UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS	**Universidade Federal de Minas Gerais** **Turma:** Ciência da Computação **Prof.:** Gisele e Wagner **Nome:** Rubia Alice Moreira de Souza **Matrícula:** 2022043507

Prática 2 - Análise de Desempenho

Belo Horizonte
2022

Sumário

1. Plano de experimentos de desempenho computacional

2. Resultados desempenho computacional

2.1. AnaliseMemoria

2.1.1. Multiplicação

2.1.2. Soma

2.1.3. Transposto

2.2. VetorEstatico

2.2.1. Produto Interno

2.2.1.1. 100

```
I 1 254562.574472362
F 2 254562.617339069 0.042866707
```

2.2.1.2. 200

```
I 1 254888.253546165
F 2 254888.253575669 0.000029504
```

2.2.1.3. 300

```
I 1 254888.264763125
F 2 254888.264803796 0.000040671
```

2.2.1.4. 400

```
I 1 254888.274335435
F 2 254888.274375542 0.000040107
```

2.2.1.5. 500

```
I 1 254888.283020725
F 2 254888.283074359 0.000053634
```

2.2.2. Norma

2.2.2.1. 100

```
I 1 254888.245297449
F 2 254888.245317604 0.000020155
```

2.2.2.2. 200

```
I 1 254888.258222163
F 2 254888.258244641 0.000022478
```

2.2.2.3. 300

```
I 1 254888.267715606
F 2 254888.267745410 0.000029804
```

2.2.2.4. 400

```
I 1 254888.277441648
F 2 254888.277476624 0.000034976
```

2.2.2.5. 500

```
I 1 254888.287386750
F 2 254888.287437468 0.000050718
```

2.2.3. Soma

2.2.3.1. 100

```
I 1 254888.240343472
F 2 254888.240380873 0.000037401
```

2.2.3.2. 200

```
I 1 254888.248010769
F 2 254888.248039996 0.000029227
```

2.2.3.3. 300

```
I 1 254888.261914917
F 2 254888.261950074 0.000035157
```

2.2.3.4. 400

```
I 1 254888.270869780
F 2 254888.270912351 0.000042571
```

2.2.3.5. 500

```
I 1 254888.280105477
F 2 254888.280177967 0.000072490
```

2.3. VetorDinamico

2.3.1. Produto Interno

2.3.1.1. 1M

```
I 1 255301.233445754
F 2 255301.290950103 0.057504349
```

2.3.1.2. 2M

```
I 1 255301.427894364
F 2 255301.514109285 0.086214921
```

2.3.1.3. 3M

```
I 1 255302.146428578
F 2 255302.296704753 0.150276175
```

2.3.1.4. 4M

```
I 1 255304.157597536
F 2 255304.328919416 0.171321880
```

2.3.1.5. 5M

```
I 1 255306.033850601
F 2 255306.258852733 0.225002132
```

2.3.2. Norma

2.3.2.1. 1M

```
I 1 255301.294175032
F 2 255301.316129585 0.021954553
```

2.3.2.2. 2M

```
I 1 255301.516815577
F 2 255301.561978585 0.045163008
```

2.3.2.3. 3M

```
I 1 255303.457869837
F 2 255303.519403575 0.061533738
```

2.3.2.4. 4M

```
I 1 255304.331639561
F 2 255304.405805429 0.074165868
```

2.3.2.5. 5M

```
I 1 255306.262175432
F 2 255306.353914311 0.091738879
```

2.3.3. Soma

2.3.3.1. 1M

```
I 1 255301.167384475
F 2 255301.230012161 0.062627686
```

2.3.3.2. 2M

```
I 1 255301.318887652
F 2 255301.423782002 0.104894350
```

2.3.3.3. 3M

```
I 1 255301.565049238
F 2 255301.721442405 0.156393167
```

2.3.3.4. 4M

```
I 1 255303.918009986
F 2 255304.152617183 0.234607197
```

2.3.3.5. 5M

```
I 1 255304.408731229
F 2 255304.663114453 0.254383224
```

3. Plano de experimentos de localidade de referência

4. Análise de Localidade de Referência

4.1. Mapa de acesso

4.1.1. AnaliseMemoria

4.1.1.1. Multiplicação

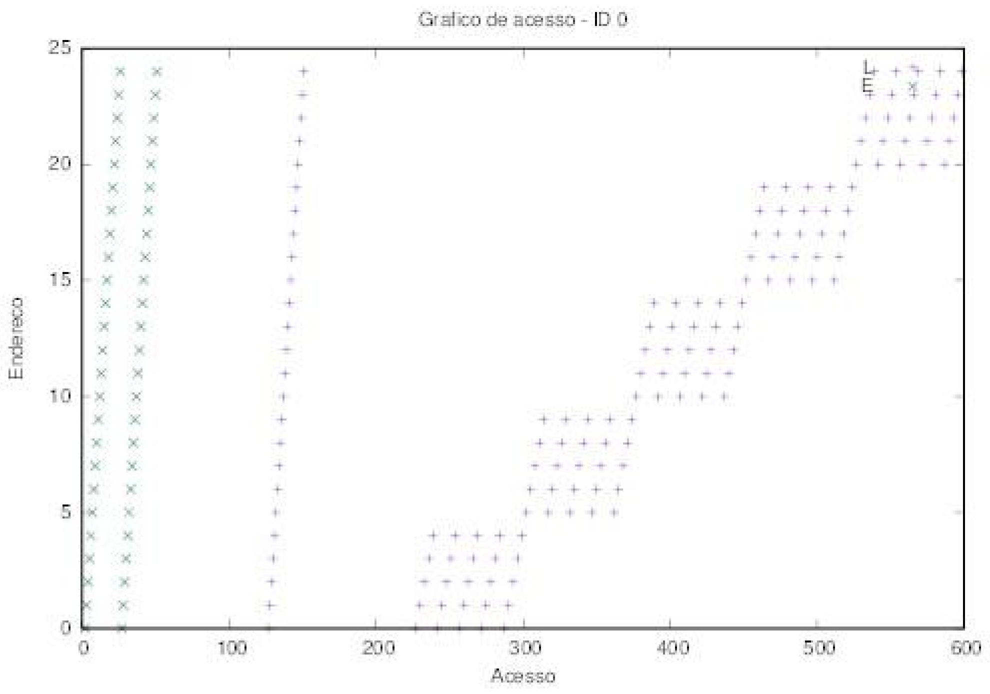


Grafico de acesso - ID 0
Endereco
Acesso
L
E
0
5
10
15
20
25
100
200
300
400
500
600

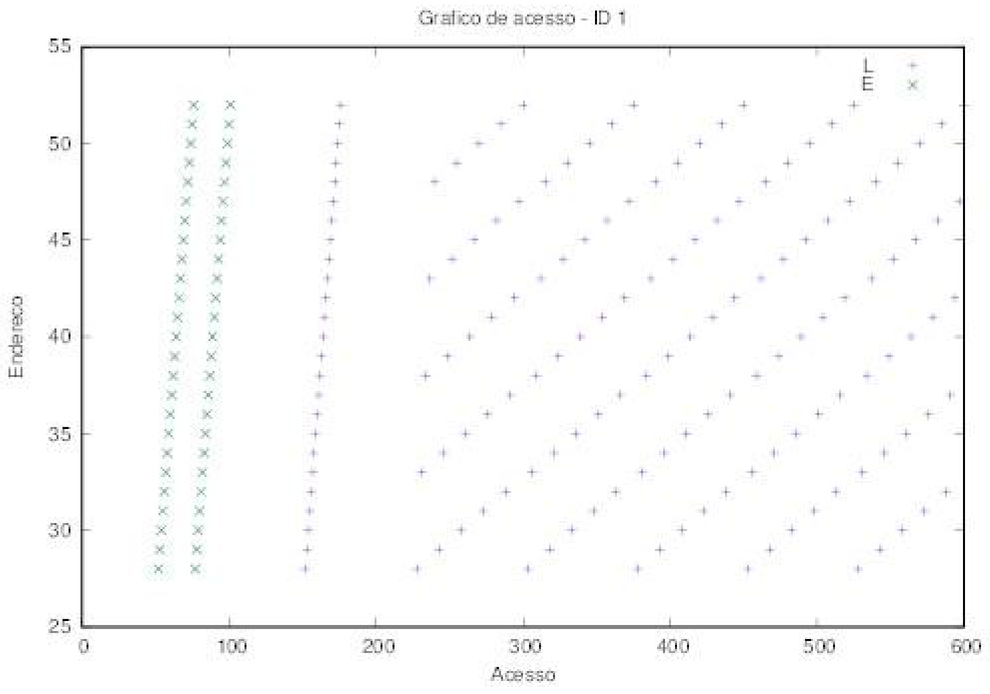


Grafico de acesso - ID 1
Endereco
Acesso
L
E
25
30
35
40
45
50
55
0
100
200
300
400
500
600

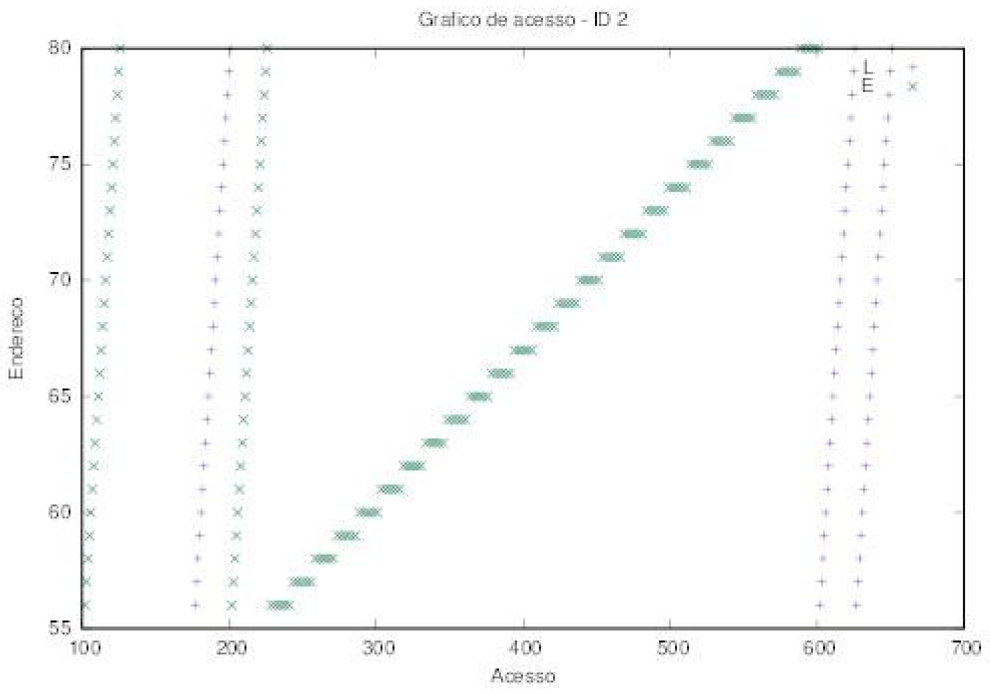


Grafico de acesso - ID 2
L
E
Endereco
Acesso
80
75
70
65
60
55
100
200
300
400
500
600
700

4.1.1.2. Soma

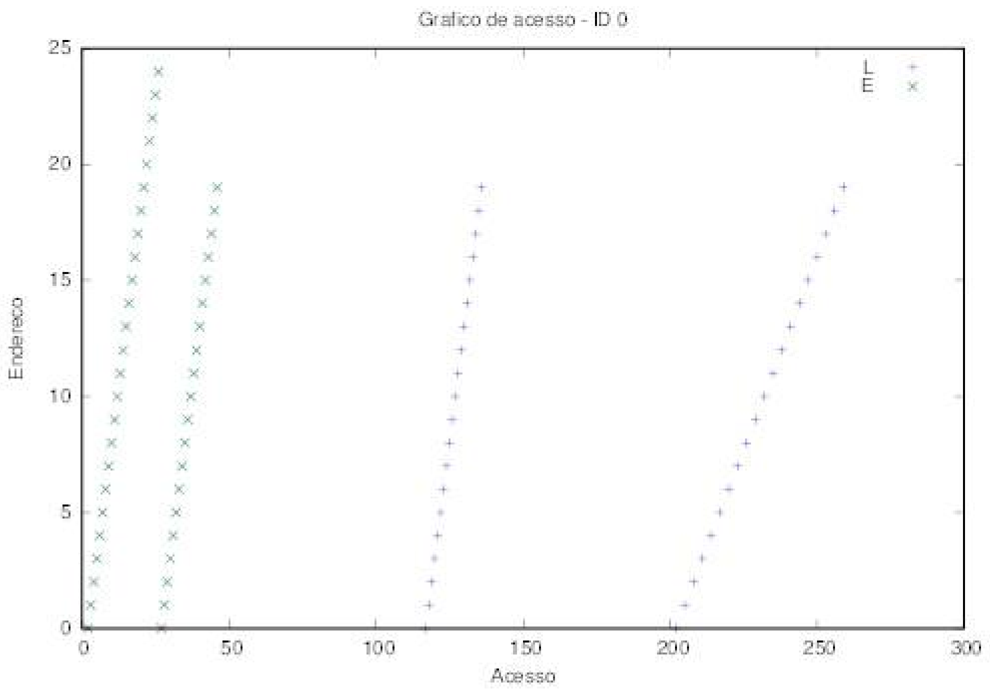


Grafico de acesso - ID 0
Endereco
Acesso
L
E
0
5
10
15
20
25
0
50
100
150
200
250
300

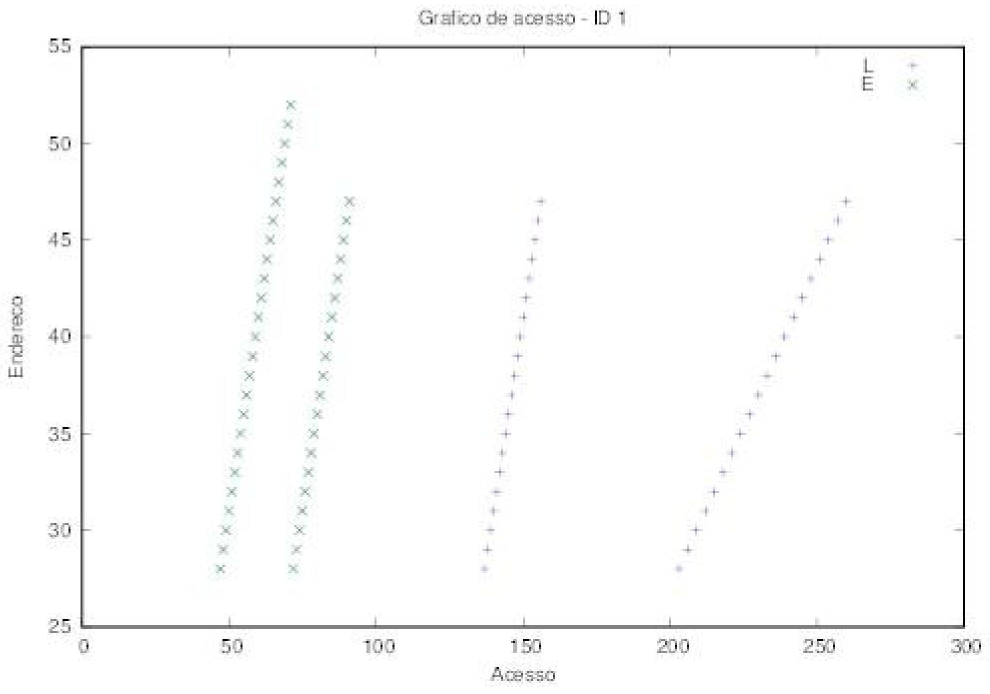
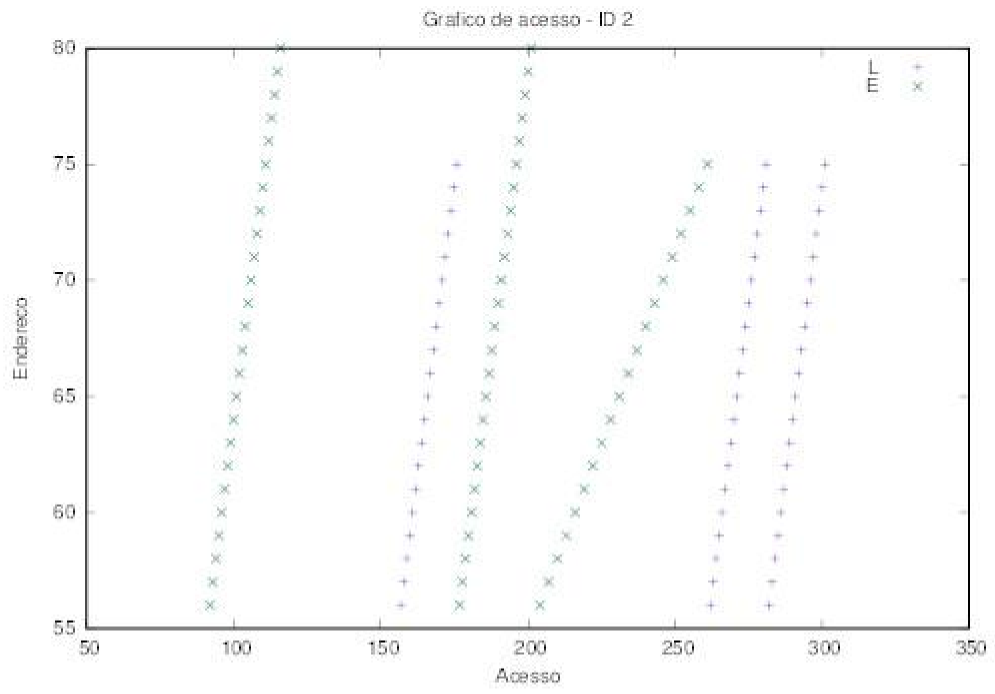
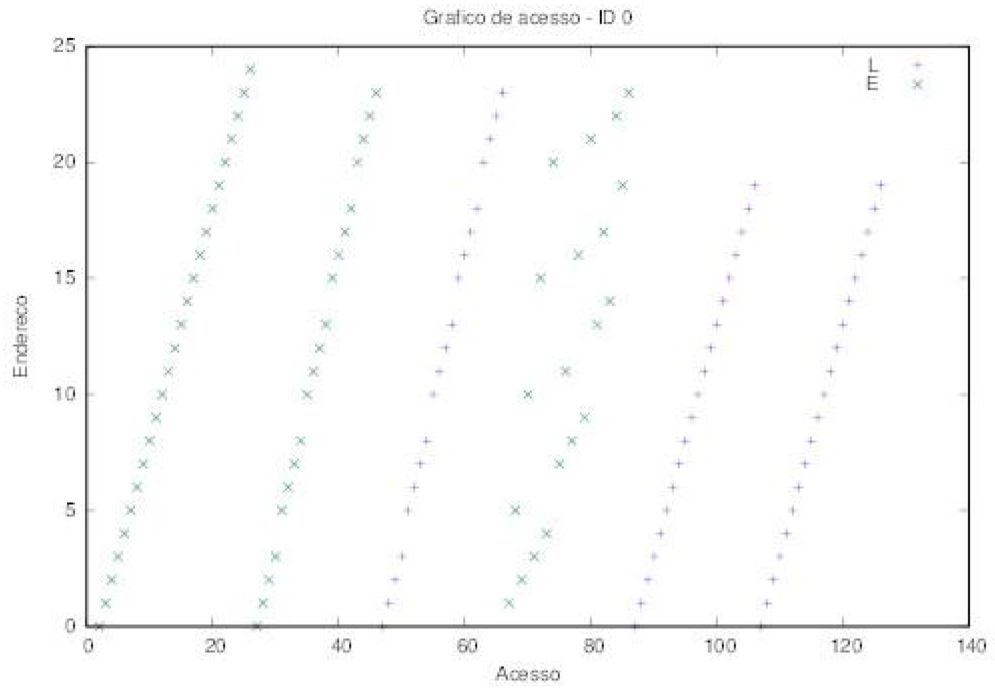


Grafico de acesso - ID 1
Endereco
Acesso
L
E
25
30
35
40
45
50
55
0
50
100
150
200
250
300



4.1.1.3. Transposto



4.1.2. VetorDinamico

4.1.2.1. Produto Interno

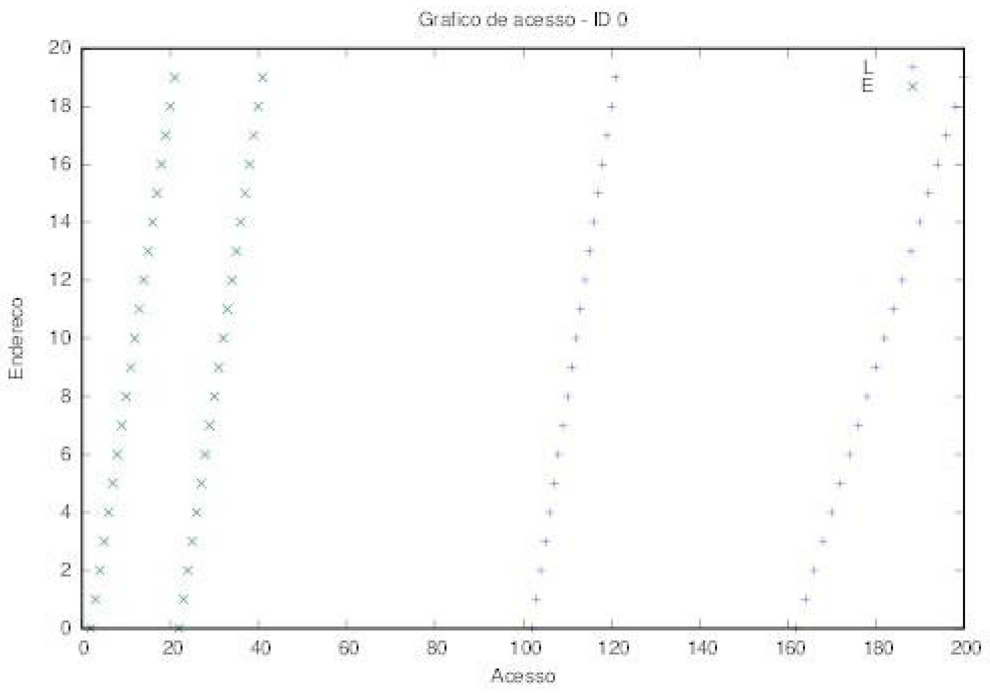


Grafico de acesso - ID 0
Endereco
Acesso
L
E
0
2
4
6
8
10
12
14
16
18
20
0
20
40
60
80
100
120
140
160
180
200

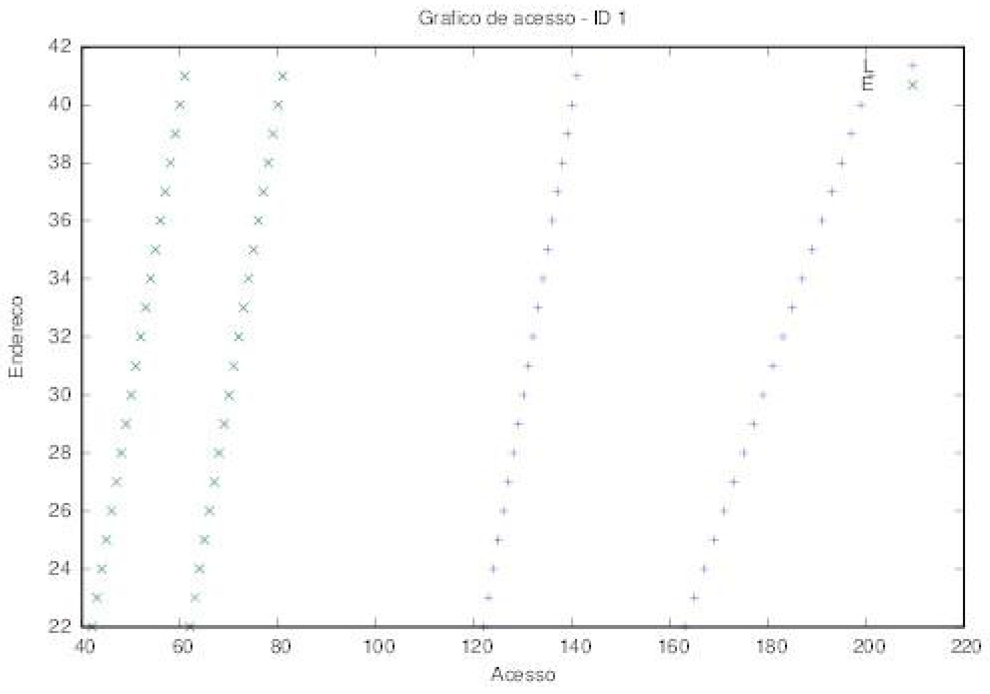


Grafico de acesso - ID 1
Endereco
Acesso
L
E
22
24
26
28
30
32
34
36
38
40
42
40
60
80
100
120
140
160
180
200
220

