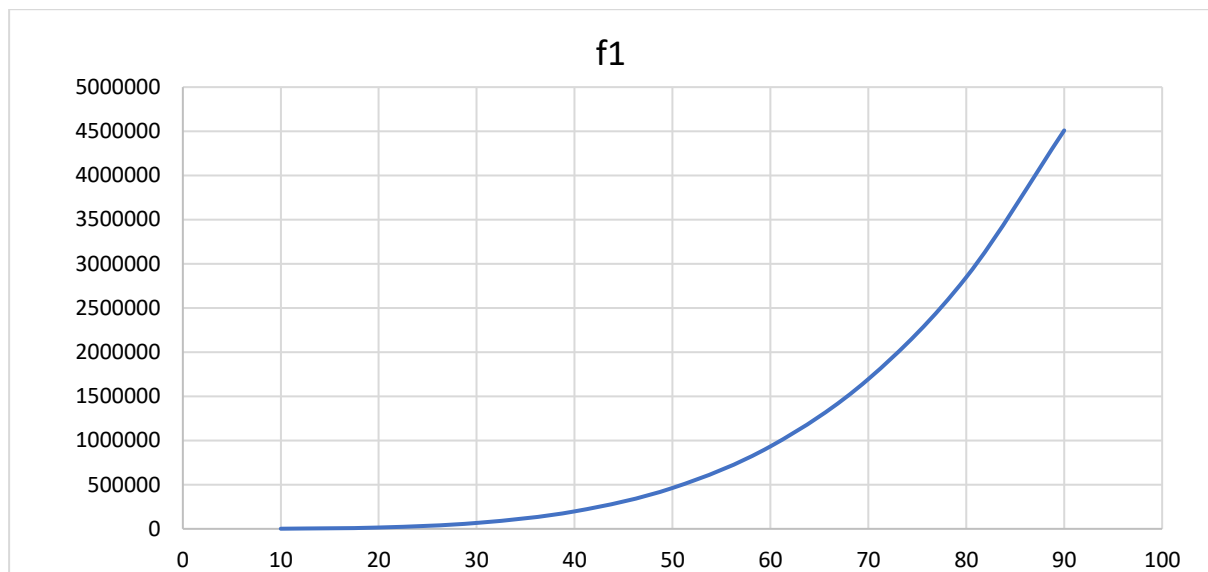
- *Algoritmo 1*

```
int f1( int n ) {
    int i, j, k, res = 0
    int cont_op = 0
    for( i = 1; i <= n+1; i += 1 )
        for( j = 1; j <= i*i; j += i+1 )
            for( k = i/2; k <= n+j; k += 2 ) {
                res = res + n-1
                cont_op++
            }
    return cont_op
}
```

- *Tabela*

n	f1
<u>10</u>	<u>1369</u>
<u>20</u>	<u>15081</u>
30	66904
40	197561
50	462839
60	933441
70	1695174
80	2848721
90	4509909

- *Gráfico*



- *Cálculo do b*

$$f(10) = 1369$$

$$f(20) = 15081$$

$$b \approx \frac{\log(15081) - \log(1369)}{\log(20) - \log(10)} \approx \frac{4,17843 - 3,13640345}{1,30103 - 1} \approx \frac{1,042027}{0,30103} \approx \underline{\underline{3,461538}}$$

- *Conclusão*

Função Polinomial: O algoritmo cresce como uma função de quarto grau, visto que b, o expoente, se aproxima de 4. Ao fazer outros testes com outros valores da tabela, b se aproxima cada vez mais de 4.

