Trabalho 1 – ALEST I

IMPLEMENTAÇÃO ALGORITMOS

```java
import static java.lang.Math.abs;

public class Funcao{

   // Algoritmo 1

   public long fa(long n) {

       long i, j, k, res = 0;

       long 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;

   }

   // Algoritmo 2

   public int fb( 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++;

               }

           }

       }
```

```java
        return cont_op;
    }


// Algoritmo 3
    public int fc( 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;
    }


        // Algoritmo 4
        public long fd( long n ) {
            long i, j, k, res = 0;
            long cont_op = 0;
            for (i = 1; i <= n * n; i += 2) {
                for (j = 2; j <= n + 1; j += i + 1) {
                    for (k = j / 2; k <= i + 1; k += j / 2 + 1) {
                        res = res + j + 1;
                        cont_op++;
                    }
                }
            }
            return cont_op;
```

```
        }
    // Algoritmo 5
    public long fe( long n ) {
        long i, j, k, res = 0;
        long cont_op = 0;
        for (i = n / 2; i <= n; i += 3) {
            for (j = i; j <= i * i; j += 2) {
                for (k = i; k <= 2 * j; k += 1) {
                    res = res + n + j;
                    cont_op++;
                }
            }
        }
        return cont_op;
    }
}
```

MARGEM PARA MEDIR AS OPERAÇÕES

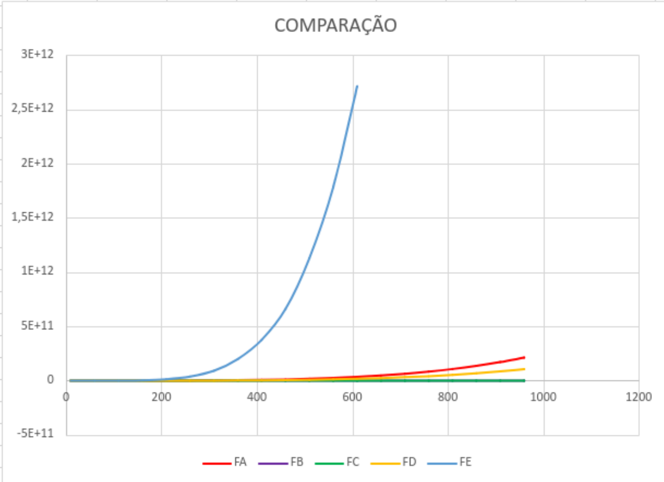
```
for (int n=10; n<1000; n+=50)
```

Observações:

Ao efetuar o cálculo, percebe-se que, conforme a escolha dos valores da coluna de *n* e de *operações*, a complexidade *O* pode variar. Desta forma, a tendência de *b* resulta em diferentes valores na função característica do algoritmo.

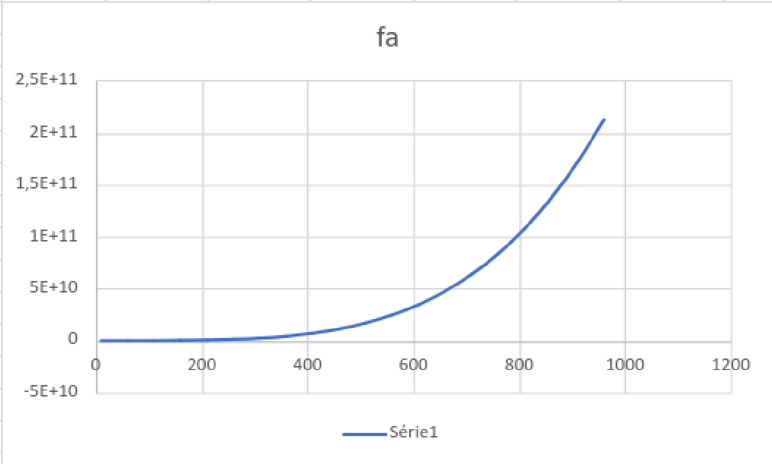
A partir disso, criei uma tabela com os resultados de todos os algoritmos para comparar o desempenho entre si. Os pontos principais para análise iniciam com o fato de que a linha de *FB* no gráfico está escondida entre *FC*, *FD* e *FA*, por ter os menores valores de operações comparados às demais, já que, por ser linear, o tempo de execução prevalece diante das outras. Em contrapartida, o *n* de *FE* foi possível de se executar até *n*=610 devido ao longo tempo de processamento do código por conta das operações terem valores muito expressivos e se tratar de uma função de quarta ordem. Por fim, a similaridade das curvas de FA e FD confirmam que a sua complexidade segue o mesmo padrão.