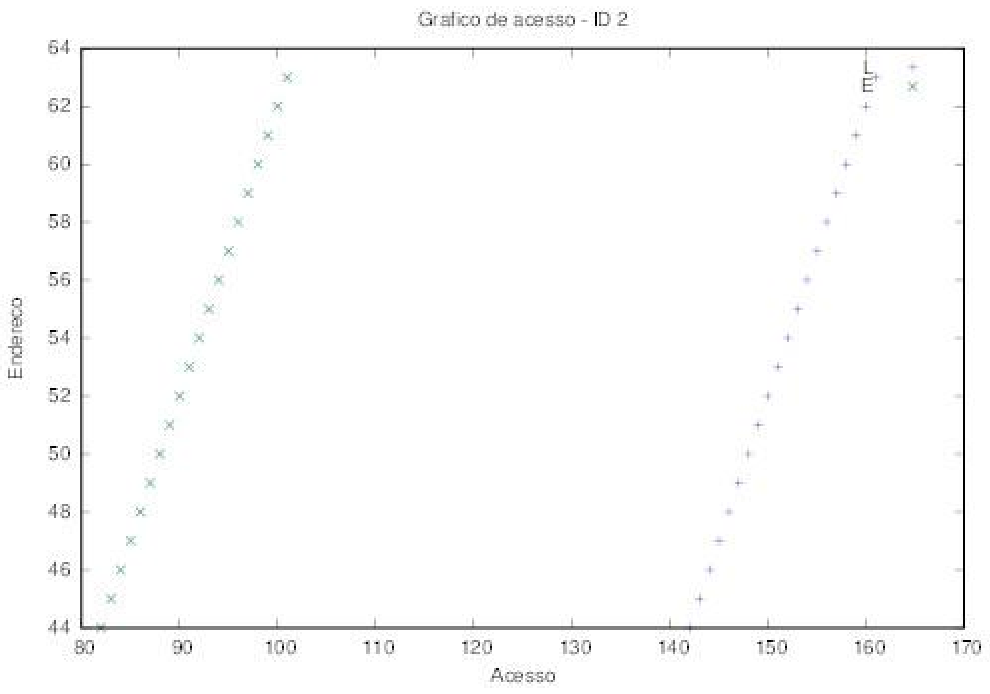


Grafico de acesso - ID 2
Endereco
Acesso
L
E

4.1.2.2. Norma

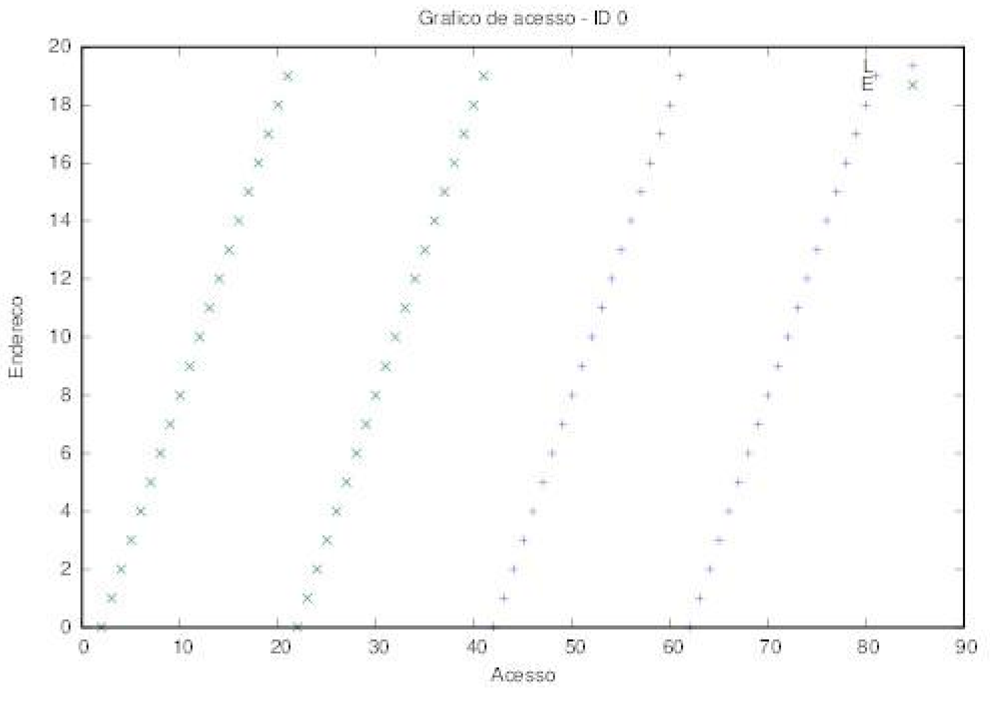


Grafico de acesso - ID 0
Endereco
Acesso
L
E

4.1.2.3. Soma

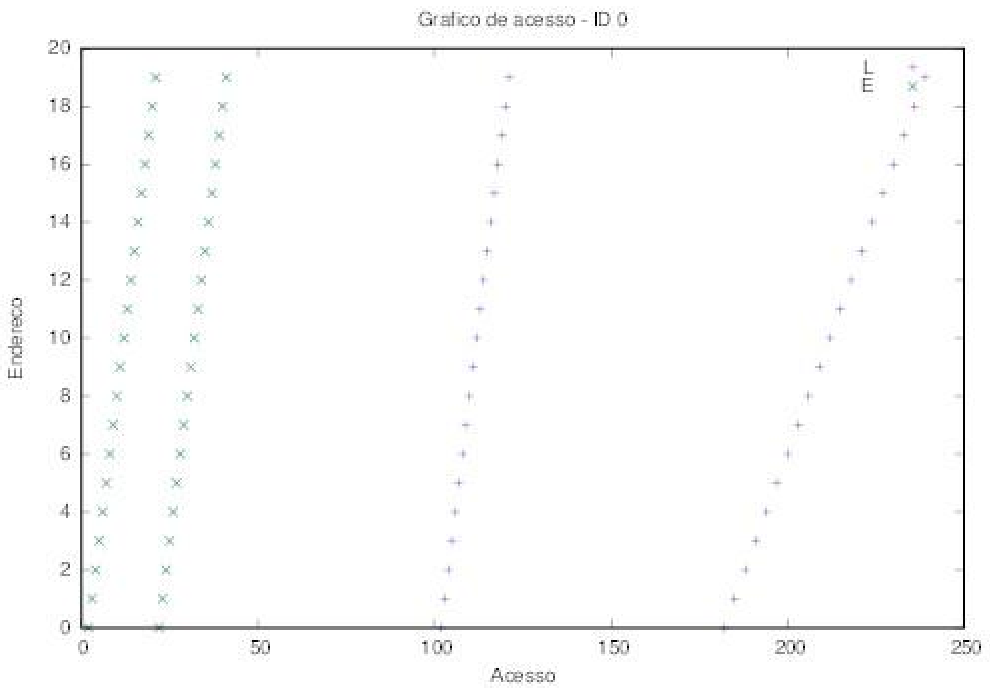


Grafico de acesso - ID 0
Endereco
Acesso
L
E
0
50
100
150
200
250
0
2
4
6
8
10
12
14
16
18
20

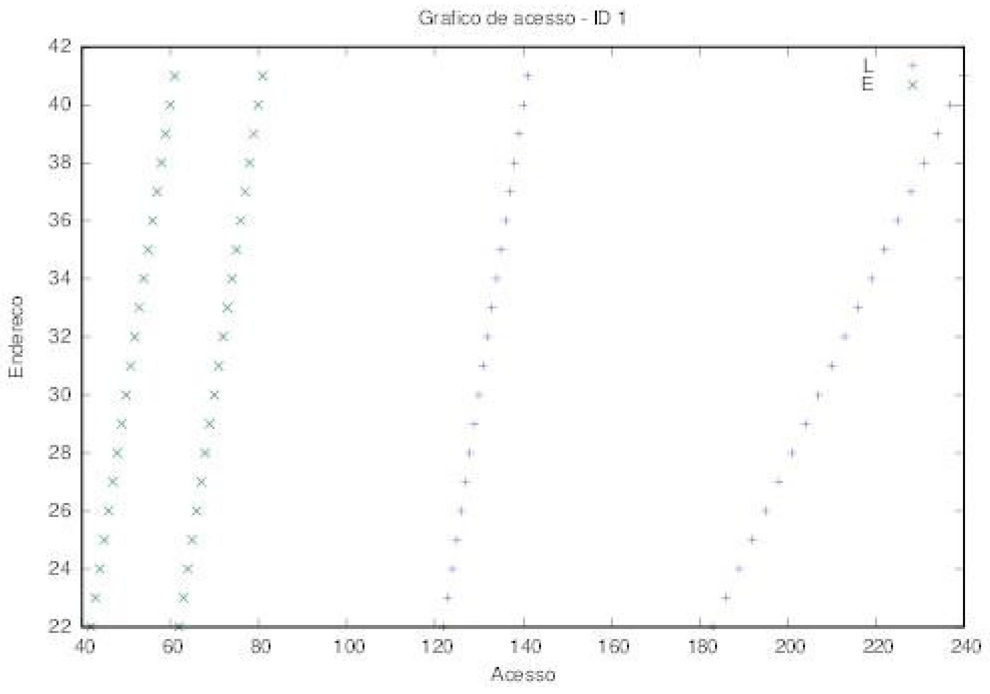


Grafico de acesso - ID 1
Endereco
Acesso
L
E
40
60
80
100
120
140
160
180
200
220
240
22
24
26
28
30
32
34
36
38
40
42

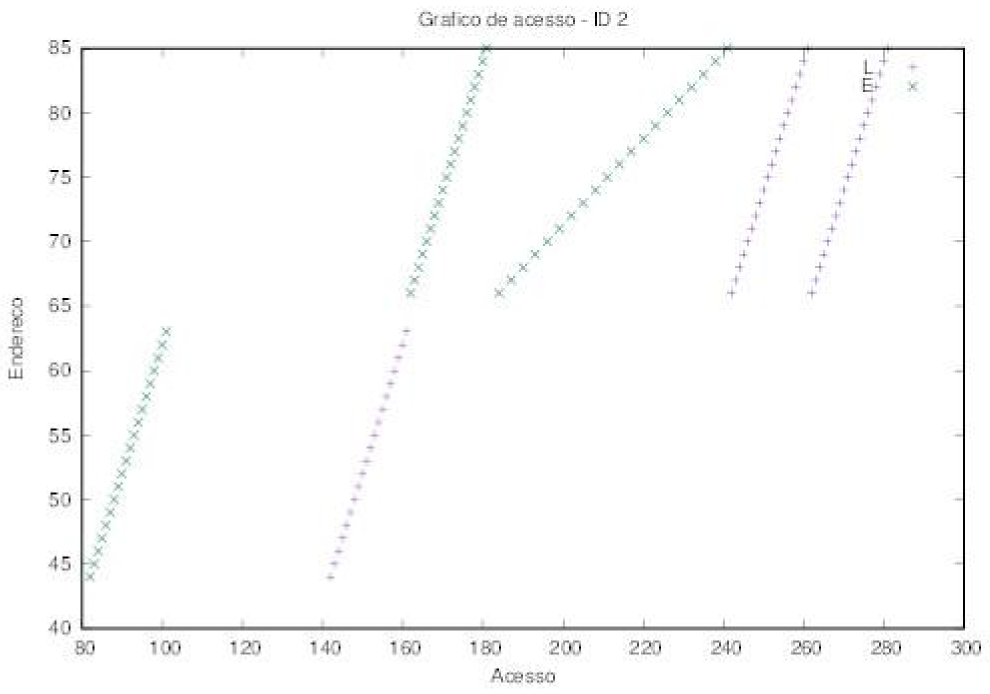


Grafico de acesso - ID 2
Endereco
Acesso
L +
E ×
85
80
75
70
65
60
55
50
45
40
80
100
120
140
160
180
200
220
240
260
280
300

4.1.3. VetorEstatico

4.1.3.1. Produto Interno

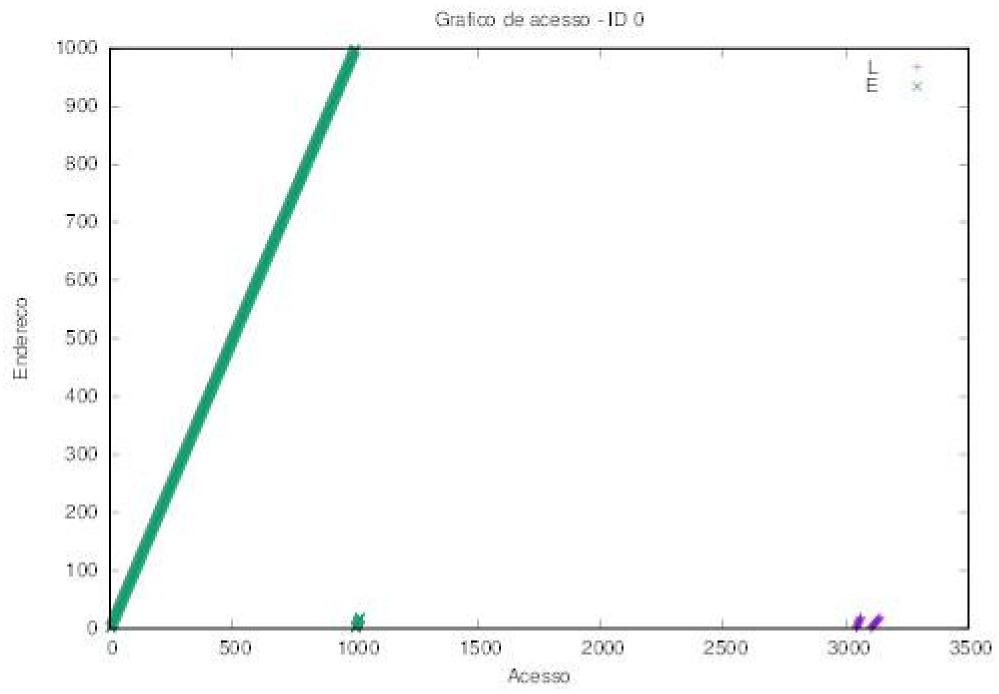


Grafico de acesso - ID 0
L +
E ×
Endereco
Acesso
0
100
200
300
400
500
600
700
800
900
1000
0
500
1000
1500
2000
2500
3000
3500

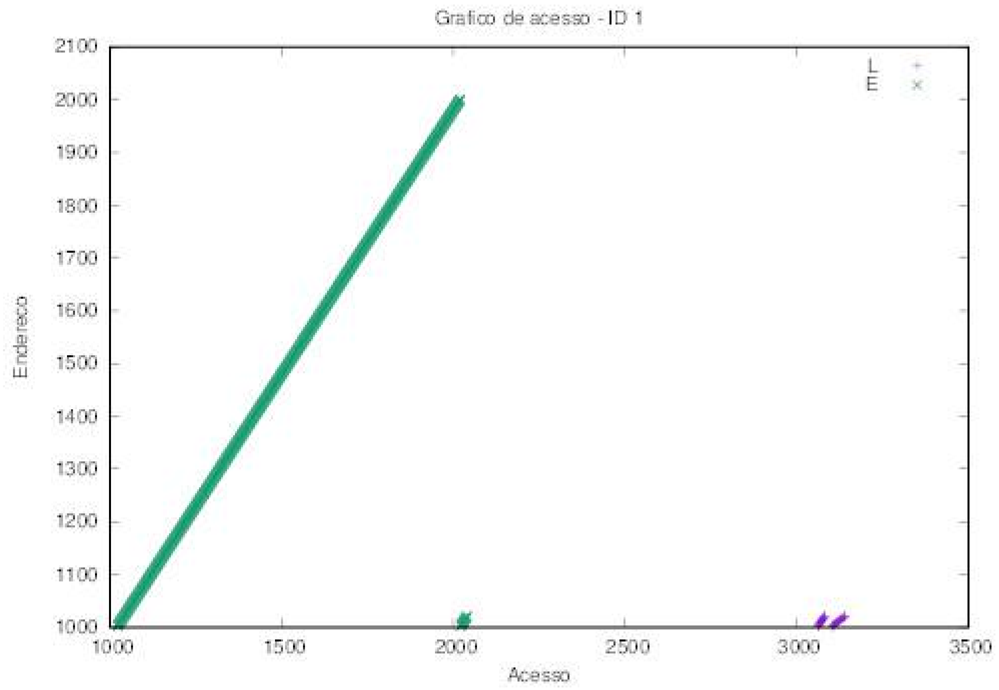


Grafico de acesso - ID 1
L +
E ×
Endereco
Acesso
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
1000
1500
2000
2500
3000
3500