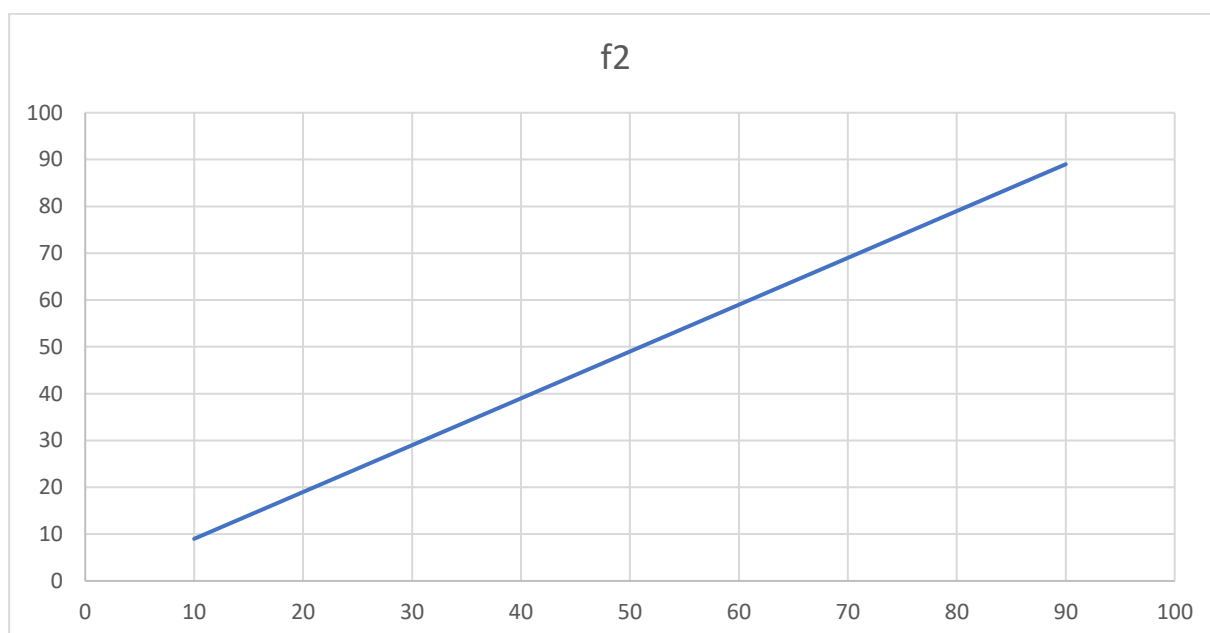
- *Algoritmo 2*

```
int f2( int n ) {
    int i, j, k, res = 0
    int cont_op = 0
    for( i = n; i <= n; i += i/2+1 )
        for( j = i/2; j <= i*i; j += i+1 )
            for( k = n; k <= 2*n; k += i+1 ) {
                res = res + n
                cont_op++
            }
    return cont_op
}
```

- Tabela*

n	f2
<u>10</u>	<u>9</u>
<u>20</u>	<u>19</u>
30	29
40	39
50	49
60	59
70	69
80	79
90	89

- Gráfico*



- Cálculo do b*

$$f(10) = 9$$

$$f(20) = 19$$

$$b \approx \frac{\log(19) - \log(9)}{\log(20) - \log(10)} \approx \frac{1,278754 - 0,95424251}{1,30103 - 1} \approx \frac{0,324511}{0,30103} \approx \underline{1,078003}$$

- Conclusão*

Função Polinomial: O algoritmo cresce como uma função do primeiro grau (linear), visto que b, o expoente, se aproxima de 1.

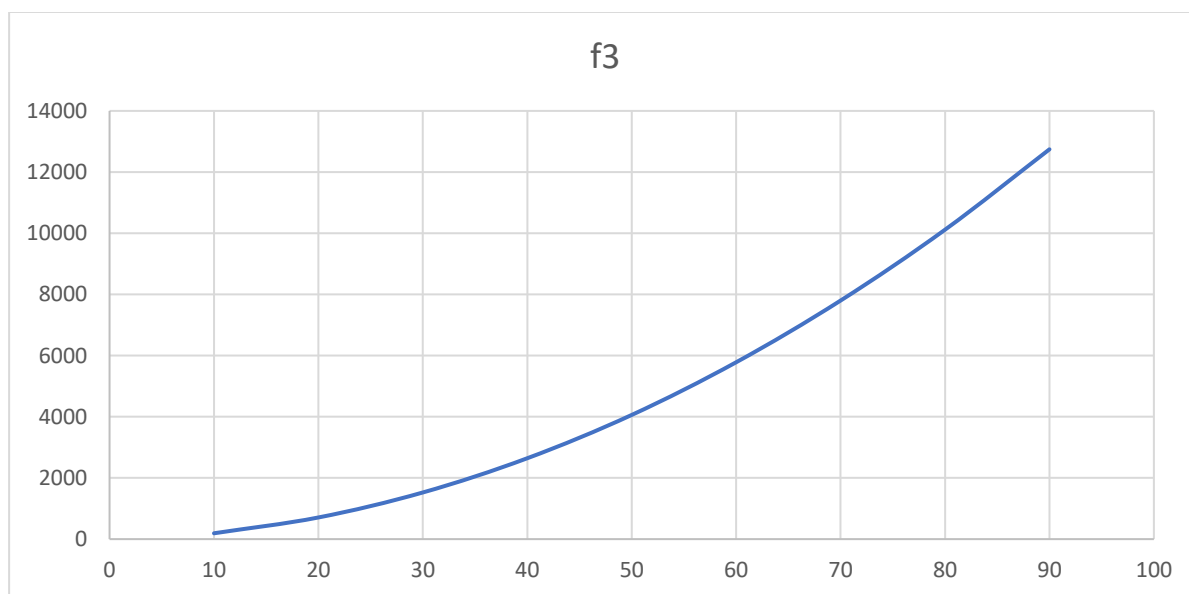
- *Algoritmo 3*

```
int f3( int n ) {  
    int i, j, k, res = 0  
    int cont_op = 0  
    for( i = 1; i <= n*n; i += 2 )  
        for( j = i/2; j <= 2*i; j += i/2+1 )  
            for( k = j+1; k <= n+j; k += k/2+1 ) {  
                res = res + abs(j-i)  
                cont_op++  
            }  
    return cont_op  
}
```