N	FA	FB	FC	FD	FE
10	2631	9	191	1279	2171
60	3245100	59	5782	1620929	28113635
110	36619756	109	18882	18304329	528569588
160	163876600	159	39485	81926479	3315936231
210	486265631	209	67591	243112379	13631687595
260	1142536850	259	103191	571237029	38637191923
310	2308940256	309	146297	1154425429	91461359041
360	4199225850	359	196908	2099552579	1,99075E+11
410	7064643631	409	255011	3532243479	3,75798E+11
460	11193943600	459	320615	5596873129	6,60305E+11
510	16913375756	509	393724	8456566529	1,12951E+12
560	24586690100	559	474332	12293198679	1,78426E+12
610	34615136631	609	562440	17307394579	2,71255E+12
660	47437465350	659	658038	23718529229	
710	63529926256	709	761154	31764727629	
760	83406269350	759	871751	41702864779	
810	1,07618E+11	809	989863	53808565679	
860	1,36753E+11	859	1115470	68376205329	
910	1,71439E+11	909	1248573	85718908729	
960	2,12338E+11	959	1389180	1,06169E+11	



ALGORITMO FA

10	2631
60	3245100
110	36619756
160	163876600
210	486265631
260	1142536850
310	2308940256
360	4199225850
410	7064643631
460	11193943600
510	16913375756
560	24586690100
610	34615136631
660	47437465350
710	63529926256
760	83406269350
810	1,07618E+11
860	1,36753E+11
910	1,71439E+11
960	2,12338E+11

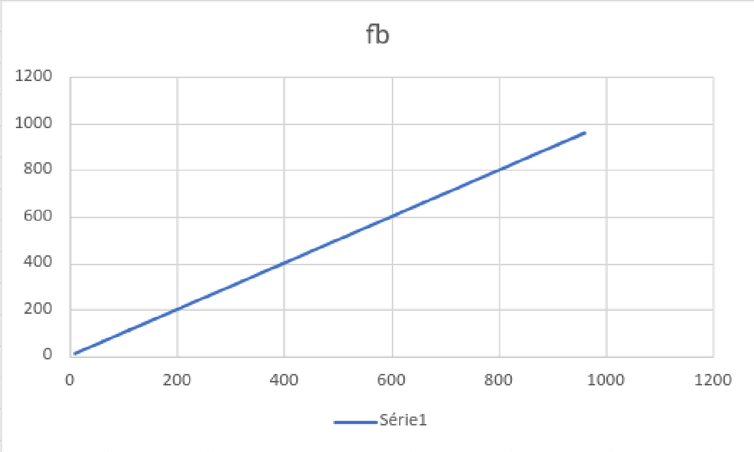


Se selecionar os valores marcados acima para verificar '***b'***, o resultado da fórmula é:

$b = \frac{log(163876600) - log(3245100)}{log(160) - log(60)} = 3,482248797 \simeq mais\ próximo\ de\ 3\ do\ que\ 4$, portanto, a classe de complexidade é n^3 (função cúbica).

ALGORITMO FB

10	9
60	59
110	109
160	159
210	209
260	259
310	309
360	359
410	409
460	459
510	509
560	559
610	609
660	659
710	709
760	759
810	809
860	859
910	909
960	959

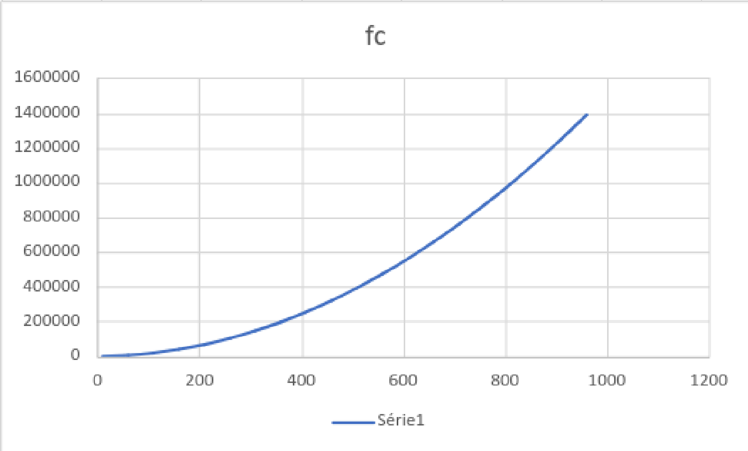


Se selecionar os valores marcados acima para verificar '***b'***, o resultado da fórmula é:

$b = \frac{log(109) - log(9)}{log(110) - log(10)} = 0,569979683 \simeq mais\ próximo\ de\ 1\ do\ que\ 0$, portanto, a classe de complexidade é n^1 (função linear).