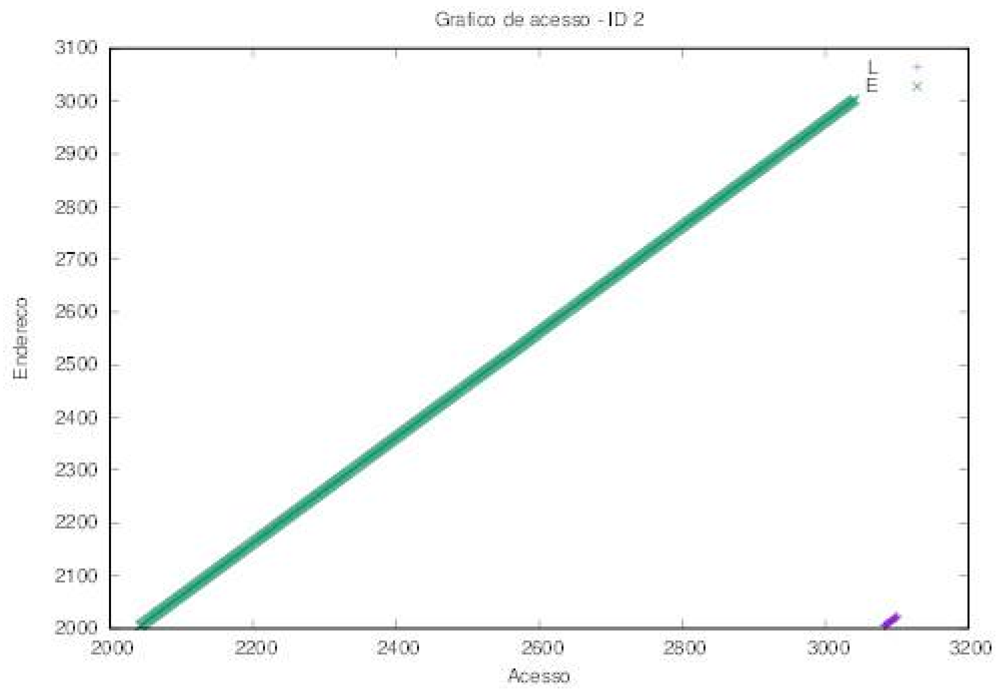


Grafico de acesso - ID 2
L
E
Endereco
Acesso

4.1.3.2. Norma

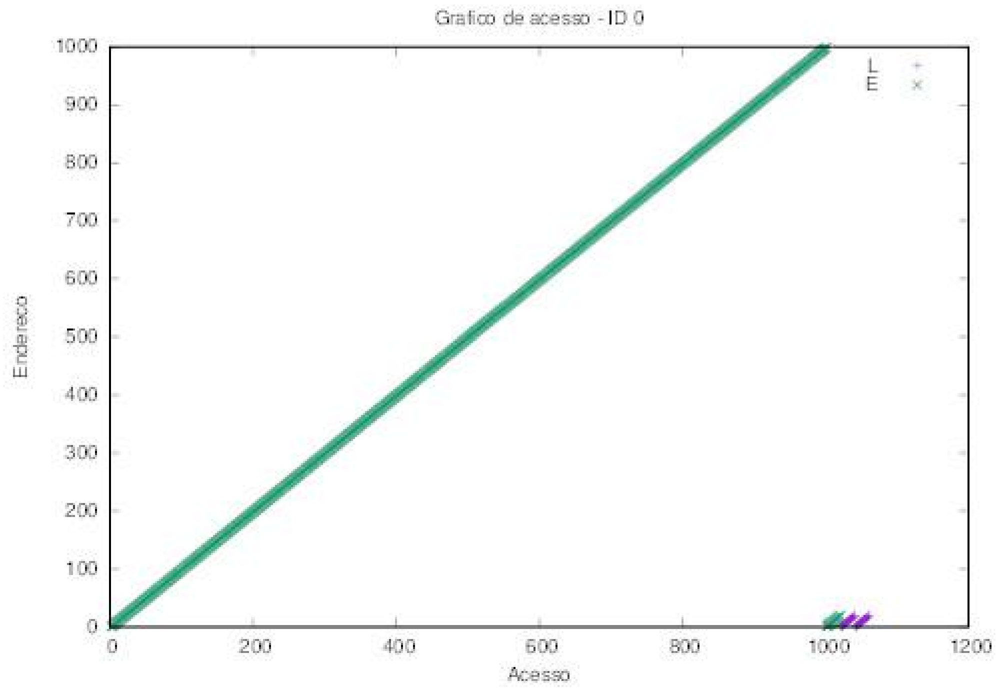


Grafico de acesso - ID 0
L
E
Endereco
Acesso

4.1.3.3. Soma

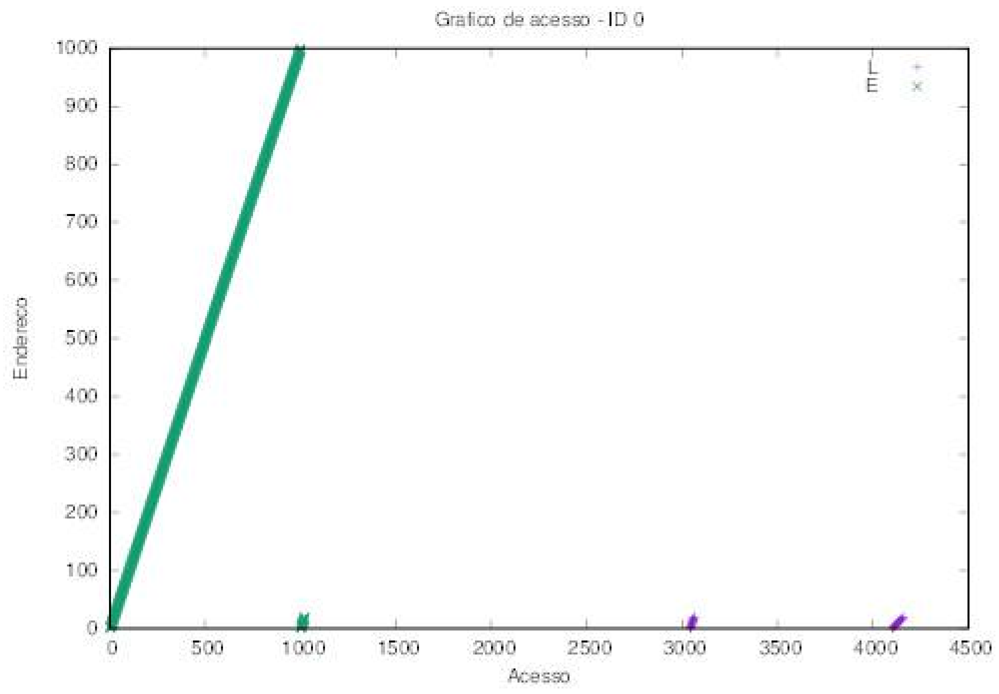


Grafico de acesso - ID 0
Endereco
Acesso
L +
E ×

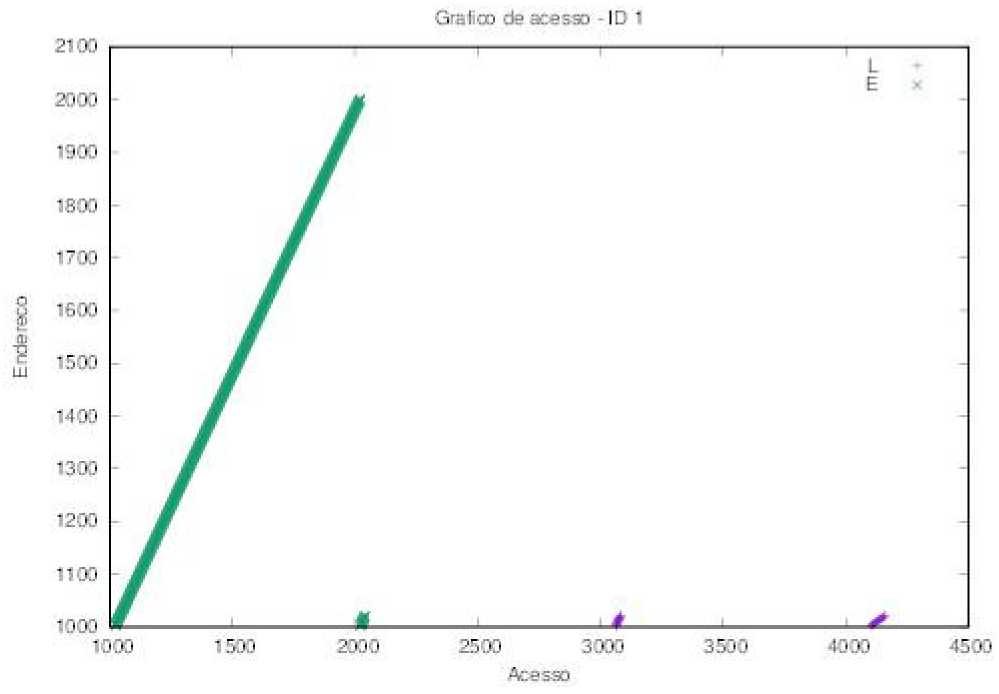


Grafico de acesso - ID 1
Endereco
Acesso
L +
E ×

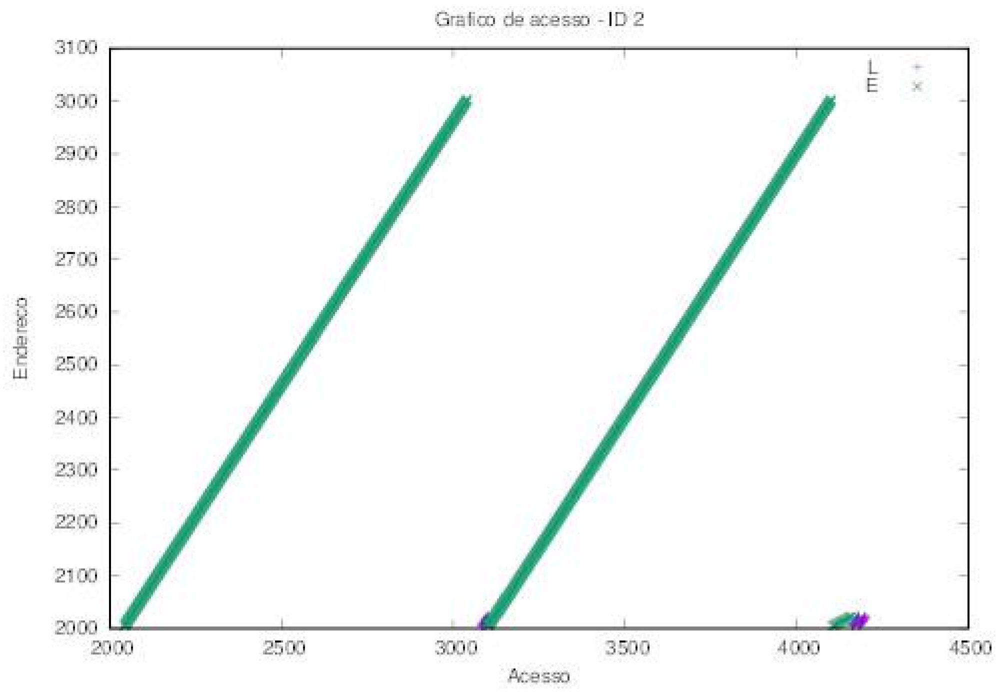
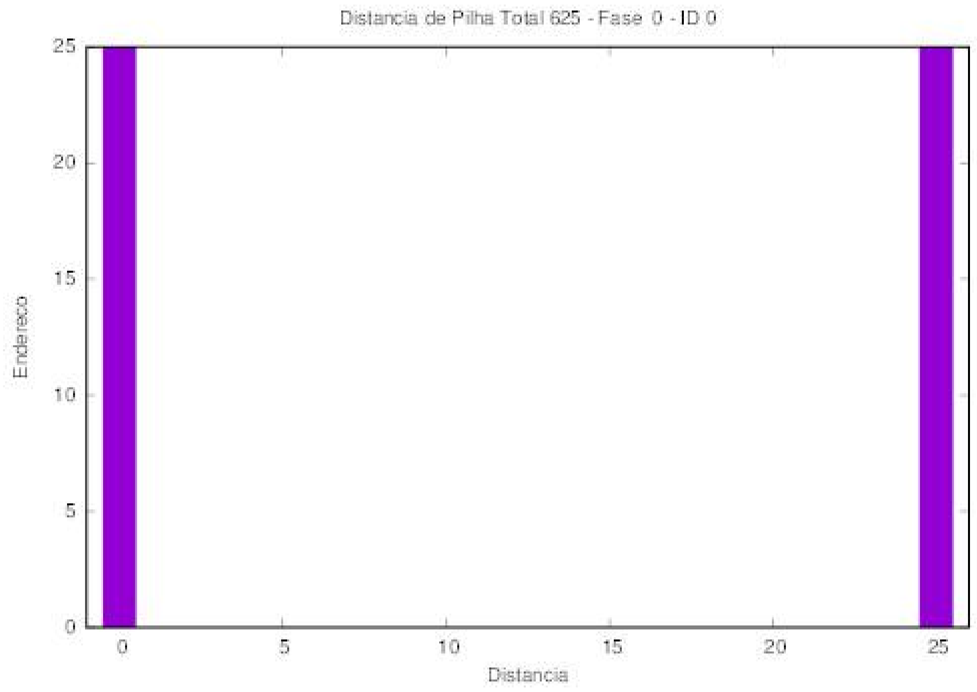


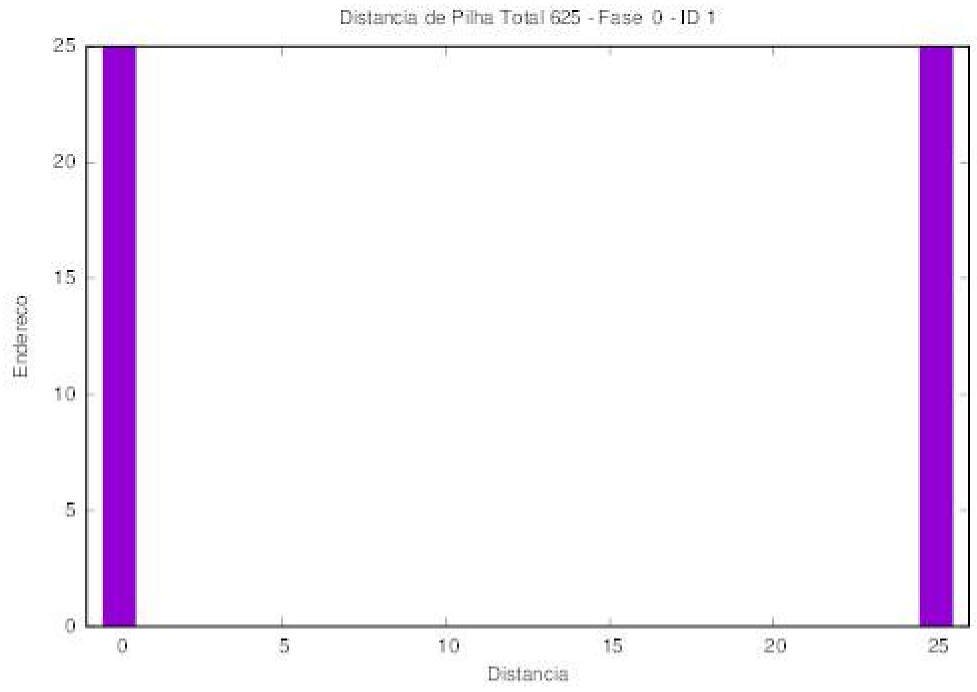
Grafico de acesso - ID 2
L
E
Endereco
Acesso
3100
3000
2900
2800
2700
2600
2500
2400
2300
2200
2100
2000
2000
2500
3000
3500
4000
4500

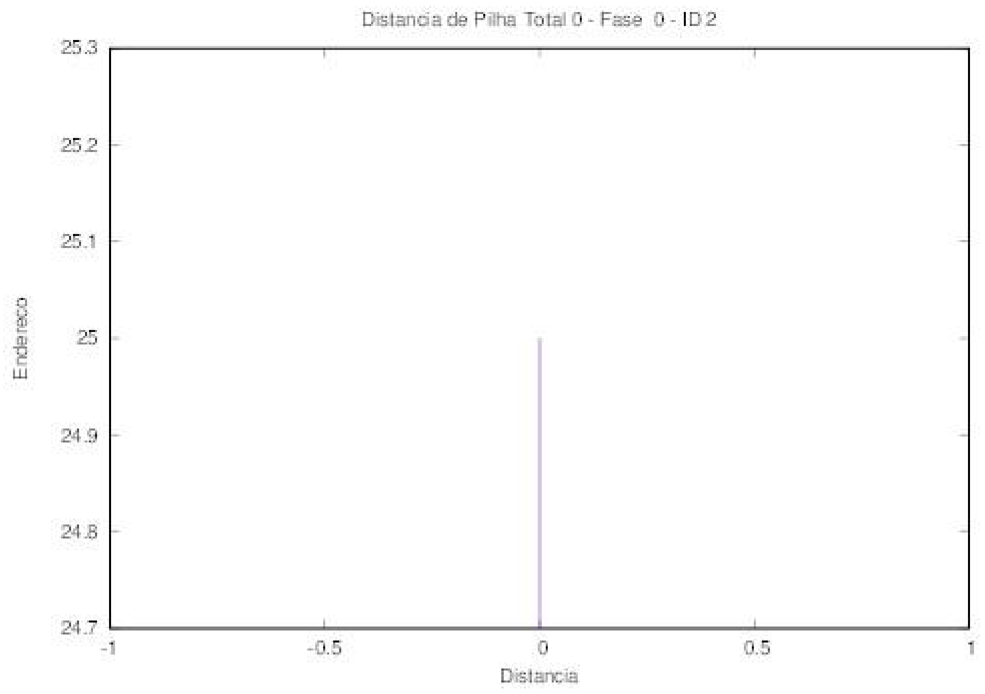
4.2. Distância de pilha

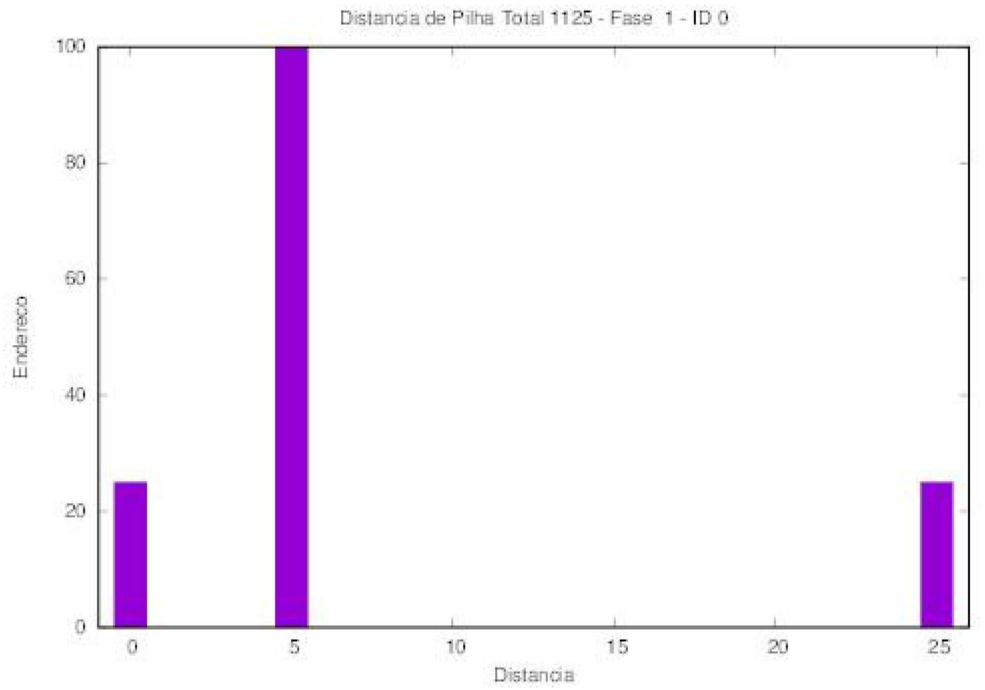
4.2.1. AnaliseMemoria

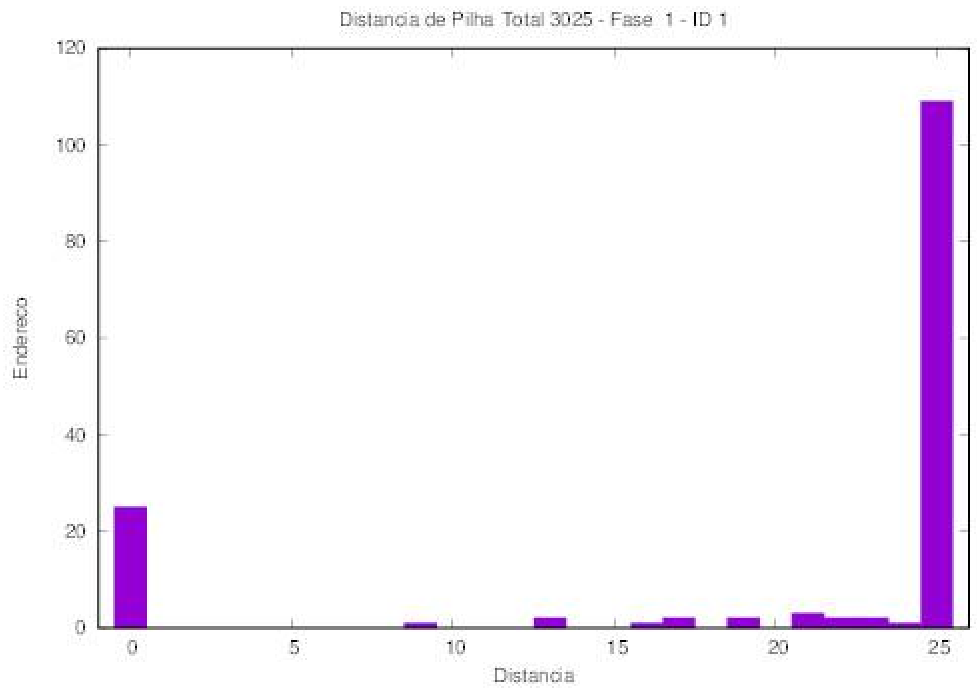
4.2.1.1. Multiplicação

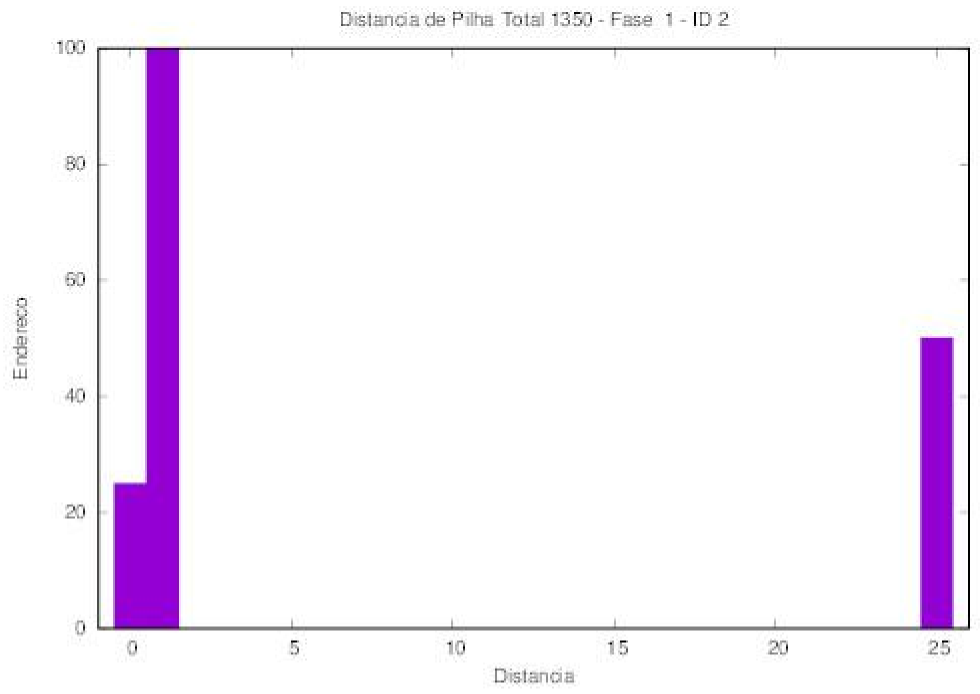


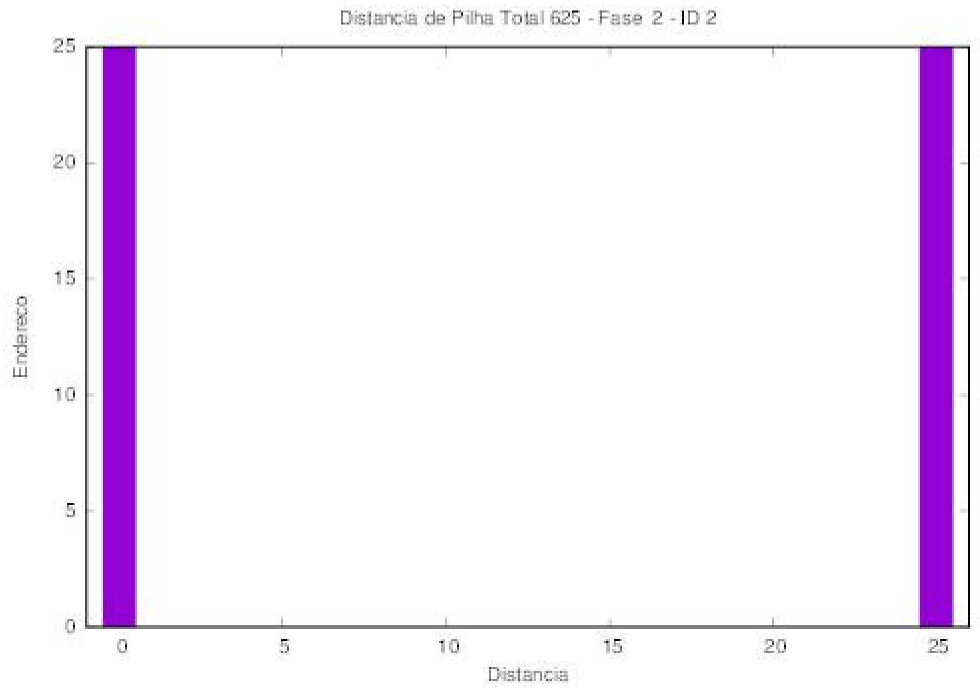



Distancia de Pilha Total 0 - Fase 0 - ID 2
Endereco
Distancia
25.3
25.2
25.1
25
24.9
24.8
24.7
-1
-0.5
0
0.5
1

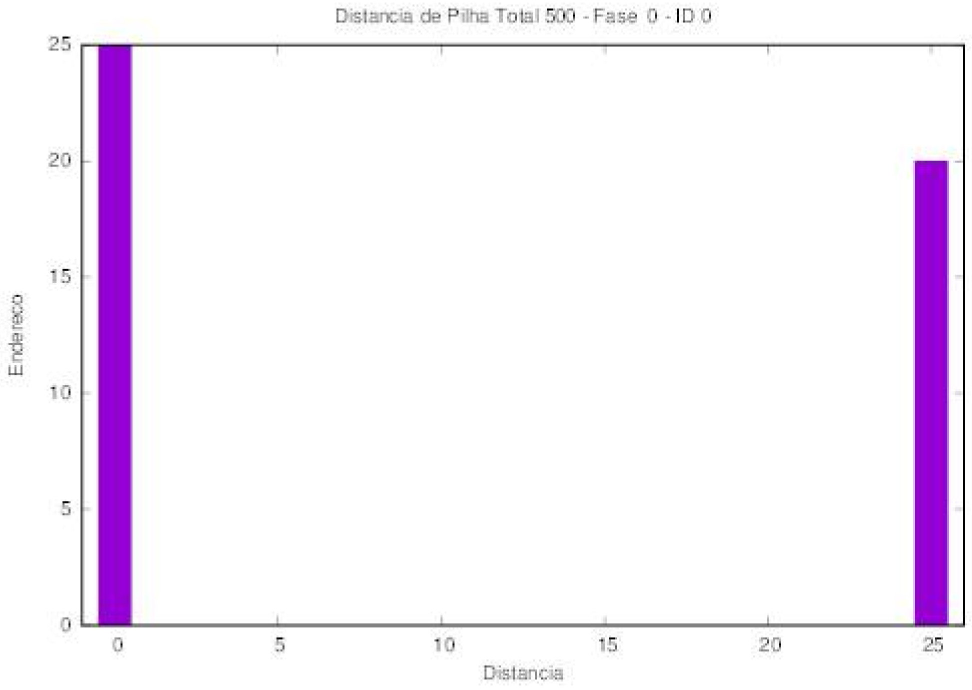

Distancia de Pilha Total 1125 - Fase 1 - ID 0
Endereco
Distancia
100
80
60
40
20
0
0
5
10
15
20
25


Distancia de Pilha Total 3025 - Fase 1 - ID 1
Endereco
Distancia
0
20
40
60
80
100
120
0
5
10
15
20
25

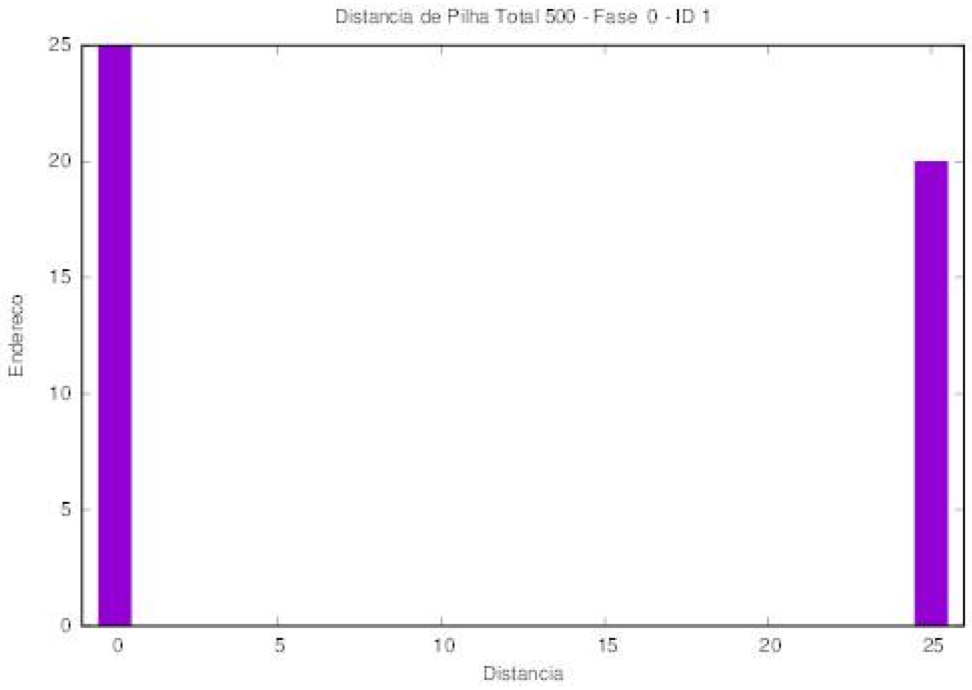

Distancia de Pilha Total 1350 - Fase 1 - ID 2
Endereco
Distancia
0
20
40
60
80
100
0
5
10
15
20
25


Distancia de Pilha Total 625 - Fase 2 - ID 2
Endereco
Distancia
0
5
10
15
20
25
0
5
10
15
20
25