- *Tabela*

n	f3
<u>10</u>	<u>191</u>
<u>20</u>	<u>708</u>
30	1525
40	2644
50	4060
60	5782
70	7800
80	10116
90	12744

- *Gráfico*



- *Cálculo do b*

$$f(10) = 191$$

$$f(20) = 708$$

$$b \approx \frac{\log(708) - \log(191)}{\log(20) - \log(10)} \approx \frac{2,850033 - 2,28103337}{1,30103 - 1} \approx \frac{0,569}{0,30103} \approx \underline{1,890177}$$

- *Conclusão*

Função Polinomial: O algoritmo cresce como uma função do segundo grau (quadrática), visto que b, o expoente, se aproxima de 2.

- *Algoritmo 4*

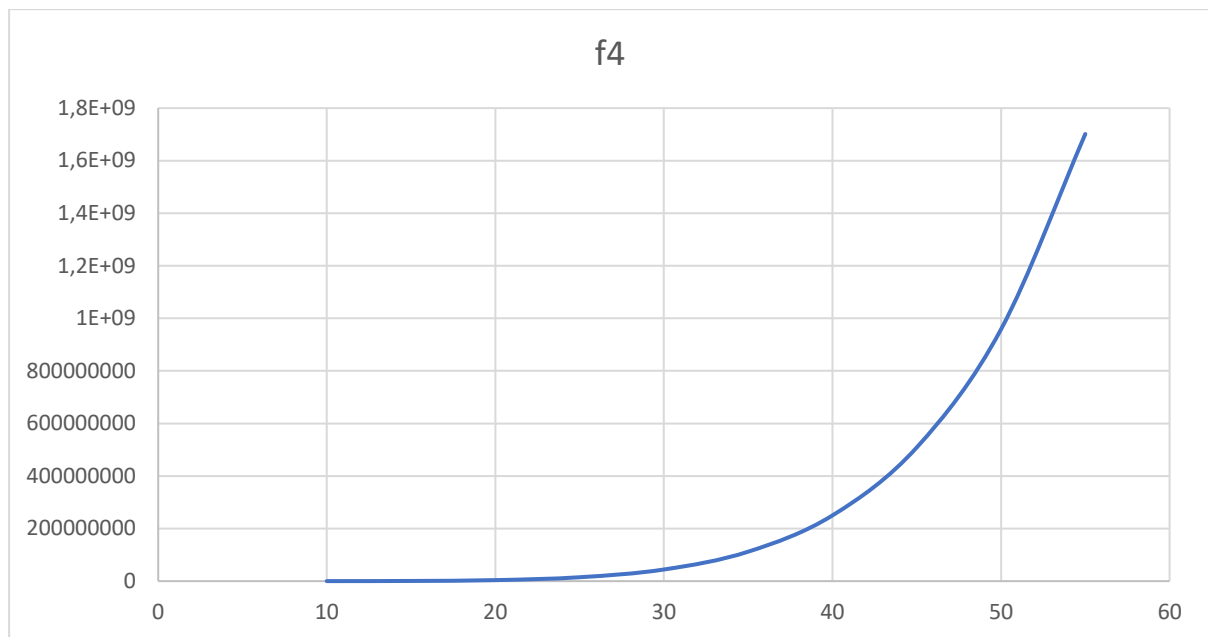
```
int f4( int n ) {
    int i, j, k, res = 0
    int cont_op = 0
    for( i = n; i <= n*n; i += 2 )
        for( j = n+1; j <= n*n; j += 2 )
            for( k = j; k <= 2*j; k += 2 ) {
                res = res + 1
                cont_op++
            }
    return cont_op
}
```

OBS: O algoritmo 4 foi testado com os valores de n=10; n<60; n+=5 por questões de processamento.