ALGORITMO FC

10	191
60	5782
110	18882
160	39485
210	67591
260	103191
310	146297
360	196908
410	255011
460	320615
510	393724
560	474332
610	562440
660	658038
710	761154
760	871751
810	989863
860	1115470
910	1248573
960	1389180

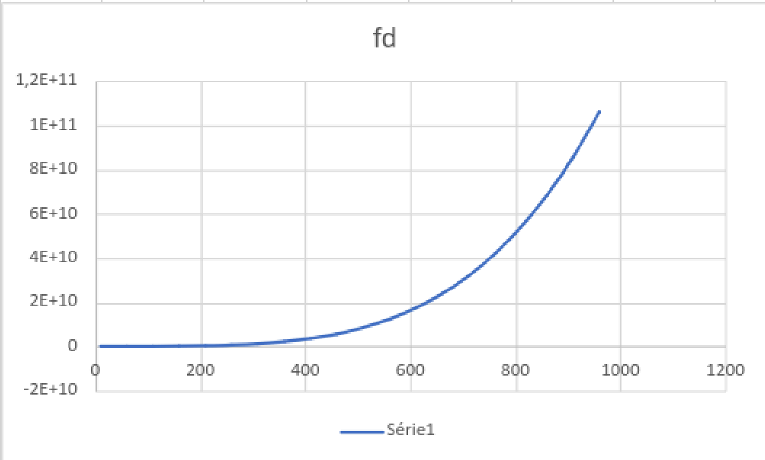


Se selecionar os valores marcados acima para verificar '***b'***, o resultado da fórmula é:

$b\ =\ \frac{log(18882) - log(191)}{log(110) - log(10)}\ = 2,158657212\ \simeq mais\ próximo\ de\ 2\ do\ que\ 1$, portanto, a classe de complexidade é n^2 (função quadrática). Entretanto, ao analisar com os demais percebe-se que o desempenho é similar a de uma função linear.

ALGORITMO FD

10	1279
60	1620929
110	18304329
160	81926479
210	243112379
260	571237029
310	1154425429
360	2099552579
410	3532243479
460	5596873129
510	8456566529
560	12293198679
610	17307394579
660	23718529229
710	31764727629
760	41702864779
810	53808565679
860	68376205329
910	85718908729
960	1,06169E+11

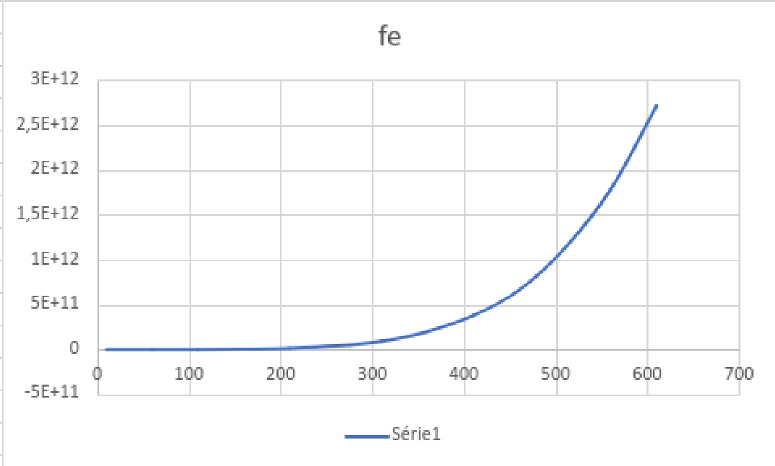


Se selecionar os valores marcados acima para verificar ‘***b’***, o resultado da fórmula é:

$b \ = \ \frac{log(81926479) - log(1620929)}{log(160) - log(60)} \ = 3,317929138 \ \simeq \ mais\ próximo\ de\ 3\ do\ que\ 2$, portanto, a classe de complexidade é n^3 (função cúbica).

ALGORITMO FE

10	2171
60	28113635
110	528569588
160	3315936231
210	13631687595
260	38637191923
310	91461359041
360	1,99075E+11
410	3,75798E+11
460	6,60305E+11
510	1,12951E+12
560	1,78426E+12
610	2,71255E+12



Se selecionar os valores marcados acima para verificar '***b'***, o resultado da fórmula é:

$b\ =\ \frac{log(3315936231)\ -\ log(28113635)}{log(160)\ -\ log(60)}\ =\ 4,36291249\ \simeq\ mais\ próximo\ de\ 4\ do\ que\ 3$, portanto, a classe de complexidade é n^4 (função de quarta ordem).