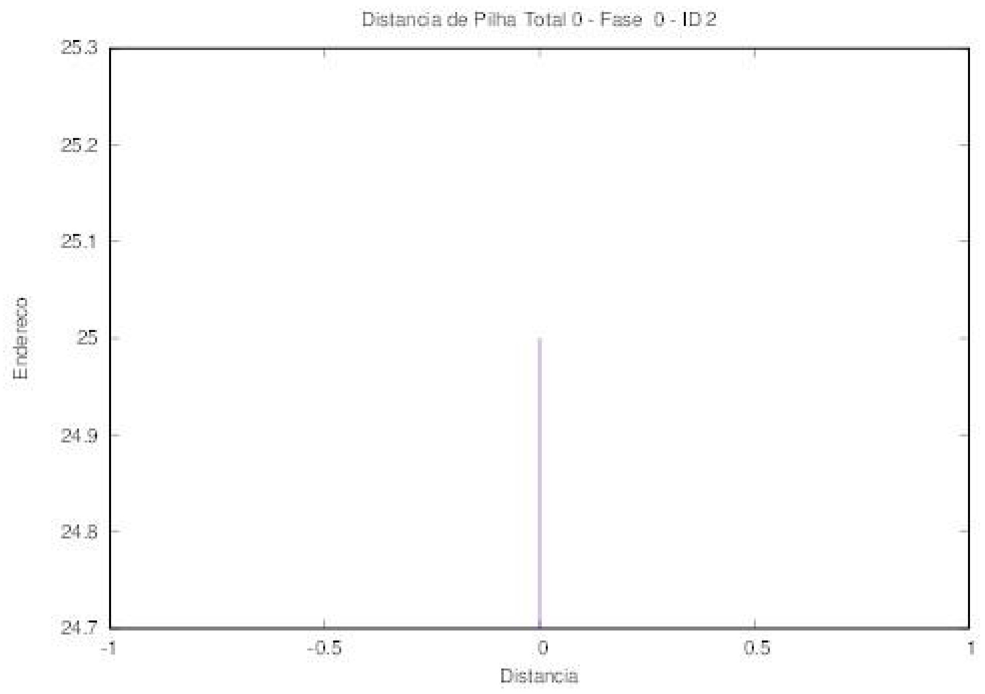
4.2.1.2. Soma

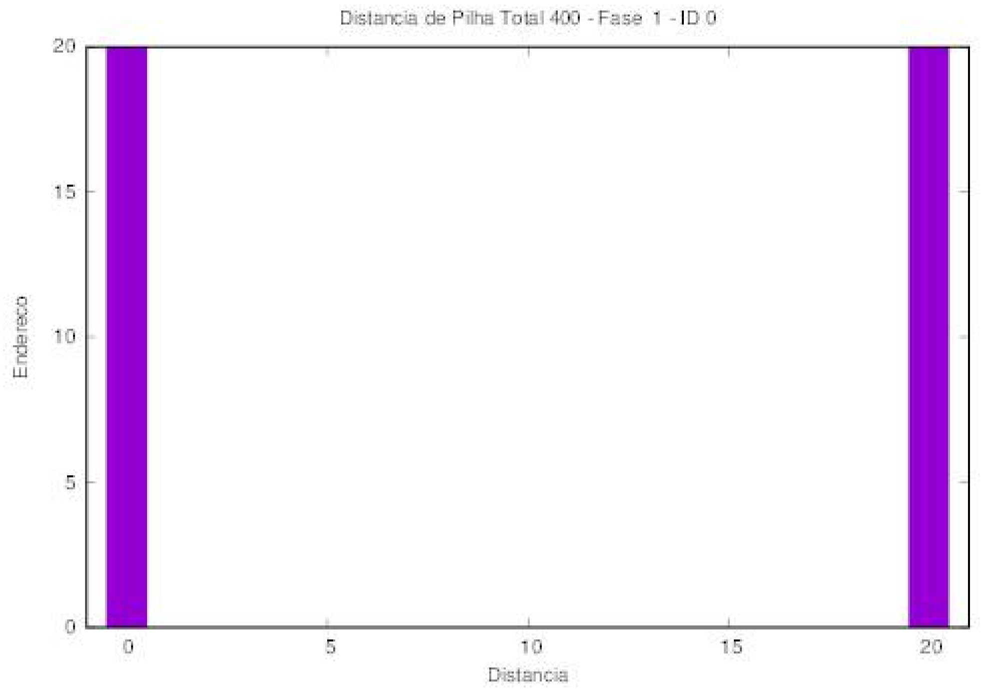


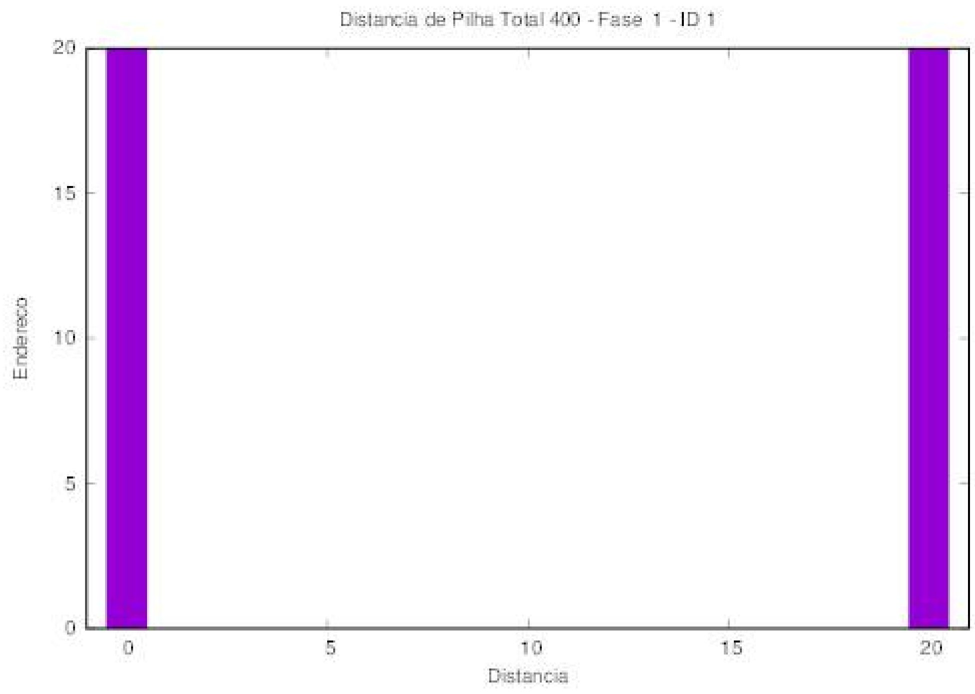
Distancia de Pilha Total 500 - Fase 0 - ID 0
Endereco
Distancia
0
5
10
15
20
25

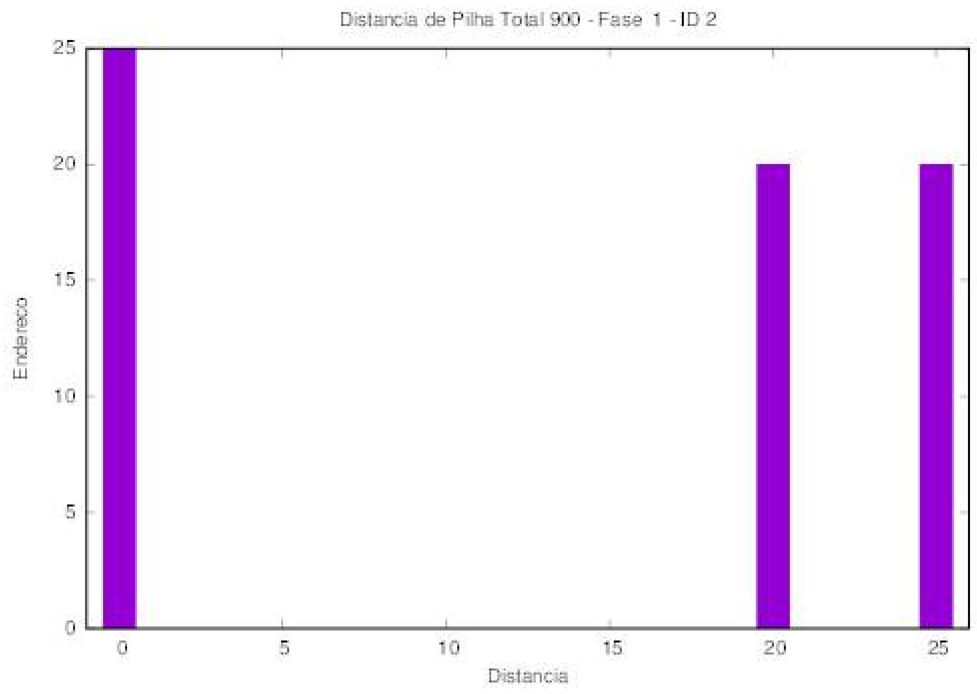


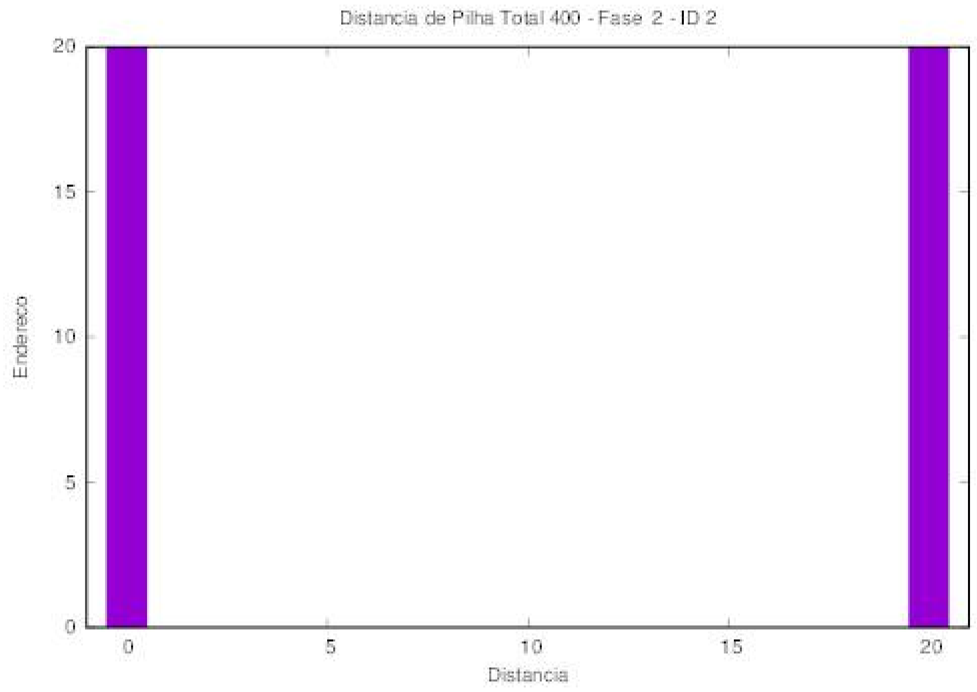
Distancia de Pilha Total 500 - Fase 0 - ID 1
Endereco
Distancia
0
5
10
15
20
25


Distancia de Pilha Total 0 - Fase 0 - ID 2
Endereco
25.3
25.2
25.1
25
24.9
24.8
24.7
-1
-0.5
0
0.5
1
Distancia

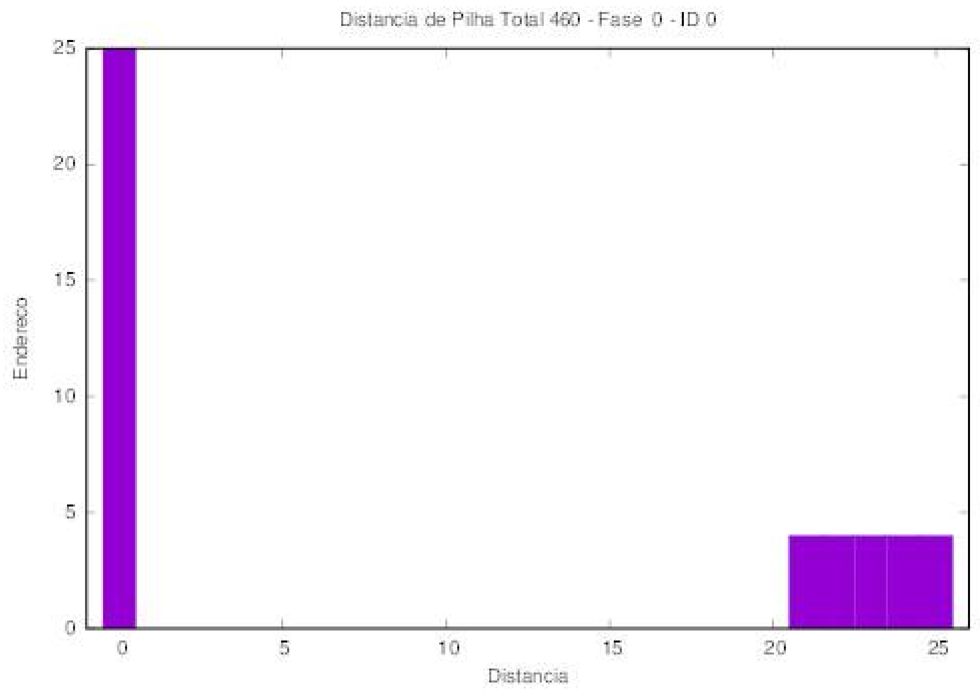

Distancia de Pilha Total 400 - Fase 1 - ID 0
Endereco
20
15
10
5
0
0
5
10
15
20
Distancia

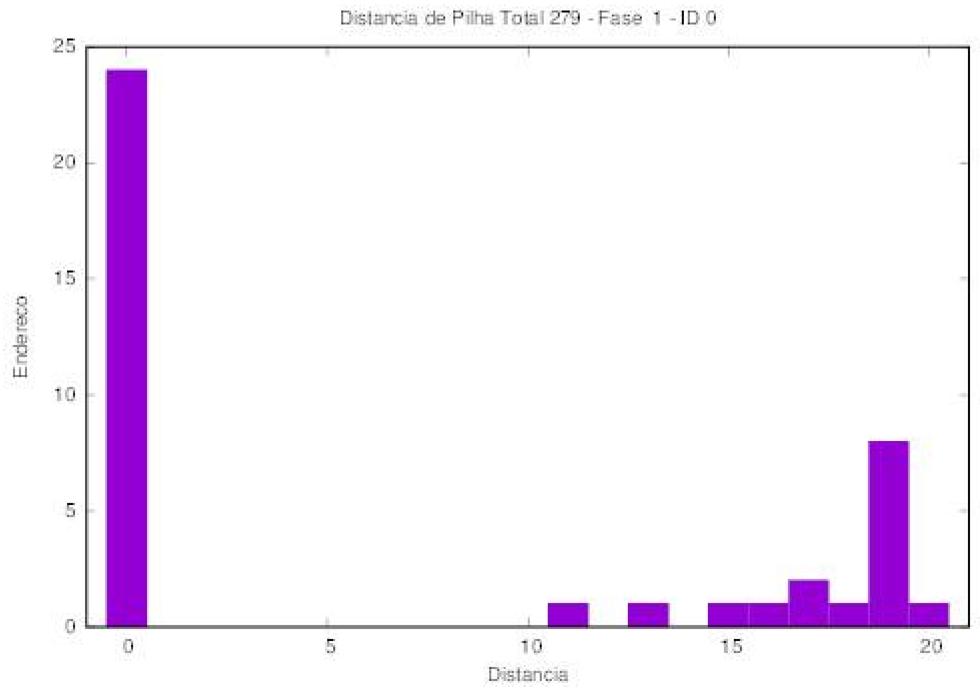

Distancia de Pilha Total 400 - Fase 1 - ID 1
Endereco
Distancia
0
5
10
15
20
0
5
10
15
20

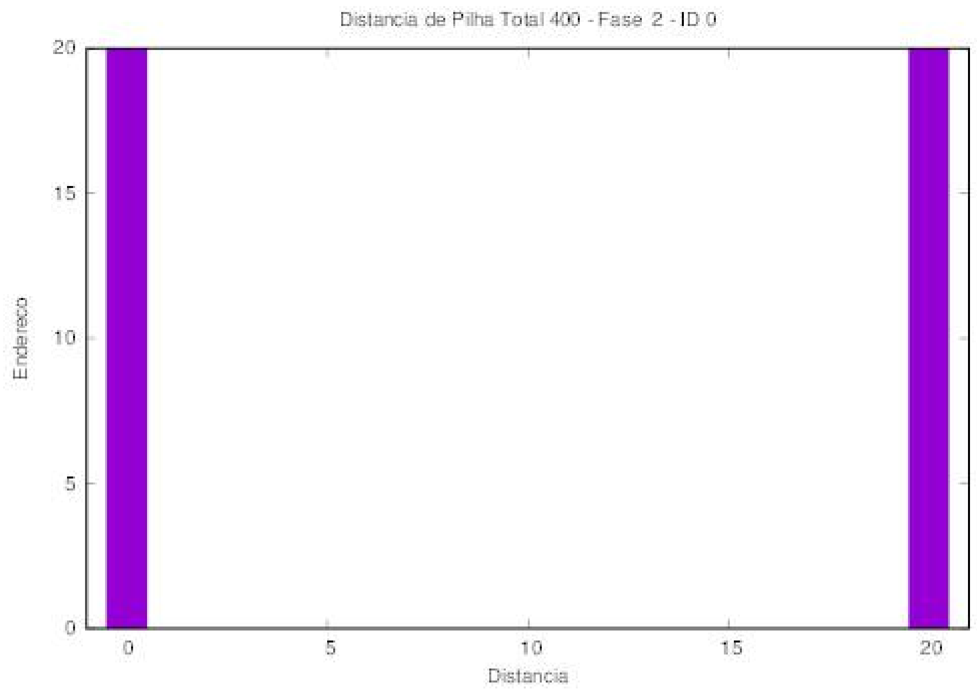

Distancia de Pilha Total 900 - Fase 1 - ID 2
Endereco
Distancia
0
5
10
15
20
25


Distancia de Pilha Total 400 - Fase 2 - ID 2
Endereco
Distancia
0
5
10
15
20

4.2.1.3. Transposto

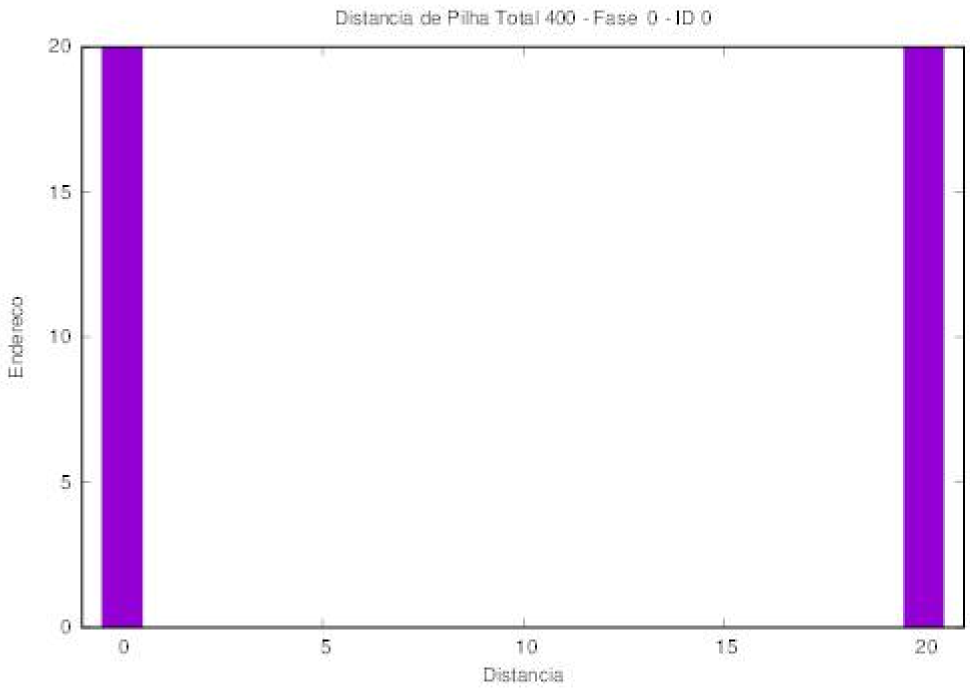

Distancia de Pilha Total 460 - Fase 0 - ID 0
Endereco
Distancia
0
5
10
15
20
25

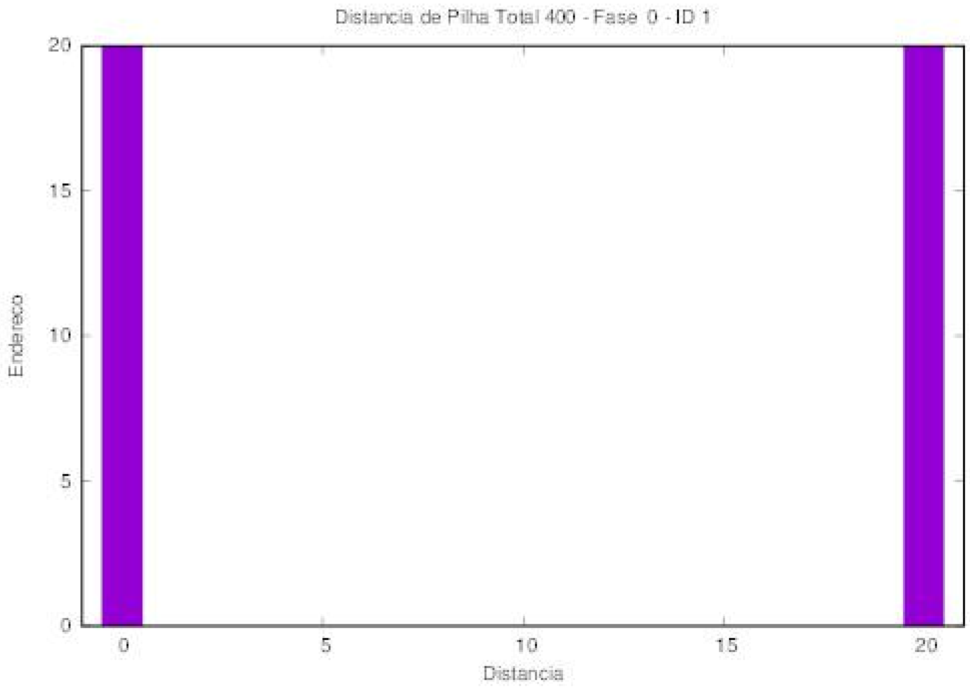

Distancia de Pilha Total 279 - Fase 1 - ID 0
Endereco
Distancia
0
5
10
15
20
25

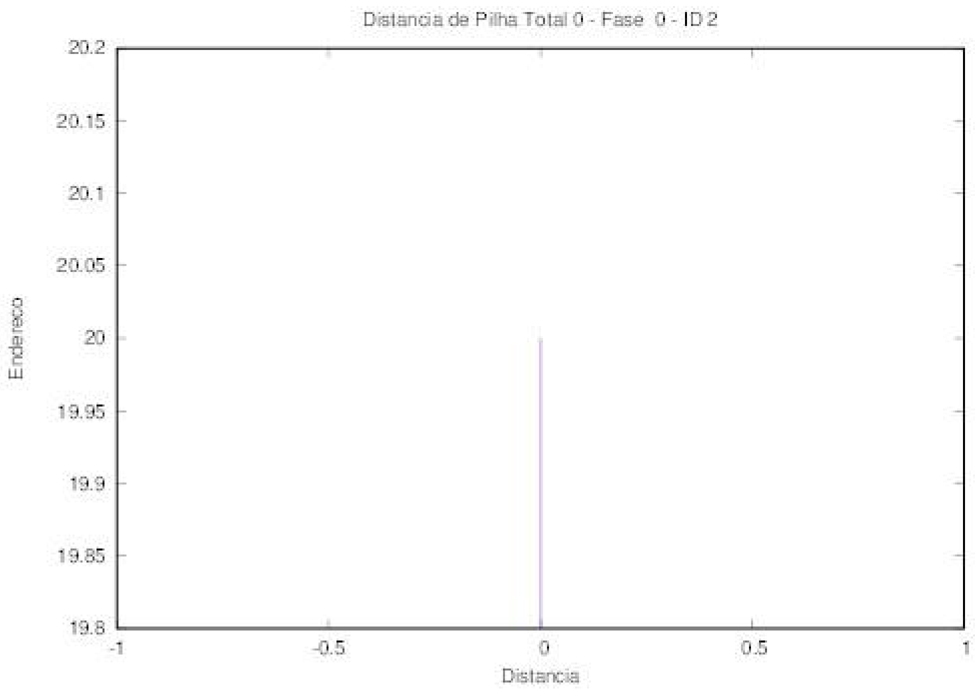

Distancia de Pilha Total 400 - Fase 2 - ID 0
Endereco
Distancia
0
5
10
15
20
0
5
10
15
20