- *Tabela*

n	f4
<u>10</u>	<u>57960</u>
<u>15</u>	<u>678930</u>
20	3828595
25	14764050
30	44190780
35	112059920
40	250068390
45	508695165
50	958180300
55	1701373410

- *Gráfico*



- *Cálculo do b*

$$f(10) = 57960$$

$$f(15) = 678930$$

$$b \approx \frac{\log(57960) - \log(678930)}{\log(15) - \log(10)} \approx \frac{5,831825 - 4,76312838}{1,176091 - 1} \approx \frac{1,068697}{0,176091} \approx \underline{\underline{6,068993}}$$

- *Conclusão*

Função Polinomial: O algoritmo cresce como uma função de sexto grau, visto que b, o expoente, se aproxima de 6.

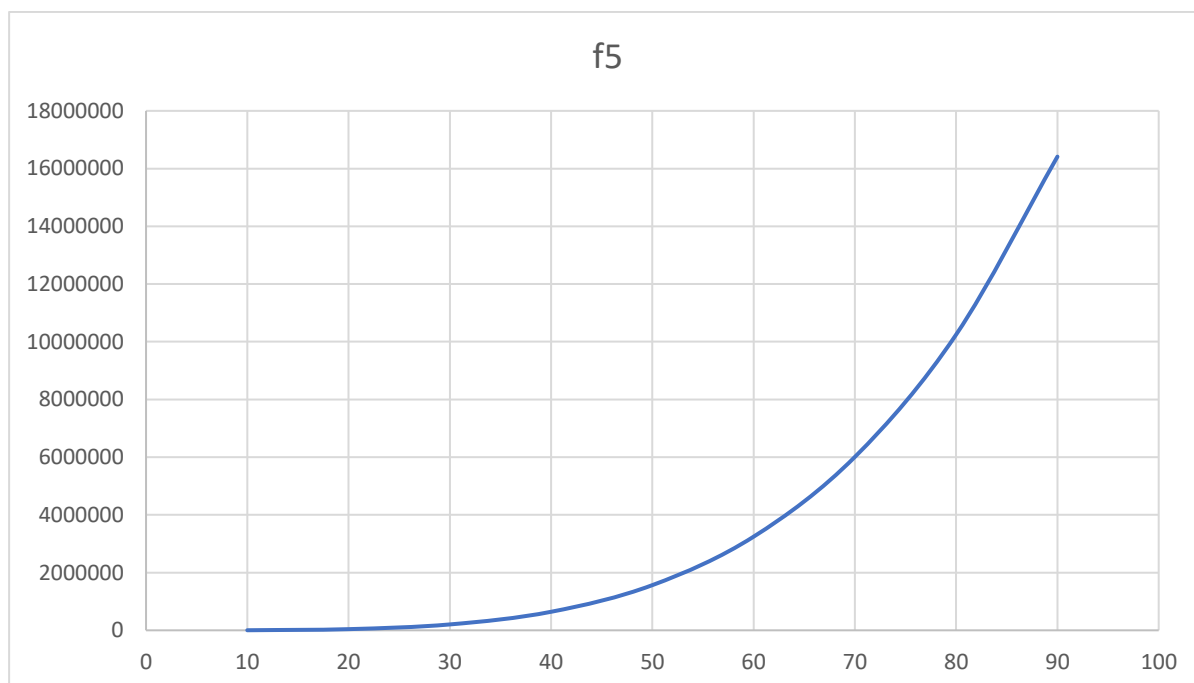
- *Algoritmo 5*

```
int f5( int n ) {
    int i, j, k, res = 0
    int cont_op = 0
    for( i = 1; i <= n*n; i += 1 )
        for( j = 1; j <= i; j += 2 )
            for( k = n+1; k <= 2*i; k += i*j ) {
                res = res + k+1
                cont_op++
            }
    return cont_op
}
```

- Tabela*

n	f5
<u>10</u>	<u>2631</u>
<u>20</u>	<u>40550</u>
30	203756
40	642250
50	1566031
60	3245100
70	6009456
80	10249100
90	16414031

- Gráfico*



- Cálculo do b*

$$f(10) = 2631$$

$$f(20) = 40550$$

$$b \approx \frac{\log(40550) - \log(2631)}{\log(20) - \log(10)} \approx \frac{4,607991 - 3,42012085}{1,30103 - 1} \approx \frac{1,18787}{0,30103} \approx \underline{3,946019}$$

- Conclusão*

Função Polinomial: O algoritmo cresce como uma função de quarto grau, visto que b, o expoente, se aproxima de 4.