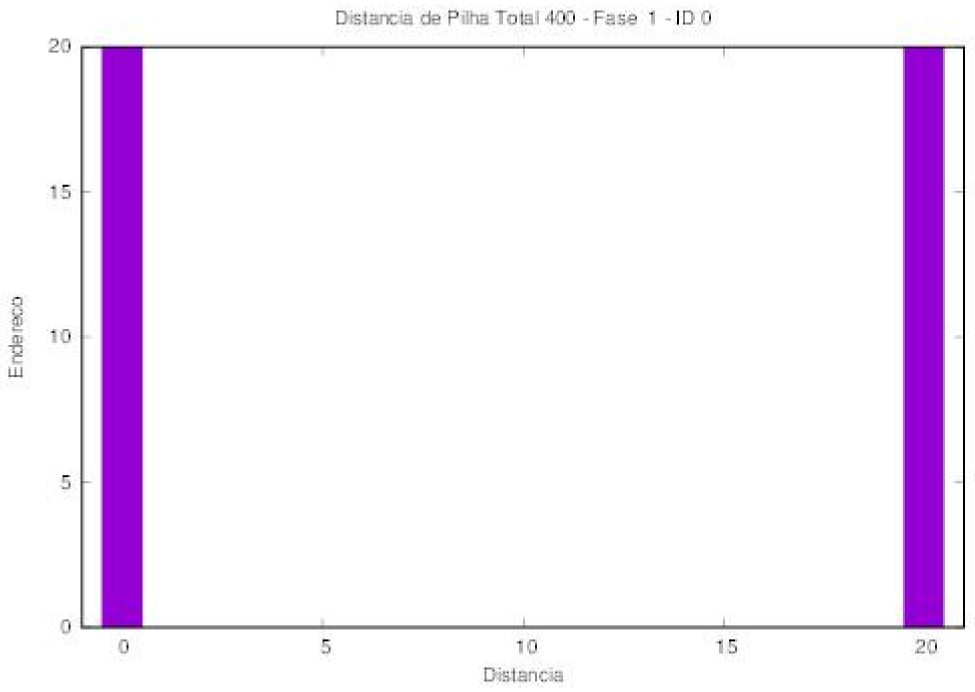
4.2.2. VetorDinamico

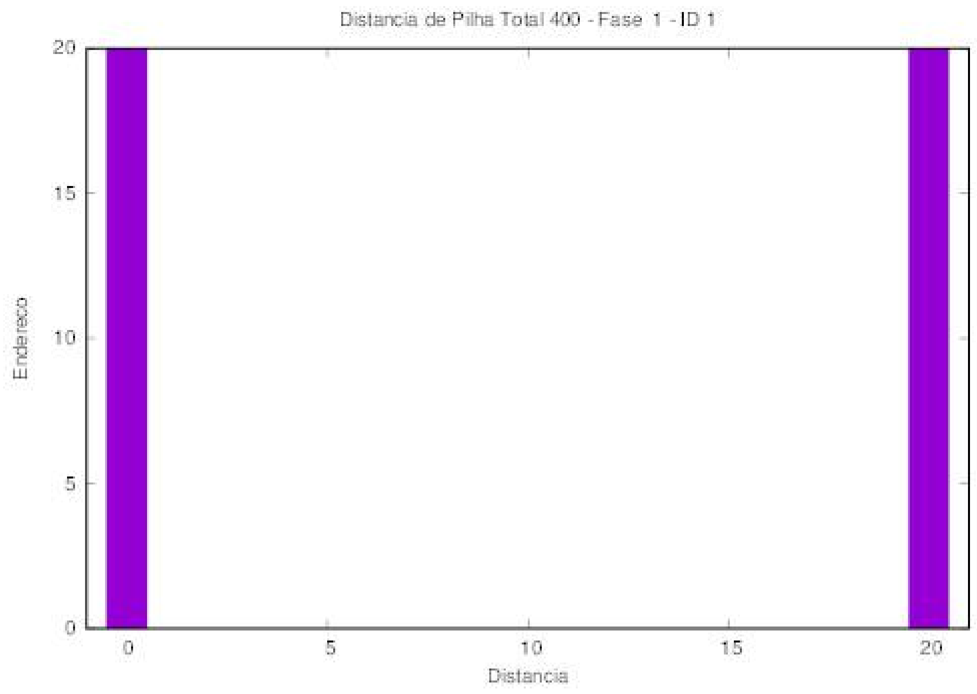
4.2.2.1. Produto Interno

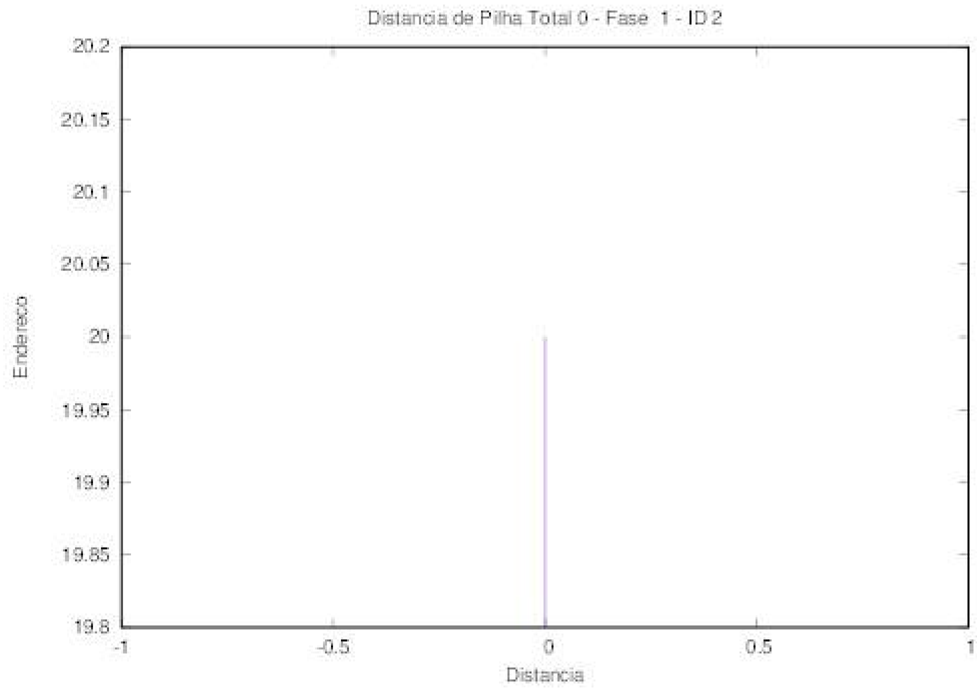




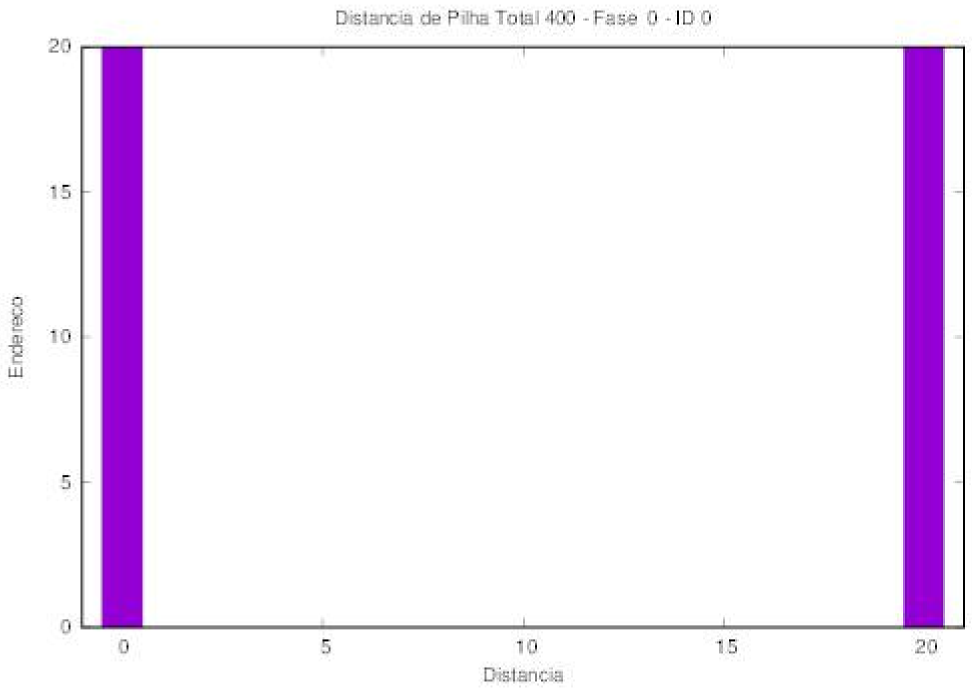

Distancia de Pilha Total 0 - Fase 0 - ID 2
Endereco
Distancia
20.2
20.15
20.1
20.05
20
19.95
19.9
19.85
19.8
-1
-0.5
0
0.5
1

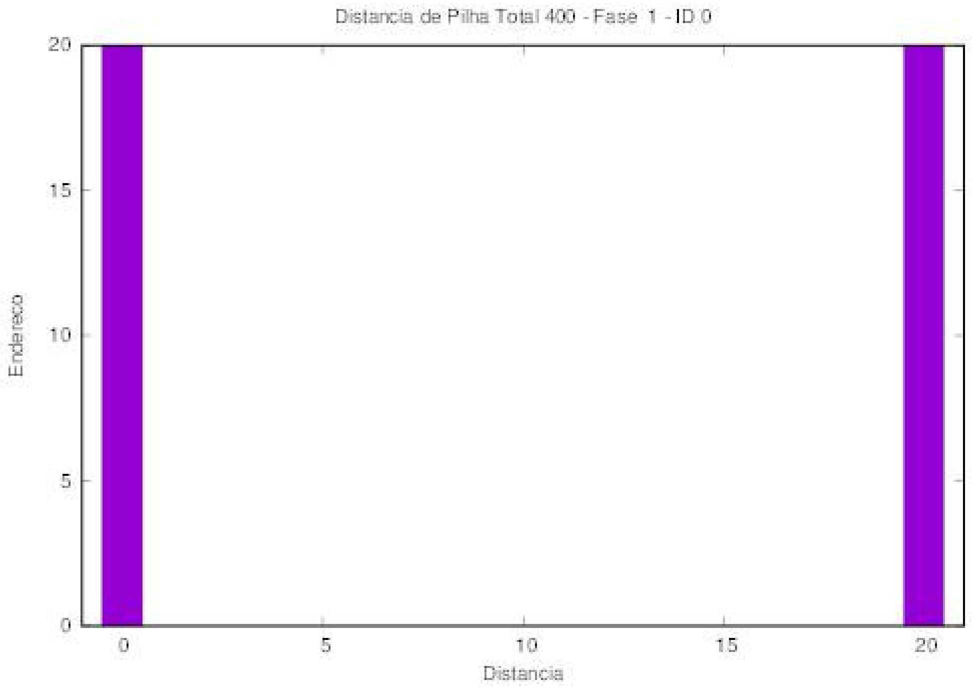

Distancia de Pilha Total 400 - Fase 1 - ID 0
Endereco
Distancia
20
15
10
5
0
0
5
10
15
20


Distancia de Pilha Total 400 - Fase 1 - ID 1
Endereco
Distancia
0
5
10
15
20

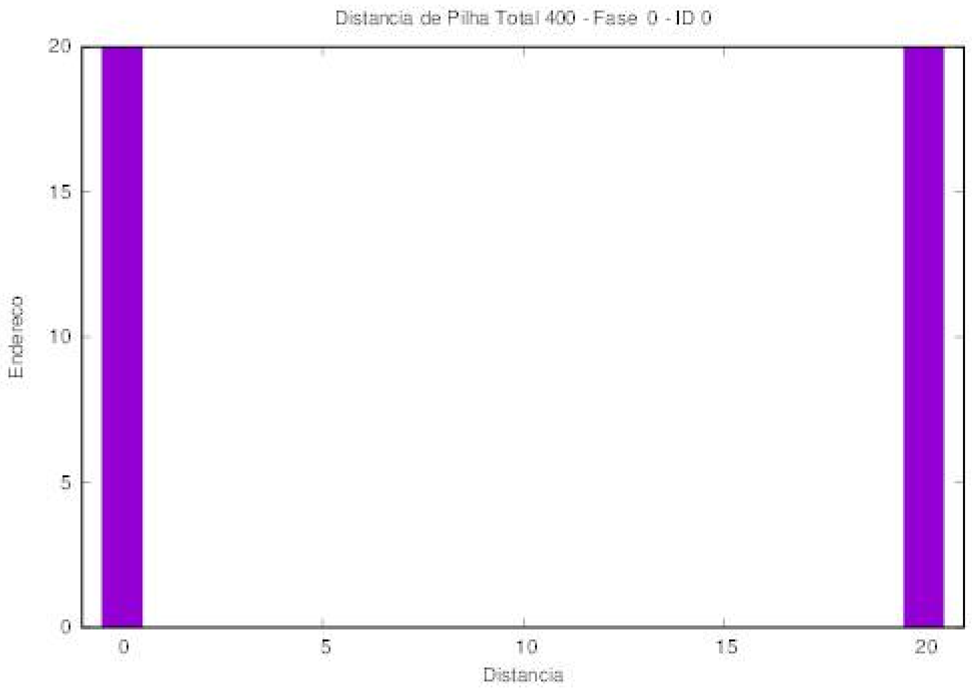

Distancia de Pilha Total 0 - Fase 1 - ID 2
Endereco
Distancia
19.8
19.85
19.9
19.95
20
20.05
20.1
20.15
20.2
-1
-0.5
0
0.5
1

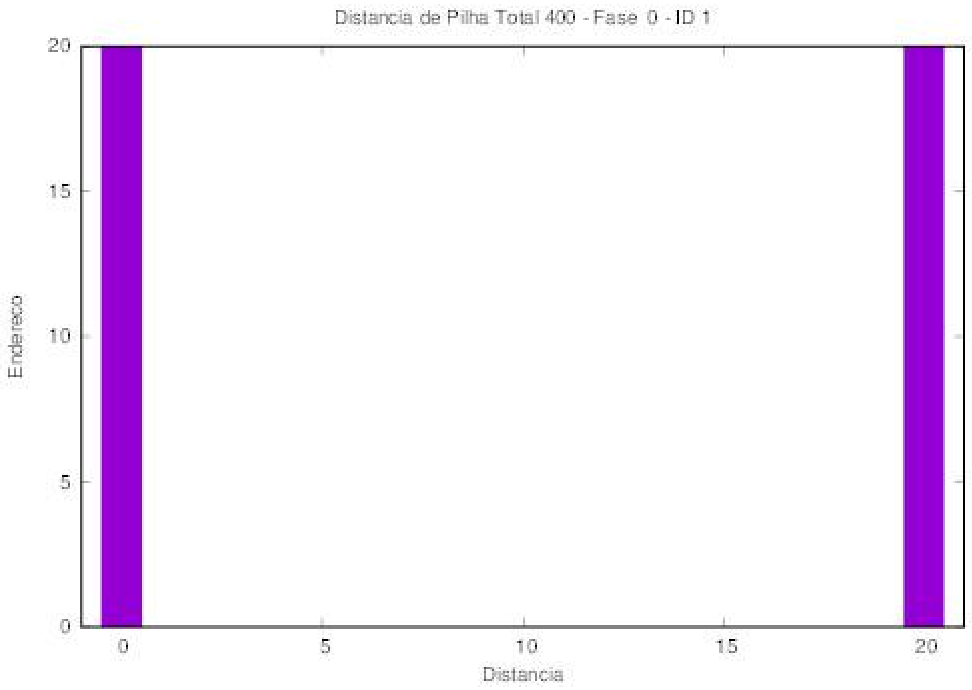
4.2.2.2. Norma

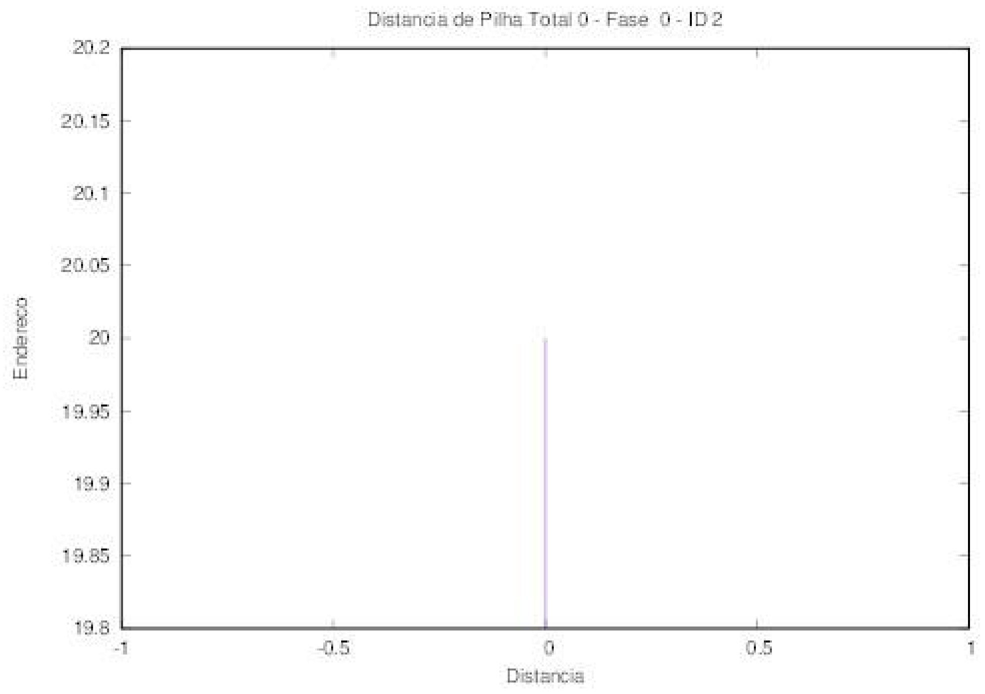

Distancia de Pilha Total 400 - Fase 0 - ID 0
Endereco
Distancia
0
5
10
15
20

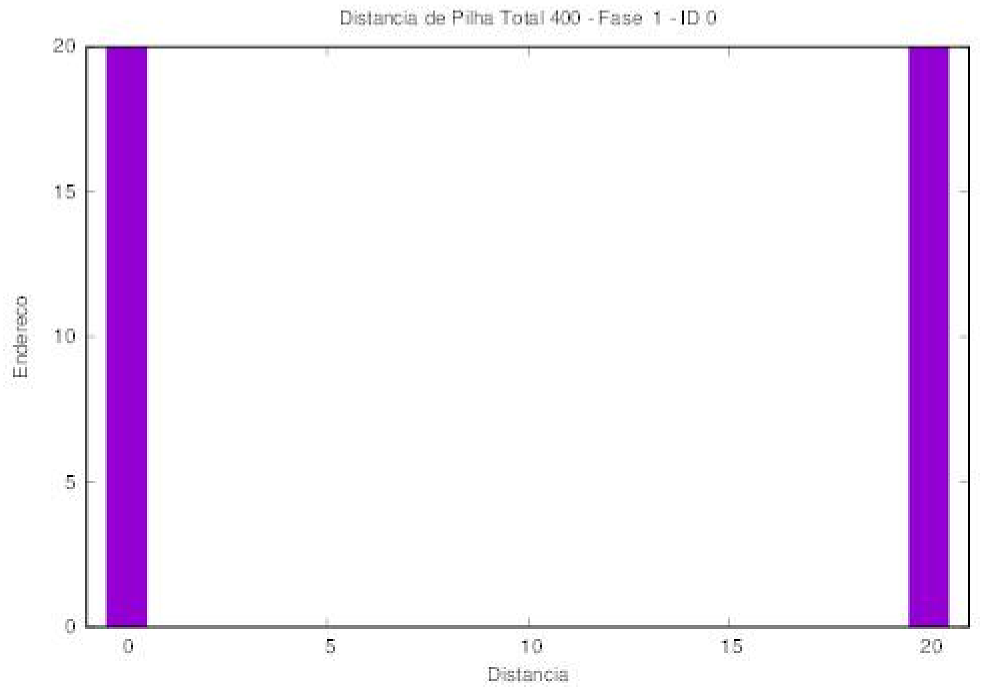

Distancia de Pilha Total 400 - Fase 1 - ID 0
Endereco
Distancia
0
5
10
15
20

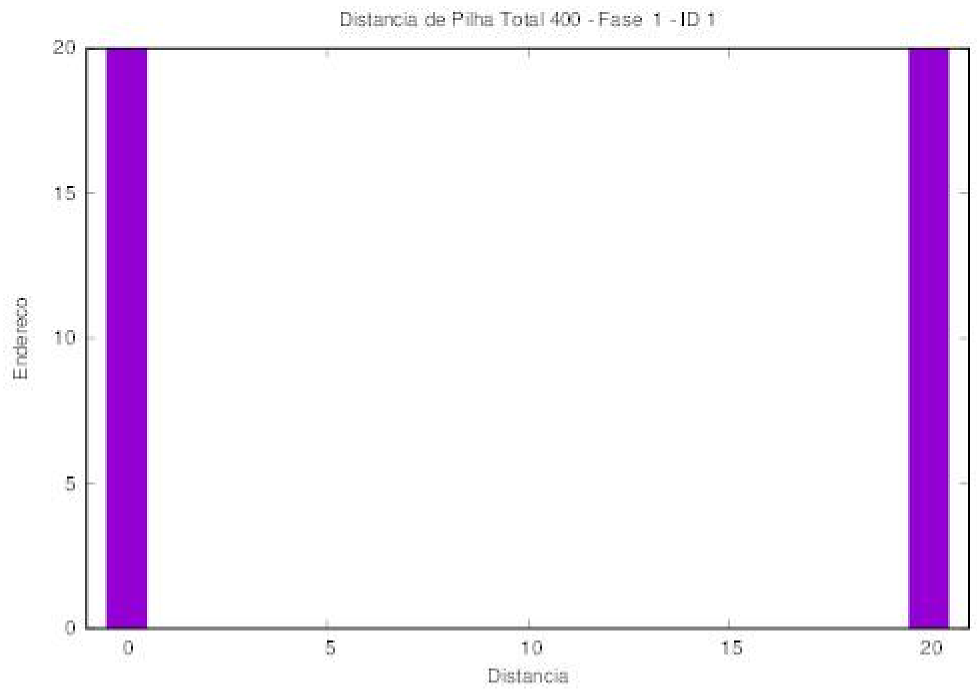
4.2.2.3. Soma


Distancia de Pilha Total 400 - Fase 0 - ID 0
Endereco
Distancia
0
5
10
15
20

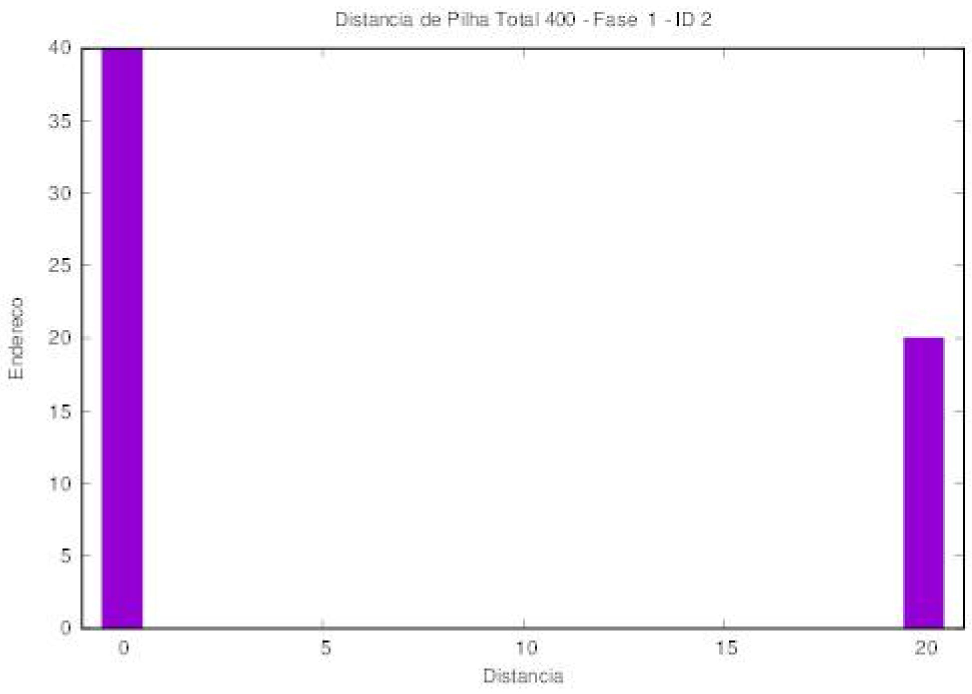

Distancia de Pilha Total 400 - Fase 0 - ID 1
Endereco
Distancia
0
5
10
15
20

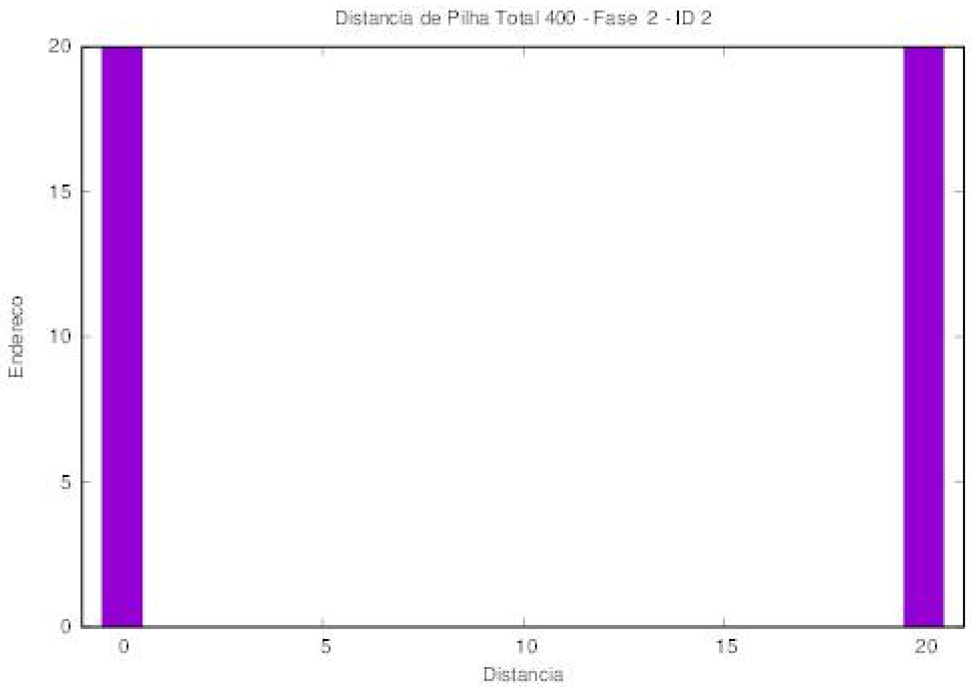

Distancia de Pilha Total 0 - Fase 0 - ID 2
Endereco
Distancia
20.2
20.15
20.1
20.05
20
19.95
19.9
19.85
19.8
-1
-0.5
0
0.5
1


Distancia de Pilha Total 400 - Fase 1 - ID 0
Endereco
Distancia
20
15
10
5
0
0
5
10
15
20



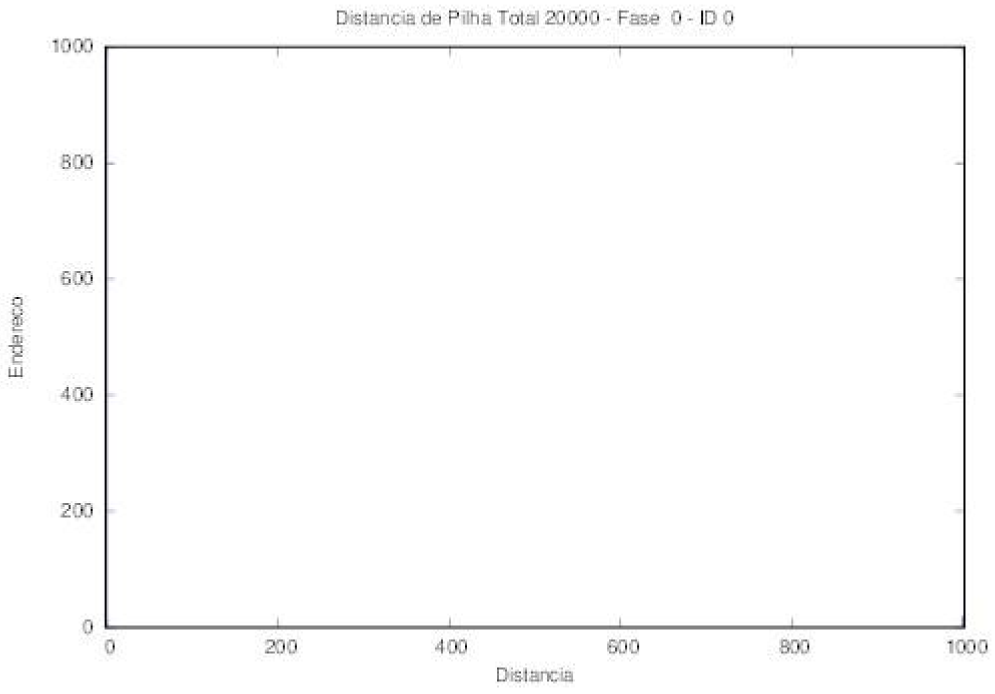
Distancia de Pilha Total 400 - Fase 1 - ID 1
Endereco
Distancia
0
5
10
15
20

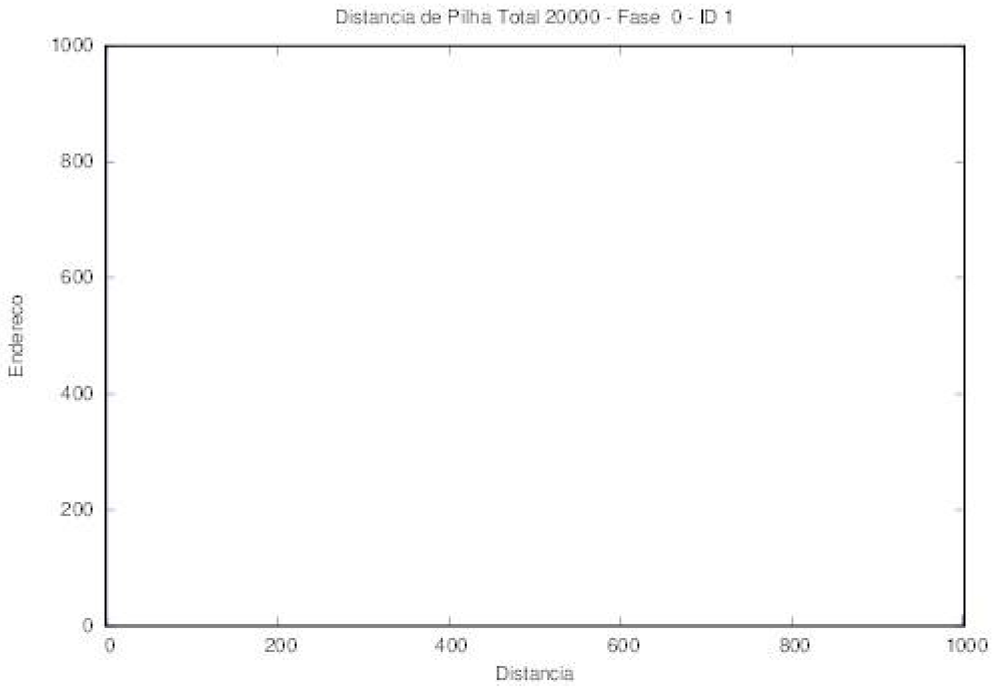

Distancia de Pilha Total 400 - Fase 1 - ID 2
Endereco
Distancia
0
5
10
15
20
25
30
35
40

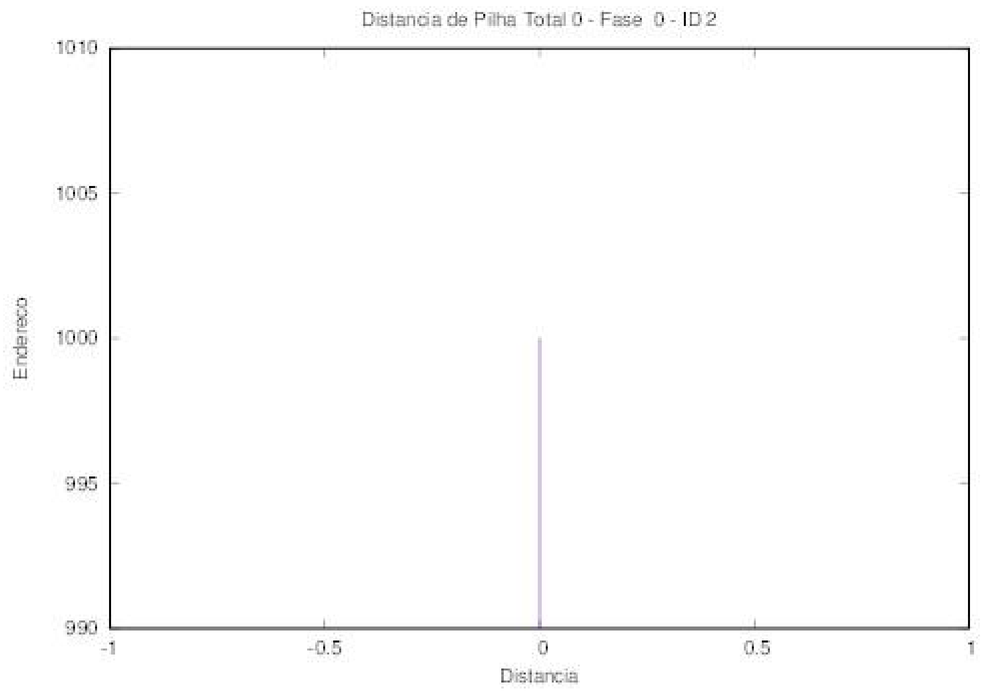

Distancia de Pilha Total 400 - Fase 2 - ID 2
Endereco
Distancia
0
5
10
15
20

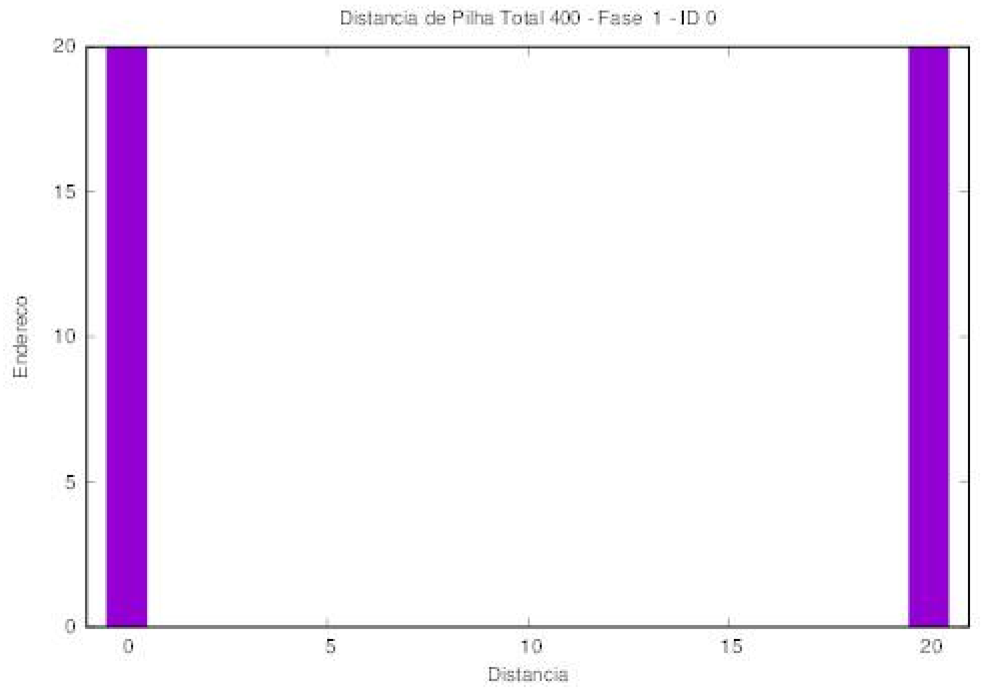
4.2.3. VetorEstatico

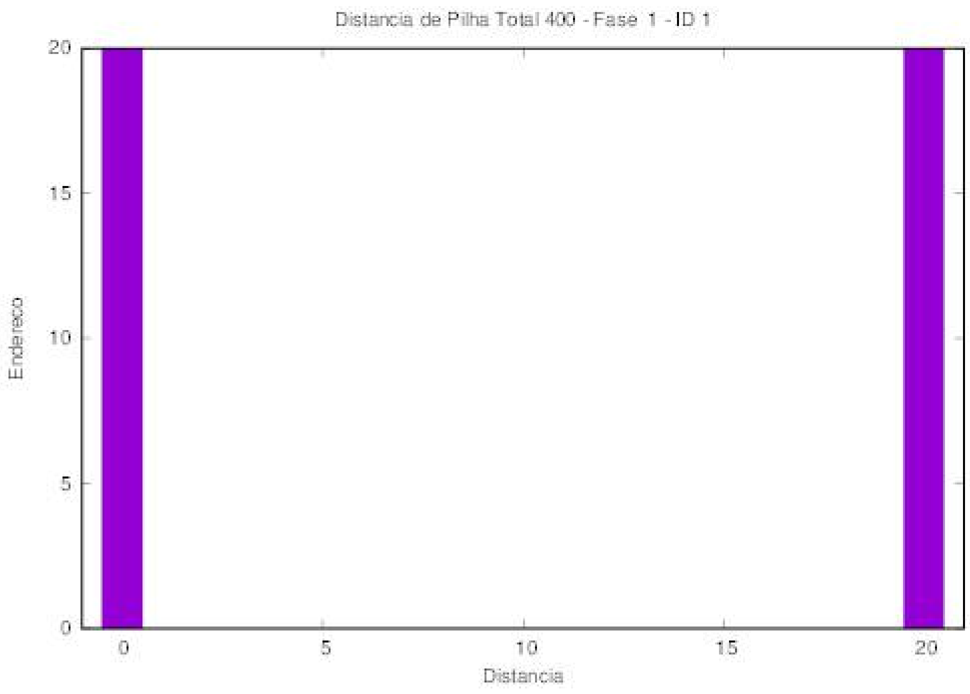
4.2.3.1. Produto Interno

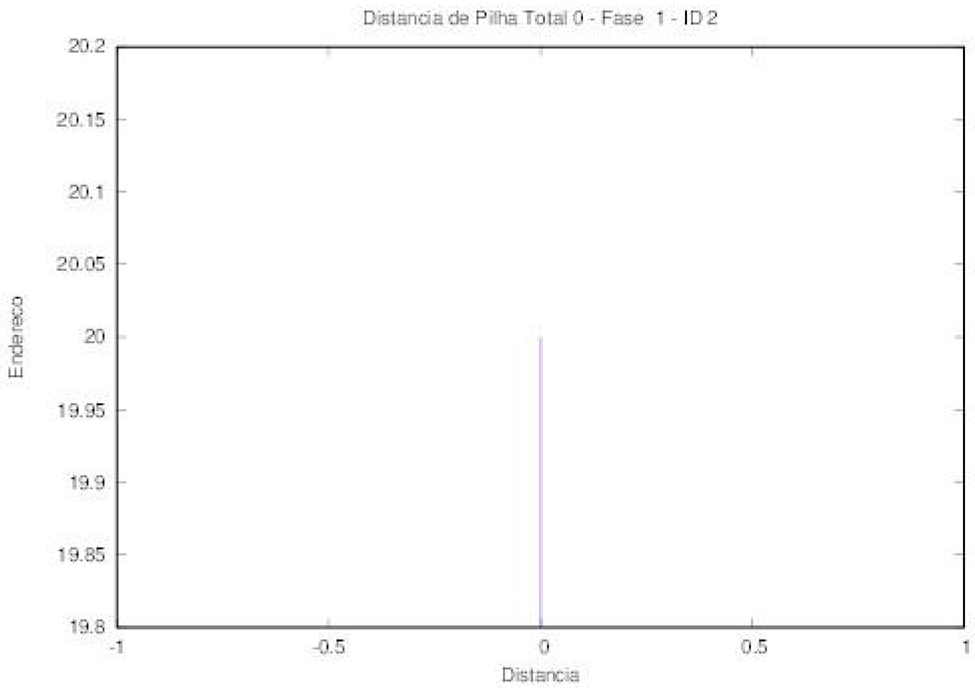




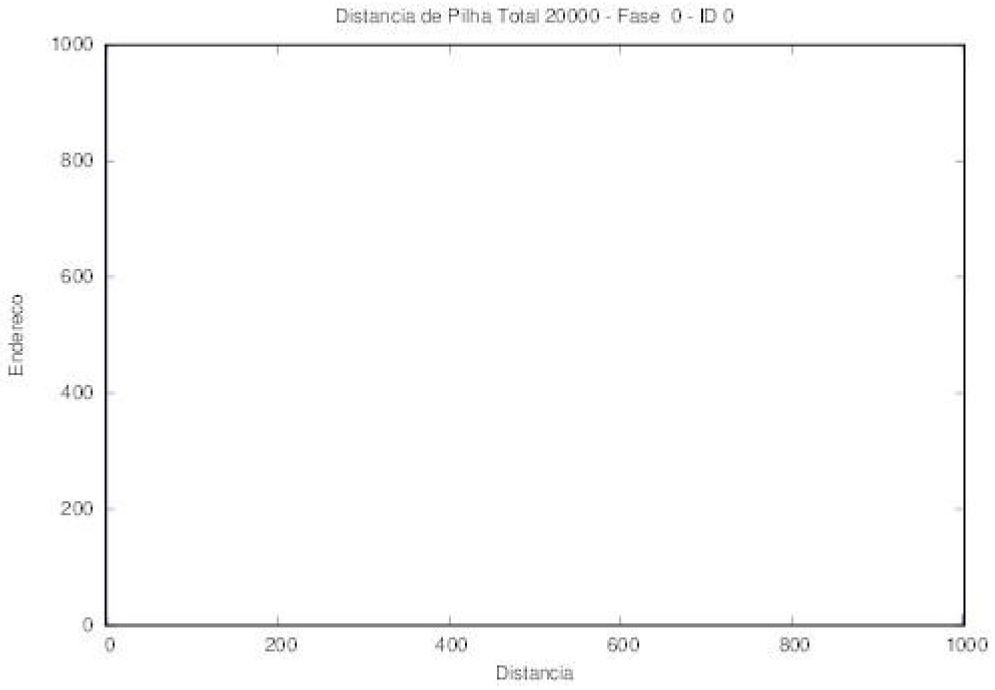

Distancia de Pilha Total 0 - Fase 0 - ID 2
1010
1005
1000
995
990
Endereco
-1
-0.5
0
0.5
1
Distancia


Distancia de Pilha Total 400 - Fase 1 - ID 0
20
15
10
5
0
Endereco
0
5
10
15
20
Distancia

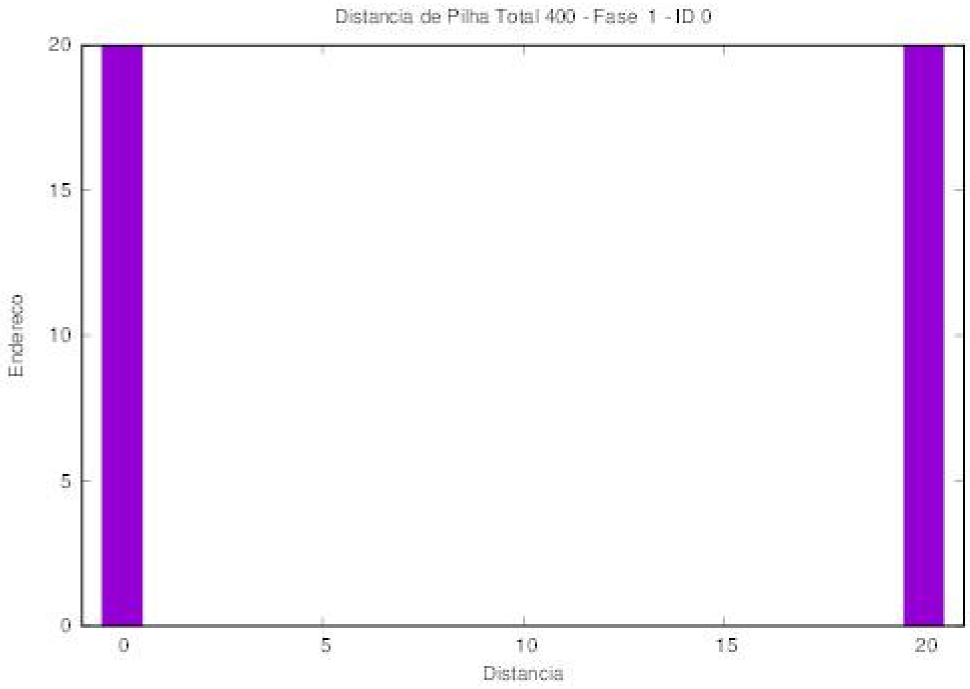

Distancia de Pilha Total 400 - Fase 1 - ID 1
Endereco
Distancia
0
5
10
15
20


Distancia de Pilha Total 0 - Fase 1 - ID 2
Endereco
Distancia
19.8
19.85
19.9
19.95
20
20.05
20.1
20.15
20.2
-1
-0.5
0
0.5
1

4.2.3.2. Norma

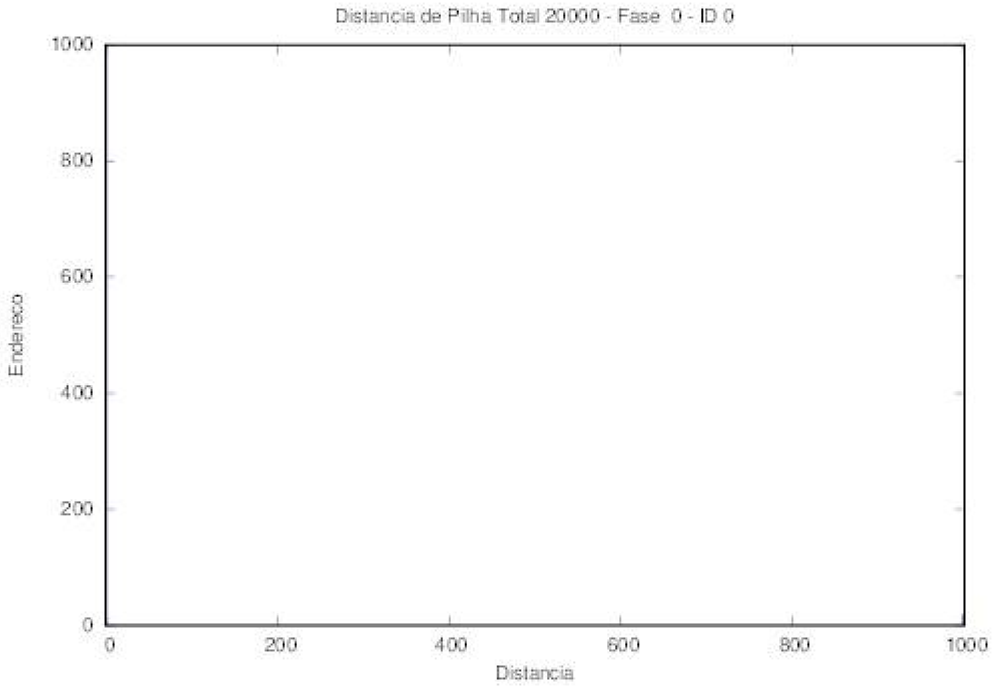


Distancia de Pilha Total 20000 - Fase 0 - ID 0
Endereco
Distancia
0
200
400
600
800
1000

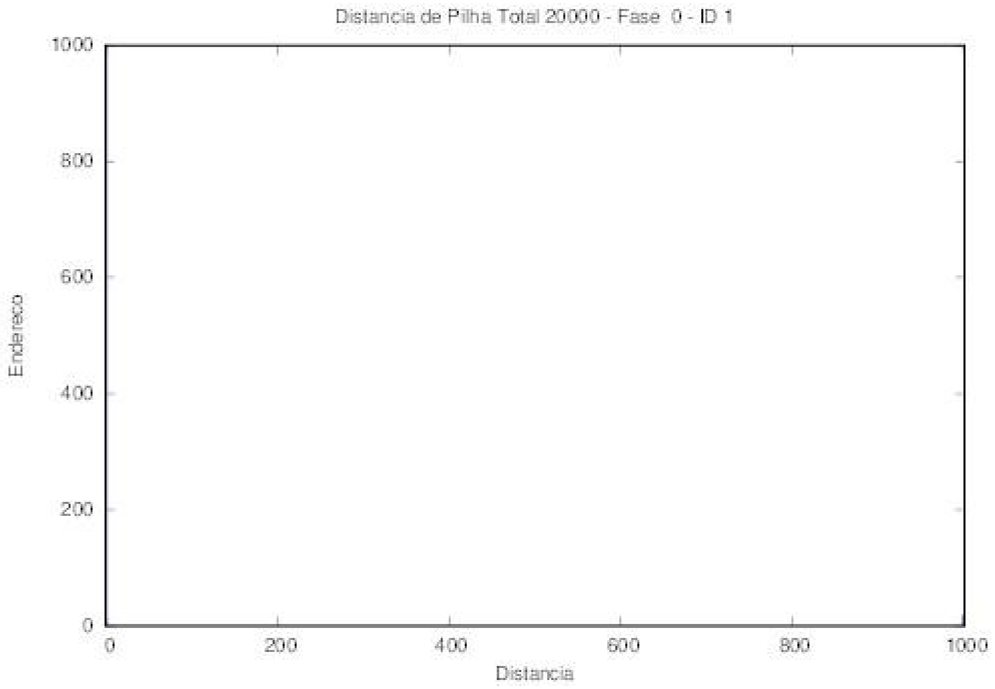


Distancia de Pilha Total 400 - Fase 1 - ID 0
Endereco
Distancia
0
5
10
15
20

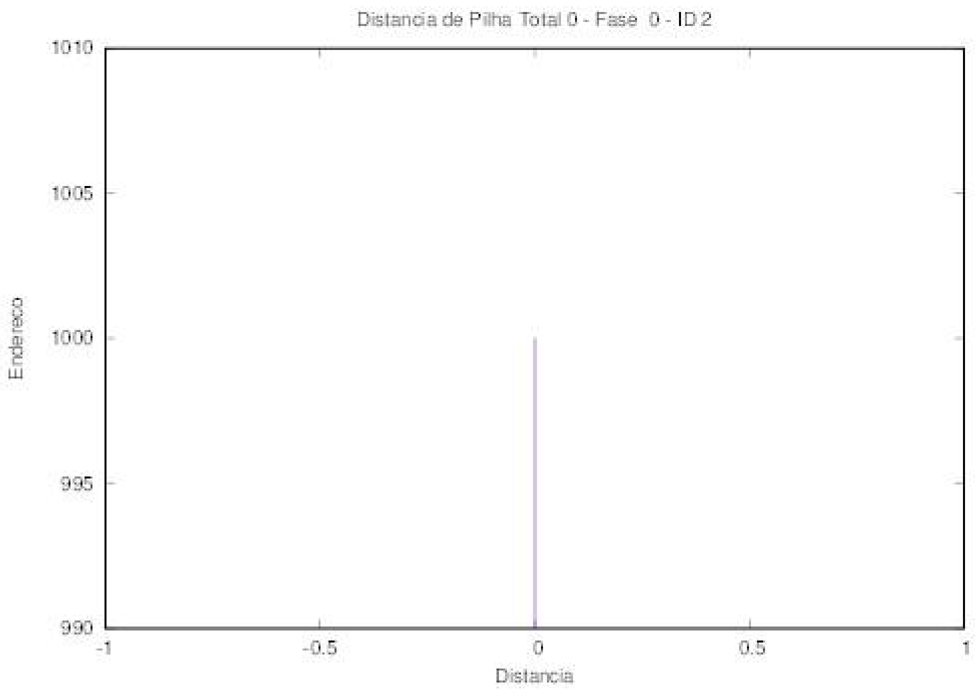
4.2.3.3. Soma

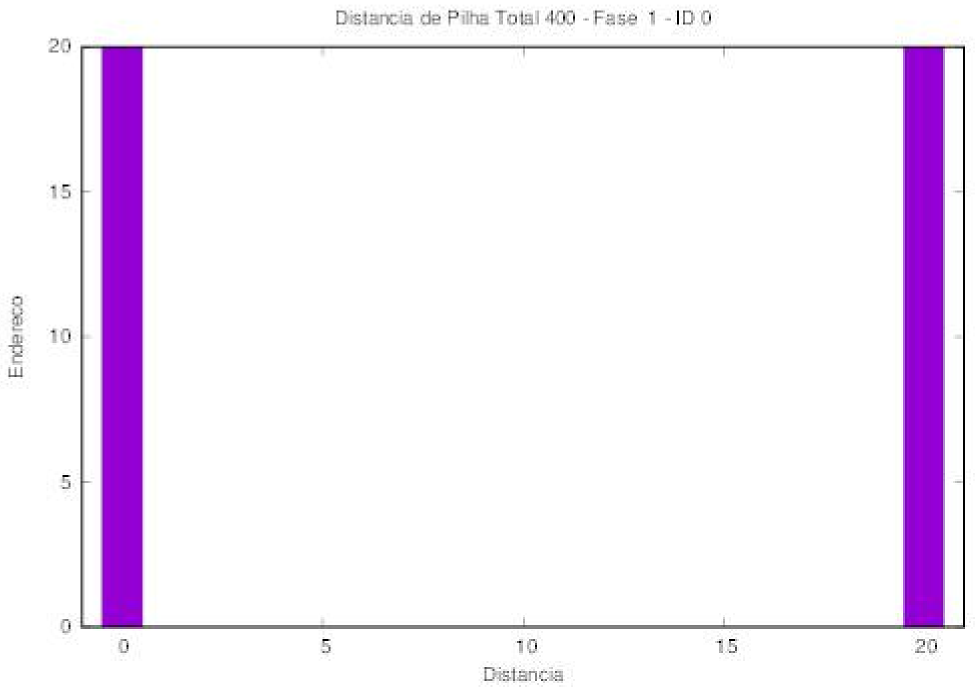


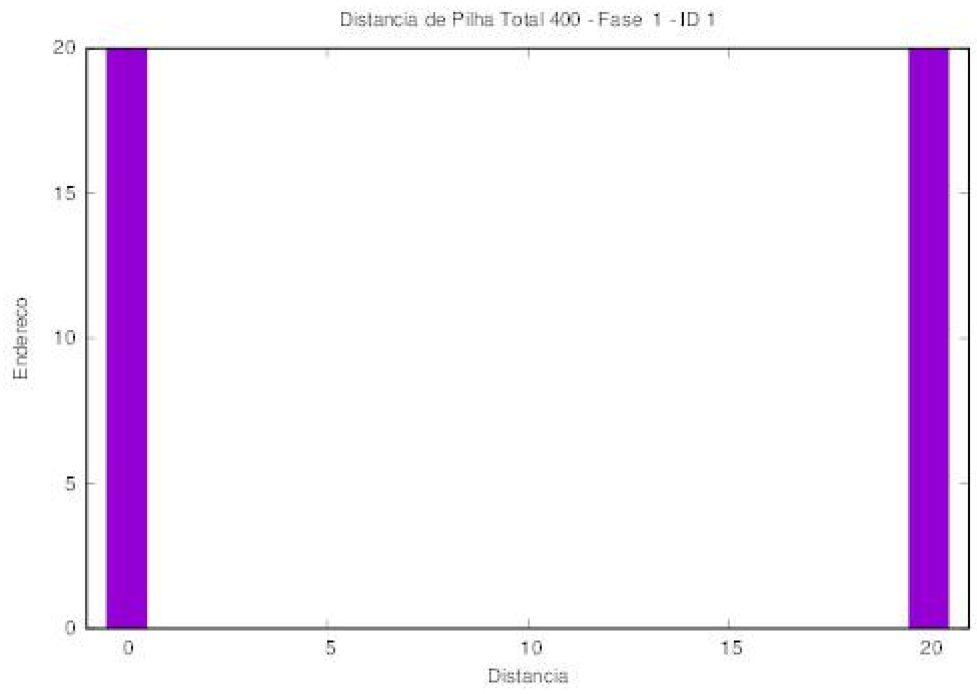
Distancia de Pilha Total 20000 - Fase 0 - ID 0
Endereco
Distancia
0
200
400
600
800
1000

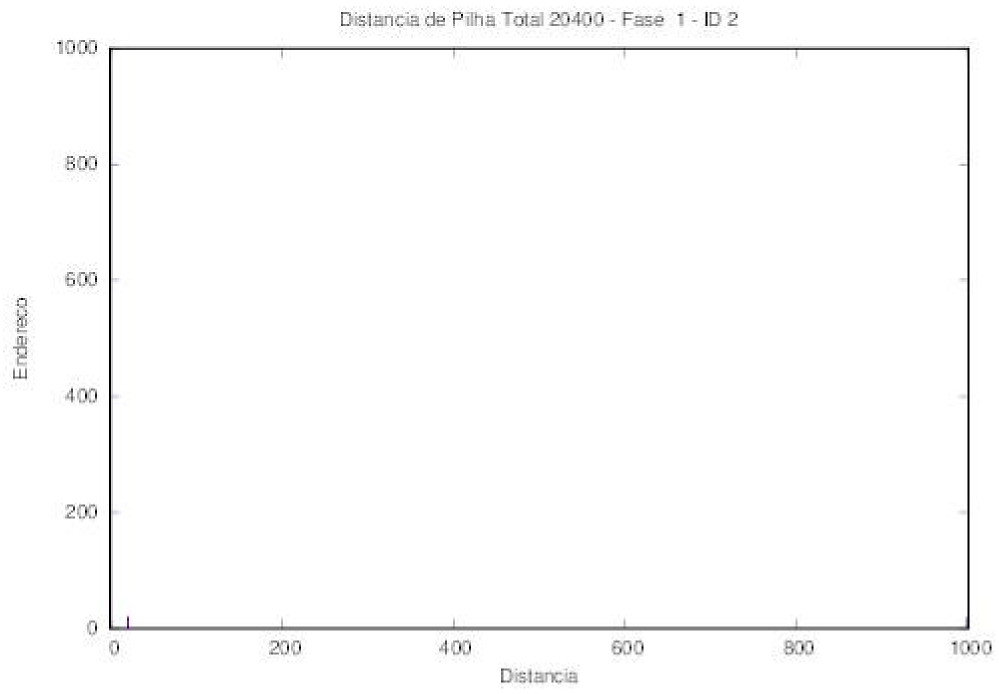


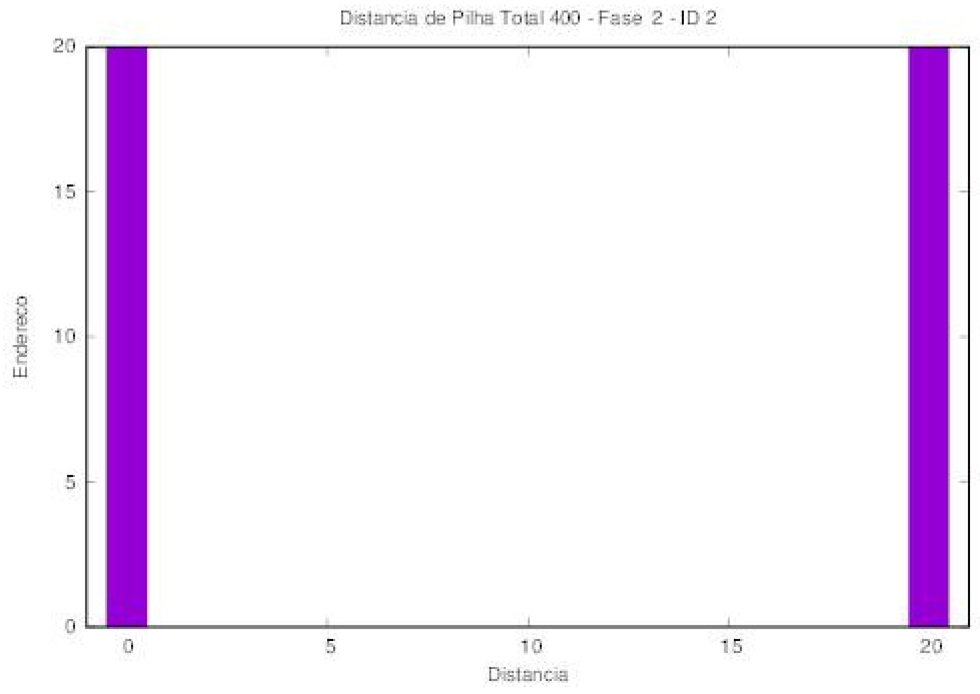
Distancia de Pilha Total 20000 - Fase 0 - ID 1
Endereco
Distancia
0
200
400
600
800
1000


Distancia de Pilha Total 0 - Fase 0 - ID 2
1010
1005
1000
995
990
Endereco
-1
-0.5
0
0.5
1
Distancia


Distancia de Pilha Total 400 - Fase 1 - ID 0
20
15
10
5
0
Endereco
0
5
10
15
20
Distancia


Distancia de Pilha Total 400 - Fase 1 - ID 1
Endereco
Distancia
0
5
10
15
20
0
5
10
15
20





5. Resultado Depuração gprof

5.1. Vetor Dinâmico

5.1.1. Interno

5.1.1.1. 1M

Flat profile:

Each sample counts as 0.01 seconds.

```
 %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 66.79           0.02     0.02     3        6.68     6.68  inicializaVetorNulo
 33.39           0.03     0.01     3        3.34     3.34  acessaVetor
  0.00   0.03     0.00     3        0.00     0.00  criaVetor
  0.00   0.03     0.00     3        0.00     0.00  defineFaseMemLog
  0.00   0.03     0.00     3        0.00     0.00  destroiVetor
  0.00   0.03     0.00     2        0.00     6.68  inicializaVetorAleatorio
  0.00   0.03     0.00     1        0.00     0.00  clkDifMemLog
  0.00   0.03     0.00     1        0.00     0.00  desativaMemLog
  0.00   0.03     0.00     1        0.00     0.00  finalizaMemLog
  0.00   0.03     0.00     1        0.00     0.00  iniciaMemLog
  0.00   0.03     0.00     1        0.00     0.00  parse_args
  0.00   0.03     0.00     1        0.00     0.00  produtoInternoVetores

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 33.27% of 0.03 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.03                 main [1]
                0.00    0.01       2/2           inicializaVetorAleatorio [3]
                0.01    0.00       3/3           acessaVetor [4]
                0.01    0.00       1/3           inicializaVetorNulo [2]
                0.00    0.00       3/3           defineFaseMemLog [6]
                0.00    0.00       3/3           criaVetor [5]
                0.00    0.00       3/3           destroiVetor [7]
                0.00    0.00       1/1           parse_args [12]
                0.00    0.00       1/1           iniciaMemLog [11]
                0.00    0.00       1/1           desativaMemLog [9]
                0.00    0.00       1/1           produtoInternoVetores [13]
                0.00    0.00       1/1           finalizaMemLog [10]
-----------------------------------------------
                0.01    0.00       1/3           main [1]
                0.01    0.00       2/3           inicializaVetorAleatorio [3]
[2]     66.7    0.02    0.00       3         inicializaVetorNulo [2]
-----------------------------------------------
                0.00    0.01       2/2           main [1]
[3]     44.4    0.00    0.01       2         inicializaVetorAleatorio [3]
                0.01    0.00       2/3           inicializaVetorNulo [2]
-----------------------------------------------
                0.01    0.00       3/3           main [1]
[4]     33.3    0.01    0.00       3         acessaVetor [4]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[5]      0.0    0.00    0.00       3         criaVetor [5]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[6]      0.0    0.00    0.00       3         defineFaseMemLog [6]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[7]      0.0    0.00    0.00       3         destroiVetor [7]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [10]
[8]      0.0    0.00    0.00       1         clkDifMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [1]
[9]      0.0    0.00    0.00       1         desativaMemLog [9]
-----------------------------------------------
                0.00    0.00       1/1           main [1]
[10]     0.0    0.00    0.00       1         finalizaMemLog [10]
```

```
                0.00    0.00    1/1             clkDifMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1         iniciaMemLog [11]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1         parse_args [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1         produtoInternoVetores [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.

     name    The name of the current function.  The index number is
      printed after it.  If the function is a member of a
      cycle, the cycle number is printed between the
      function's name and the index number.


 For the function's parents, the fields have the following meanings:
```

self   This is the amount of time that was propagated directly
from the function into this parent.

children   This is the amount of time that was propagated from
the function's children into this parent.

called   This is the number of times this parent called the
function `/' the total number of times the function
was called. Recursive calls to the function are not
included in the number after the `/'.

name   This is the name of the parent. The parent's index
number is printed after it. If the parent is a
member of a cycle, the cycle number is printed between
the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
from the child into the function.

children   This is the amount of time that was propagated from the
child's children to the function.

called   This is the number of times the function called
this child `/' the total number of times the child
was called. Recursive calls by the child are not
listed in the number after the `/'.

name   This is the name of the child. The child's index
number is printed after it. If the child is a
member of a cycle, the cycle number is printed
between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

```
Index by function name

   [4] acessaVetor              [9] desativaMemLog           [3] inicializaVetorAleatorio
   [8] clkDifMemLog             [7] destroiVetor          [2] inicializaVetorNulo
   [5] criaVetor           [10] finalizaMemLog          [12] parse_args
   [6] defineFaseMemLog     [11] iniciaMemLog              [13] produtoInternoVetores
```

5.1.1.2. 2M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 40.07          0.02     0.02     3         6.68     6.68  acessaVetor
 40.07          0.04     0.02     3         6.68     6.68  inicializaVetorNulo
 20.04          0.05     0.01     1         10.02   10.02  produtoInternoVetores
  0.00   0.05     0.00     3         0.00     0.00  criaVetor
  0.00   0.05     0.00     3         0.00     0.00  defineFaseMemLog
  0.00   0.05     0.00     3         0.00     0.00  destroiVetor
  0.00   0.05     0.00     2         0.00     6.68  inicializaVetorAleatorio
  0.00   0.05     0.00     1         0.00     0.00  clkDifMemLog
  0.00   0.05     0.00     1         0.00     0.00  desativaMemLog
  0.00   0.05     0.00     1         0.00     0.00  finalizaMemLog
  0.00   0.05     0.00     1         0.00     0.00  iniciaMemLog
  0.00   0.05     0.00     1         0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.
 total    the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
          function is profiled, else blank.

name            the name of the function.  This is the minor sort
          for this listing. The index shows the location of
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 19.96% of 0.05 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.05                 main [1]
                0.02    0.00       3/3           acessaVetor [2]
                0.00    0.01       2/2           inicializaVetorAleatorio [4]
                0.01    0.00       1/1           produtoInternoVetores [5]
                0.01    0.00       1/3           inicializaVetorNulo [3]
                0.00    0.00       3/3           defineFaseMemLog [7]
                0.00    0.00       3/3           criaVetor [6]
                0.00    0.00       3/3           destroiVetor [8]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [12]
                0.00    0.00       1/1           desativaMemLog [10]
                0.00    0.00       1/1           finalizaMemLog [11]
-----------------------------------------------
                0.02    0.00       3/3           main [1]
[2]     40.0    0.02    0.00       3         acessaVetor [2]
-----------------------------------------------
                0.01    0.00       1/3           main [1]
                0.01    0.00       2/3           inicializaVetorAleatorio [4]
[3]     40.0    0.02    0.00       3         inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.01       2/2           main [1]
[4]     26.7    0.00    0.01       2         inicializaVetorAleatorio [4]
                0.01    0.00       2/3           inicializaVetorNulo [3]
-----------------------------------------------
                0.01    0.00       1/1           main [1]
```

```
[5]     20.0    0.01    0.00    1       produtoInternoVetores [5]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[6]     0.0     0.00    0.00    3       criaVetor [6]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[7]     0.0     0.00    0.00    3       defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[8]     0.0     0.00    0.00    3       destroiVetor [8]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [11]
[9]     0.0     0.00    0.00    1       clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1       desativaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1       finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1       iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1       parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index      A unique number given to each element of the table.
                Index numbers are sorted numerically.
                The index number is printed next to every function name so
                it is easier to look up where the function is in the table.

     % time     This is the percentage of the `total' time that was spent
                in this function and its children.  Note that due to
                different viewpoints, functions excluded by options, etc,
                these numbers will NOT add up to 100%.

     self       This is the total amount of time spent in this function.
```

children This is the total amount of time propagated into this function by its children.

called This is the number of times the function was called. If the function called itself recursively, the number only includes non-recursive calls, and is followed by a `+' and the number of recursive calls.

name The name of the current function. The index number is printed after it. If the function is a member of a cycle, the cycle number is printed between the function's name and the index number.

For the function's parents, the fields have the following meanings:

self This is the amount of time that was propagated directly from the function into this parent.

children This is the amount of time that was propagated from the function's children into this parent.

called This is the number of times this parent called the function `/' the total number of times the function was called. Recursive calls to the function are not included in the number after the `/'.

name This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>' is printed in the `name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly from the child into the function.

children This is the amount of time that was propagated from the child's children to the function.

called This is the number of times the function called this child `/' the total number of times the child was called. Recursive calls by the child are not listed in the number after the `/'.

```
          name    This is the name of the child.  The child's index
           number is printed after it.  If the child is a
           member of a cycle, the cycle number is printed
           between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.
```

```
Index by function name

   [2] acessaVetor            [10] desativaMemLog         [4] inicializaVetorAleatorio
   [9] clkDifMemLog           [8] destroiVetor        [3] inicializaVetorNulo
   [6] criaVetor          [11] finalizaMemLog         [13] parse_args
   [7] defineFaseMemLog       [12] iniciaMemLog           [5] produtoInternoVetores
```

5.1.1.3. 3M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self          self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 37.57        0.03     0.03     3     10.02    10.02  acessaVetor
 25.05        0.05     0.02     3     6.68     6.68  inicializaVetorNulo
 25.05        0.07     0.02     2     10.02    16.70  inicializaVetorAleatorio
 12.52        0.08     0.01     1     10.02    10.02  produtoInternoVetores
  0.00   0.08    0.00    3     0.00    0.00  criaVetor
  0.00   0.08    0.00    3     0.00    0.00  defineFaseMemLog
  0.00   0.08    0.00    3     0.00    0.00  destroiVetor
  0.00   0.08    0.00    1     0.00    0.00  clkDifMemLog
  0.00   0.08    0.00    1     0.00    0.00  desativaMemLog
  0.00   0.08    0.00    1     0.00    0.00  finalizaMemLog
  0.00   0.08    0.00    1     0.00    0.00  iniciaMemLog
  0.00   0.08    0.00    1     0.00    0.00  parse_args
 %          the percentage of the total running time of the
time        program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self       the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
            listing.

calls       the number of times this function was invoked, if
            this function is profiled, else blank.

 self       the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
            else blank.

 total      the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
            function is profiled, else blank.

name             the name of the function.  This is the minor sort
            for this listing. The index shows the location of
            the function in the gprof listing. If the index is
            in parenthesis it shows where it would appear in
            the gprof listing if it were to be printed.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 12.48% of 0.08 seconds

index % time    self  children    called     name
                                             <spontaneous>
[1]     100.0    0.00    0.08                main [1]
                 0.02    0.01      2/2           inicializaVetorAleatorio [2]
                 0.03    0.00      3/3           acessaVetor [3]
                 0.01    0.00      1/1           produtoInternoVetores [5]
                 0.01    0.00      1/3           inicializaVetorNulo [4]
                 0.00    0.00      3/3           defineFaseMemLog [7]
                 0.00    0.00      3/3           criaVetor [6]
```

```
                0.00    0.00    3/3           destroiVetor [8]
                0.00    0.00    1/1           parse_args [13]
                0.00    0.00    1/1           iniciaMemLog [12]
                0.00    0.00    1/1           desativaMemLog [10]
                0.00    0.00    1/1           finalizaMemLog [11]
-----------------------------------------------
                0.02    0.01    2/2           main [1]
[2]     41.7    0.02    0.01    2         inicializaVetorAleatorio [2]
                0.01    0.00    2/3           inicializaVetorNulo [4]
-----------------------------------------------
                0.03    0.00    3/3           main [1]
[3]     37.5    0.03    0.00    3         acessaVetor [3]
-----------------------------------------------
                0.01    0.00    1/3           main [1]
                0.01    0.00    2/3           inicializaVetorAleatorio [2]
[4]     25.0    0.02    0.00    3         inicializaVetorNulo [4]
-----------------------------------------------
                0.01    0.00    1/1           main [1]
[5]     12.5    0.01    0.00    1         produtoInternoVetores [5]
-----------------------------------------------
                0.00    0.00    3/3           main [1]
[6]     0.0     0.00    0.00    3         criaVetor [6]
-----------------------------------------------
                0.00    0.00    3/3           main [1]
[7]     0.0     0.00    0.00    3         defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00    3/3           main [1]
[8]     0.0     0.00    0.00    3         destroiVetor [8]
-----------------------------------------------
                0.00    0.00    1/1           finalizaMemLog [11]
[9]     0.0     0.00    0.00    1         clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1           main [1]
[10]    0.0     0.00    0.00    1         desativaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1           main [1]
[11]    0.0     0.00    0.00    1         finalizaMemLog [11]
                0.00    0.00    1/1           clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1           main [1]
[12]    0.0     0.00    0.00    1         iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1           main [1]
[13]    0.0     0.00    0.00    1         parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.
```

```
Each entry in this table consists of several lines.  The line with the
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.

     name    The name of the current function.  The index number is
      printed after it.  If the function is a member of a
      cycle, the cycle number is printed between the
      function's name and the index number.


For the function's parents, the fields have the following meanings:

     self    This is the amount of time that was propagated directly
      from the function into this parent.

     children    This is the amount of time that was propagated from
      the function's children into this parent.

     called    This is the number of times this parent called the
      function `/' the total number of times the function
      was called.  Recursive calls to the function are not
      included in the number after the `/'.

     name    This is the name of the parent.  The parent's index
      number is printed after it.  If the parent is a
      member of a cycle, the cycle number is printed between
```

```
            the name and the index number.

 If the parents of the function cannot be determined, the word
 `<spontaneous>' is printed in the `name' field, and all the other
 fields are blank.

 For the function's children, the fields have the following meanings:

           self    This is the amount of time that was propagated directly
            from the child into the function.

           children    This is the amount of time that was propagated from the
            child's children to the function.

           called    This is the number of times the function called
            this child `/' the total number of times the child
            was called.  Recursive calls by the child are not
            listed in the number after the `/'.

           name    This is the name of the child.  The child's index
            number is printed after it.  If the child is a
            member of a cycle, the cycle number is printed
            between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.
```

```
Index by function name

   [3] acessaVetor             [10] desativaMemLog          [2] inicializaVetorAleatorio
   [9] clkDifMemLog             [8] destroiVetor         [4] inicializaVetorNulo
   [6] criaVetor           [11] finalizaMemLog          [13] parse_args
   [7] defineFaseMemLog        [12] iniciaMemLog             [5] produtoInternoVetores
```

5.1.1.4. 4M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self           self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 36.43          0.04    0.04    3        13.36   13.36  acessaVetor
 36.43          0.08    0.04    3        13.36   13.36  inicializaVetorNulo
 18.21          0.10    0.02    2        10.02   23.38  inicializaVetorAleatorio
  9.11   0.11     0.01    1         10.02   10.02  produtoInternoVetores
  0.00   0.11     0.00    3         0.00    0.00  criaVetor
  0.00   0.11     0.00    3         0.00    0.00  defineFaseMemLog
  0.00   0.11     0.00    3         0.00    0.00  destroiVetor
  0.00   0.11     0.00    1         0.00    0.00  clkDifMemLog
  0.00   0.11     0.00    1         0.00    0.00  desativaMemLog
  0.00   0.11     0.00    1         0.00    0.00  finalizaMemLog
  0.00   0.11     0.00    1         0.00    0.00  iniciaMemLog
  0.00   0.11     0.00    1         0.00    0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
           function is profiled, else blank.

name             the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 9.07% of 0.11 seconds

index % time    self  children    called     name
                                             <spontaneous>
[1]    100.0    0.00    0.11                 main [1]
                0.02    0.03      2/2            inicializaVetorAleatorio [2]
                0.04    0.00      3/3            acessaVetor [3]
                0.01    0.00      1/3            inicializaVetorNulo [4]
                0.01    0.00      1/1            produtoInternoVetores [5]
                0.00    0.00      3/3            defineFaseMemLog [7]
                0.00    0.00      3/3            criaVetor [6]
                0.00    0.00      3/3            destroiVetor [8]
                0.00    0.00      1/1            parse_args [13]
                0.00    0.00      1/1            iniciaMemLog [12]
                0.00    0.00      1/1            desativaMemLog [10]
                0.00    0.00      1/1            finalizaMemLog [11]
-----------------------------------------------
                0.02    0.03      2/2            main [1]
[2]     42.4    0.02    0.03      2          inicializaVetorAleatorio [2]
                0.03    0.00      2/3            inicializaVetorNulo [4]
-----------------------------------------------
                0.04    0.00      3/3            main [1]
[3]     36.4    0.04    0.00      3          acessaVetor [3]
-----------------------------------------------
                0.01    0.00      1/3            main [1]
                0.03    0.00      2/3            inicializaVetorAleatorio [2]
[4]     36.4    0.04    0.00      3          inicializaVetorNulo [4]
-----------------------------------------------
                0.01    0.00      1/1            main [1]
[5]      9.1    0.01    0.00      1          produtoInternoVetores [5]
-----------------------------------------------
                0.00    0.00      3/3            main [1]
[6]      0.0    0.00    0.00      3          criaVetor [6]
-----------------------------------------------
                0.00    0.00      3/3            main [1]
[7]      0.0    0.00    0.00      3          defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00      3/3            main [1]
[8]      0.0    0.00    0.00      3          destroiVetor [8]
-----------------------------------------------
```

```
                0.00    0.00     1/1              finalizaMemLog [11]
[9]     0.0     0.00    0.00     1          clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[10]    0.0     0.00    0.00     1          desativaMemLog [10]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[11]    0.0     0.00    0.00     1          finalizaMemLog [11]
                0.00    0.00     1/1              clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[12]    0.0     0.00    0.00     1          iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[13]    0.0     0.00    0.00     1          parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index	A unique number given to each element of the table.
		Index numbers are sorted numerically.
		The index number is printed next to every function name so
		it is easier to look up where the function is in the table.

     % time	This is the percentage of the `total' time that was spent
		in this function and its children.  Note that due to
		different viewpoints, functions excluded by options, etc,
		these numbers will NOT add up to 100%.

     self	This is the total amount of time spent in this function.

     children	This is the total amount of time propagated into this
		function by its children.

     called	This is the number of times the function was called.
		If the function called itself recursively, the number
		only includes non-recursive calls, and is followed by
		a `+' and the number of recursive calls.

     name	The name of the current function.  The index number is
		printed after it.  If the function is a member of a
		cycle, the cycle number is printed between the
```

function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly from the function into this parent.

children   This is the amount of time that was propagated from the function's children into this parent.

called   This is the number of times this parent called the function `/' the total number of times the function was called.  Recursive calls to the function are not included in the number after the `/'.

name   This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>' is printed in the `name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly from the child into the function.

children   This is the amount of time that was propagated from the child's children to the function.

called   This is the number of times the function called this child `/' the total number of times the child was called.  Recursive calls by the child are not listed in the number after the `/'.

name   This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The `+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows,

```
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [3] acessaVetor              [10] desativaMemLog          [2] inicializaVetorAleatorio
   [9] clkDifMemLog              [8] destroiVetor          [4] inicializaVetorNulo
   [6] criaVetor           [11] finalizaMemLog            [13] parse_args
   [7] defineFaseMemLog         [12] iniciaMemLog              [5] produtoInternoVetores
```

5.1.1.5. 5M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self           self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 45.54            0.05    0.05    3        16.70   16.70  acessaVetor
 45.54            0.10    0.05    3        16.70   16.70  inicializaVetorNulo
  9.11   0.11     0.01    1       10.02   10.02  produtoInternoVetores
  0.00   0.11     0.00    3       0.00    0.00  criaVetor
  0.00   0.11     0.00    3       0.00    0.00  defineFaseMemLog
  0.00   0.11     0.00    3       0.00    0.00  destroiVetor
  0.00   0.11     0.00    2       0.00    16.70  inicializaVetorAleatorio
  0.00   0.11     0.00    1       0.00    0.00  clkDifMemLog
  0.00   0.11     0.00    1       0.00    0.00  desativaMemLog
  0.00   0.11     0.00    1       0.00    0.00  finalizaMemLog
  0.00   0.11     0.00    1       0.00    0.00  iniciaMemLog
  0.00   0.11     0.00    1       0.00    0.00  parse_args

 %       the percentage of the total running time of the
time     program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self     the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
          listing.
```

```
calls    the number of times this function was invoked, if
         this function is profiled, else blank.

 self    the average number of milliseconds spent in this
ms/call  function per call, if this function is profiled,
         else blank.

 total   the average number of milliseconds spent in this
ms/call  function and its descendents per call, if this
         function is profiled, else blank.

name           the name of the function.  This is the minor sort
         for this listing. The index shows the location of
         the function in the gprof listing. If the index is
         in parenthesis it shows where it would appear in
         the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 9.07% of 0.11 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.11                 main [1]
                0.05    0.00       3/3           acessaVetor [2]
                0.00    0.03       2/2           inicializaVetorAleatorio [4]
                0.02    0.00       1/3           inicializaVetorNulo [3]
                0.01    0.00       1/1           produtoInternoVetores [5]
                0.00    0.00       3/3           defineFaseMemLog [7]
                0.00    0.00       3/3           criaVetor [6]
                0.00    0.00       3/3           destroiVetor [8]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [12]
                0.00    0.00       1/1           desativaMemLog [10]
                0.00    0.00       1/1           finalizaMemLog [11]
-----------------------------------------------
                0.05    0.00       3/3           main [1]
[2]     45.5    0.05    0.00       3         acessaVetor [2]
-----------------------------------------------
                0.02    0.00       1/3           main [1]
                0.03    0.00       2/3           inicializaVetorAleatorio [4]
```

```
[3]     45.5    0.05    0.00    3       inicializaVetorNulo [3]
------------------------------------------------
                0.00    0.03    2/2             main [1]
[4]     30.3    0.00    0.03    2       inicializaVetorAleatorio [4]
                0.03    0.00    2/3             inicializaVetorNulo [3]
------------------------------------------------
                0.01    0.00    1/1             main [1]
[5]     9.1     0.01    0.00    1       produtoInternoVetores [5]
------------------------------------------------
                0.00    0.00    3/3             main [1]
[6]     0.0     0.00    0.00    3       criaVetor [6]
------------------------------------------------
                0.00    0.00    3/3             main [1]
[7]     0.0     0.00    0.00    3       defineFaseMemLog [7]
------------------------------------------------
                0.00    0.00    3/3             main [1]
[8]     0.0     0.00    0.00    3       destroiVetor [8]
------------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [11]
[9]     0.0     0.00    0.00    1       clkDifMemLog [9]
------------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1       desativaMemLog [10]
------------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1       finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [9]
------------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1       iniciaMemLog [12]
------------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1       parse_args [13]
------------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
         index    A unique number given to each element of the table.
          Index numbers are sorted numerically.
          The index number is printed next to every function name so
          it is easier to look up where the function is in the table.
```

% time   This is the percentage of the `total' time that was spent
 in this function and its children. Note that due to
 different viewpoints, functions excluded by options, etc,
 these numbers will NOT add up to 100%.

self   This is the total amount of time spent in this function.

children   This is the total amount of time propagated into this
 function by its children.

called   This is the number of times the function was called.
 If the function called itself recursively, the number
 only includes non-recursive calls, and is followed by
 a `+' and the number of recursive calls.

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

```
     children	This is the amount of time that was propagated from the
		child's children to the function.

     called	This is the number of times the function called
		this child `/' the total number of times the child
		was called.  Recursive calls by the child are not
		listed in the number after the `/'.

     name	This is the name of the child.  The child's index
		number is printed after it.  If the child is a
		member of a cycle, the cycle number is printed
		between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.
```

```
Index by function name

   [2] acessaVetor            [10] desativaMemLog          [4] inicializaVetorAleatorio
   [9] clkDifMemLog            [8] destroiVetor            [3] inicializaVetorNulo
   [6] criaVetor              [11] finalizaMemLog         [13] parse_args
   [7] defineFaseMemLog       [12] iniciaMemLog            [5] produtoInternoVetores
```

5.1.2. Norma

5.1.2.1. 1M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
100.18      0.01     0.01        1    10.02    10.02  acessaVetor
  0.00      0.01     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.01     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.01     0.00        1     0.00     0.00  criaVetor
  0.00  0.01    0.00    1        0.00    0.00  desativaMemLog
  0.00  0.01    0.00    1        0.00    0.00  destroiVetor
  0.00  0.01    0.00    1        0.00    0.00  finalizaMemLog
  0.00  0.01    0.00    1        0.00    0.00  iniciaMemLog
  0.00  0.01    0.00    1        0.00    0.00  inicializaVetorAleatorio
  0.00  0.01    0.00    1        0.00    0.00  inicializaVetorNulo
  0.00  0.01    0.00    1        0.00    0.00  normaVetor
  0.00  0.01    0.00    1        0.00    0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 99.82% of 0.01 seconds

index % time    self  children    called     name
```

```
                0.01   0.00   1/1          main [2]
[1]     100.0   0.01   0.00   1        acessaVetor [1]
-----------------------------------------------
                              <spontaneous>
[2]     100.0   0.00   0.01            main [2]
                0.01   0.00   1/1          acessaVetor [1]
                0.00   0.00   3/3          defineFaseMemLog [3]
                0.00   0.00   1/1          parse_args [13]
                0.00   0.00   1/1          iniciaMemLog [9]
                0.00   0.00   1/1          desativaMemLog [6]
                0.00   0.00   1/1          criaVetor [5]
                0.00   0.00   1/1          inicializaVetorAleatorio [10]
                0.00   0.00   1/1          normaVetor [12]
                0.00   0.00   1/1          destroiVetor [7]
                0.00   0.00   1/1          finalizaMemLog [8]
-----------------------------------------------
                0.00   0.00   3/3          main [2]
[3]     0.0     0.00   0.00   3        defineFaseMemLog [3]
-----------------------------------------------
                0.00   0.00   1/1          finalizaMemLog [8]
[4]     0.0     0.00   0.00   1        clkDifMemLog [4]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[5]     0.0     0.00   0.00   1        criaVetor [5]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[6]     0.0     0.00   0.00   1        desativaMemLog [6]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[7]     0.0     0.00   0.00   1        destroiVetor [7]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[8]     0.0     0.00   0.00   1        finalizaMemLog [8]
                0.00   0.00   1/1          clkDifMemLog [4]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[9]     0.0     0.00   0.00   1        iniciaMemLog [9]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[10]    0.0     0.00   0.00   1        inicializaVetorAleatorio [10]
                0.00   0.00   1/1          inicializaVetorNulo [11]
-----------------------------------------------
                0.00   0.00   1/1          inicializaVetorAleatorio [10]
[11]    0.0     0.00   0.00   1        inicializaVetorNulo [11]
-----------------------------------------------
                0.00   0.00   1/1          main [2]
[12]    0.0     0.00   0.00   1        normaVetor [12]
-----------------------------------------------
```

```
                0.00    0.00    1/1             main [2]
[13]    0.0     0.00    0.00    1         parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
              Index numbers are sorted numerically.
              The index number is printed next to every function name so
              it is easier to look up where the function is in the table.

     % time   This is the percentage of the `total' time that was spent
              in this function and its children.  Note that due to
              different viewpoints, functions excluded by options, etc,
              these numbers will NOT add up to 100%.

     self     This is the total amount of time spent in this function.

     children This is the total amount of time propagated into this
              function by its children.

     called   This is the number of times the function was called.
              If the function called itself recursively, the number
              only includes non-recursive calls, and is followed by
              a `+' and the number of recursive calls.

     name     The name of the current function.  The index number is
              printed after it.  If the function is a member of a
              cycle, the cycle number is printed between the
              function's name and the index number.


 For the function's parents, the fields have the following meanings:

     self     This is the amount of time that was propagated directly
              from the function into this parent.

     children This is the amount of time that was propagated from
              the function's children into this parent.

     called   This is the number of times this parent called the
              function `/' the total number of times the function
```

was called. Recursive calls to the function are not
included in the number after the `/'.

name This is the name of the parent. The parent's index
number is printed after it. If the parent is a
member of a cycle, the cycle number is printed between
the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly
from the child into the function.

children This is the amount of time that was propagated from the
child's children to the function.

called This is the number of times the function called
this child `/' the total number of times the child
was called. Recursive calls by the child are not
listed in the number after the `/'.

name This is the name of the child. The child's index
number is printed after it. If the child is a
member of a cycle, the cycle number is printed
between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

Index by function name

```
  [1] acessaVetor             [6] desativaMemLog         [10] inicializaVetorAleatorio
  [4] clkDifMemLog            [7] destroiVetor       [11] inicializaVetorNulo
```

```
   [5] criaVetor            [8] finalizaMemLog     [12] normaVetor
   [3] defineFaseMemLog       [9] iniciaMemLog       [13] parse_args
```

5.1.2.2. 2M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 50.09      0.01     0.01        1    10.02    10.02  acessaVetor
 50.09      0.02     0.01        1    10.02    10.02  normaVetor
  0.00      0.02     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.02     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.02     0.00        1     0.00     0.00  criaVetor
  0.00      0.02     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.02     0.00        1     0.00     0.00  destroiVetor
  0.00      0.02     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.02     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.02     0.00        1     0.00     0.00  inicializaVetorAleatorio
  0.00      0.02     0.00        1     0.00     0.00  inicializaVetorNulo
  0.00      0.02     0.00        1     0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
```

```
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 49.91% of 0.02 seconds

index % time    self  children    called     name
                                             <spontaneous>
[1]    100.0    0.00    0.02                 main [1]
                0.01    0.00       1/1           acessaVetor [2]
                0.01    0.00       1/1           normaVetor [3]
                0.00    0.00       3/3           defineFaseMemLog [4]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [10]
                0.00    0.00       1/1           desativaMemLog [7]
                0.00    0.00       1/1           criaVetor [6]
                0.00    0.00       1/1           inicializaVetorAleatorio [11]
                0.00    0.00       1/1           destroiVetor [8]
                0.00    0.00       1/1           finalizaMemLog [9]
-----------------------------------------------
                0.01    0.00       1/1           main [1]
[2]     50.0    0.01    0.00       1         acessaVetor [2]
-----------------------------------------------
                0.01    0.00       1/1           main [1]
[3]     50.0    0.01    0.00       1         normaVetor [3]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[4]      0.0    0.00    0.00       3         defineFaseMemLog [4]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[5]      0.0    0.00    0.00       1         clkDifMemLog [5]
-----------------------------------------------
                0.00    0.00       1/1           main [1]
[6]      0.0    0.00    0.00       1         criaVetor [6]
-----------------------------------------------
                0.00    0.00       1/1           main [1]
[7]      0.0    0.00    0.00       1         desativaMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [1]
```

```
[8]     0.0     0.00    0.00    1       destroiVetor [8]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[9]     0.0     0.00    0.00    1       finalizaMemLog [9]
                0.00    0.00    1/1             clkDifMemLog [5]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1       iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1       inicializaVetorAleatorio [11]
                0.00    0.00    1/1             inicializaVetorNulo [12]
-----------------------------------------------
                0.00    0.00    1/1             inicializaVetorAleatorio [11]
[12]    0.0     0.00    0.00    1       inicializaVetorNulo [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1       parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.
```

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

children   This is the amount of time that was propagated from the
 child's children to the function.

called   This is the number of times the function called
 this child `/' the total number of times the child
 was called. Recursive calls by the child are not
 listed in the number after the `/'.

name   This is the name of the child. The child's index
 number is printed after it. If the child is a
 member of a cycle, the cycle number is printed
 between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the

```
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [2] acessaVetor              [7] desativaMemLog          [11] inicializaVetorAleatorio
   [5] clkDifMemLog             [8] destroiVetor        [12] inicializaVetorNulo
   [6] criaVetor         [9] finalizaMemLog             [3] normaVetor
   [4] defineFaseMemLog        [10] iniciaMemLog             [13] parse_args
```

5.1.2.3. 3M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 33.39      0.01     0.01        1    10.02    10.02  acessaVetor
 33.39      0.02     0.01        1    10.02    10.02  inicializaVetorAleatorio
 33.39      0.03     0.01        1    10.02    10.02  normaVetor
  0.00      0.03     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.03     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.03     0.00        1     0.00     0.00  criaVetor
  0.00      0.03     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.03     0.00        1     0.00     0.00  destroiVetor
  0.00      0.03     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.03     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.03     0.00        1     0.00     0.00  inicializaVetorNulo
  0.00      0.03     0.00        1     0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
```

```
seconds          function alone.  This is the major sort for this
          listing.

calls     the number of times this function was invoked, if
          this function is profiled, else blank.

 self     the average number of milliseconds spent in this
ms/call   function per call, if this function is profiled,
          else blank.

 total    the average number of milliseconds spent in this
ms/call   function and its descendents per call, if this
          function is profiled, else blank.

name             the name of the function.  This is the minor sort
          for this listing. The index shows the location of
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 33.27% of 0.03 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.03                 main [1]
                0.01    0.00       1/1           inicializaVetorAleatorio [3]
                0.01    0.00       1/1           acessaVetor [2]
                0.01    0.00       1/1           normaVetor [4]
                0.00    0.00       3/3           defineFaseMemLog [5]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [11]
                0.00    0.00       1/1           desativaMemLog [8]
                0.00    0.00       1/1           criaVetor [7]
                0.00    0.00       1/1           destroiVetor [9]
                0.00    0.00       1/1           finalizaMemLog [10]
-----------------------------------------------
                0.01    0.00       1/1           main [1]
[2]     33.3    0.01    0.00       1         acessaVetor [2]
-----------------------------------------------
```

```
                0.01    0.00    1/1              main [1]
[3]     33.3    0.01    0.00    1           inicializaVetorAleatorio [3]
                0.00    0.00    1/1              inicializaVetorNulo [12]
-----------------------------------------------
                0.01    0.00    1/1              main [1]
[4]     33.3    0.01    0.00    1           normaVetor [4]
-----------------------------------------------
                0.00    0.00    3/3              main [1]
[5]     0.0     0.00    0.00    3           defineFaseMemLog [5]
-----------------------------------------------
                0.00    0.00    1/1              finalizaMemLog [10]
[6]     0.0     0.00    0.00    1           clkDifMemLog [6]
-----------------------------------------------
                0.00    0.00    1/1              main [1]
[7]     0.0     0.00    0.00    1           criaVetor [7]
-----------------------------------------------
                0.00    0.00    1/1              main [1]
[8]     0.0     0.00    0.00    1           desativaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1              main [1]
[9]     0.0     0.00    0.00    1           destroiVetor [9]
-----------------------------------------------
                0.00    0.00    1/1              main [1]
[10]    0.0     0.00    0.00    1           finalizaMemLog [10]
                0.00    0.00    1/1              clkDifMemLog [6]
-----------------------------------------------
                0.00    0.00    1/1              main [1]
[11]    0.0     0.00    0.00    1           iniciaMemLog [11]
-----------------------------------------------
                0.00    0.00    1/1              inicializaVetorAleatorio [3]
[12]    0.0     0.00    0.00    1           inicializaVetorNulo [12]
-----------------------------------------------
                0.00    0.00    1/1              main [1]
[13]    0.0     0.00    0.00    1           parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
              Index numbers are sorted numerically.
              The index number is printed next to every function name so
              it is easier to look up where the function is in the table.
```

% time   This is the percentage of the `total' time that was spent
 in this function and its children. Note that due to
 different viewpoints, functions excluded by options, etc,
 these numbers will NOT add up to 100%.

self   This is the total amount of time spent in this function.

children   This is the total amount of time propagated into this
 function by its children.

called   This is the number of times the function was called.
 If the function called itself recursively, the number
 only includes non-recursive calls, and is followed by
 a `+' and the number of recursive calls.

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

```
          children    This is the amount of time that was propagated from the
           child's children to the function.

          called    This is the number of times the function called
           this child `/' the total number of times the child
           was called.  Recursive calls by the child are not
           listed in the number after the `/'.

          name    This is the name of the child.  The child's index
           number is printed after it.  If the child is a
           member of a cycle, the cycle number is printed
           between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [2] acessaVetor              [8] desativaMemLog            [3] inicializaVetorAleatorio
   [6] clkDifMemLog             [9] destroiVetor         [12] inicializaVetorNulo
   [7] criaVetor           [10] finalizaMemLog          [4] normaVetor
   [5] defineFaseMemLog       [11] iniciaMemLog             [13] parse_args
```

5.1.2.4.   4M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self          self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 28.62          0.02    0.02    1        20.04   20.04  acessaVetor
 28.62          0.04    0.02    1        20.04   30.05  inicializaVetorAleatorio
 28.62          0.06    0.02    1        20.04   20.04  normaVetor
 14.31          0.07    0.01    1        10.02   10.02  inicializaVetorNulo
```

```
  0.00   0.07     0.00     3          0.00     0.00  defineFaseMemLog
  0.00   0.07     0.00     1          0.00     0.00  clkDifMemLog
  0.00   0.07     0.00     1          0.00     0.00  criaVetor
  0.00   0.07     0.00     1          0.00     0.00  desativaMemLog
  0.00   0.07     0.00     1          0.00     0.00  destroiVetor
  0.00   0.07     0.00     1          0.00     0.00  finalizaMemLog
  0.00   0.07     0.00     1          0.00     0.00  iniciaMemLog
  0.00   0.07     0.00     1          0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 14.26% of 0.07 seconds

index % time    self  children    called     name
```

```
                                      <spontaneous>
[1]     100.0   0.00    0.07                  main [1]
                0.02    0.01    1/1                     inicializaVetorAleatorio [2]
                0.02    0.00    1/1                     acessaVetor [3]
                0.02    0.00    1/1                     normaVetor [4]
                0.00    0.00    3/3                     defineFaseMemLog [6]
                0.00    0.00    1/1                     parse_args [13]
                0.00    0.00    1/1                     iniciaMemLog [12]
                0.00    0.00    1/1                     desativaMemLog [9]
                0.00    0.00    1/1                     criaVetor [8]
                0.00    0.00    1/1                     destroiVetor [10]
                0.00    0.00    1/1                     finalizaMemLog [11]
-----------------------------------------------
                0.02    0.01    1/1                     main [1]
[2]     42.9    0.02    0.01    1             inicializaVetorAleatorio [2]
                0.01    0.00    1/1                     inicializaVetorNulo [5]
-----------------------------------------------
                0.02    0.00    1/1                     main [1]
[3]     28.6    0.02    0.00    1             acessaVetor [3]
-----------------------------------------------
                0.02    0.00    1/1                     main [1]
[4]     28.6    0.02    0.00    1             normaVetor [4]
-----------------------------------------------
                0.01    0.00    1/1                     inicializaVetorAleatorio [2]
[5]     14.3    0.01    0.00    1             inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00    3/3                     main [1]
[6]     0.0     0.00    0.00    3             defineFaseMemLog [6]
-----------------------------------------------
                0.00    0.00    1/1                     finalizaMemLog [11]
[7]     0.0     0.00    0.00    1             clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1                     main [1]
[8]     0.0     0.00    0.00    1             criaVetor [8]
-----------------------------------------------
                0.00    0.00    1/1                     main [1]
[9]     0.0     0.00    0.00    1             desativaMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1                     main [1]
[10]    0.0     0.00    0.00    1             destroiVetor [10]
-----------------------------------------------
                0.00    0.00    1/1                     main [1]
[11]    0.0     0.00    0.00    1             finalizaMemLog [11]
                0.00    0.00    1/1                     clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1                     main [1]
[12]    0.0     0.00    0.00    1             iniciaMemLog [12]
-----------------------------------------------
```

```
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1           parse_args [13]
-----------------------------------------------
```

```
 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index	A unique number given to each element of the table.
		Index numbers are sorted numerically.
		The index number is printed next to every function name so
		it is easier to look up where the function is in the table.

     % time	This is the percentage of the `total' time that was spent
		in this function and its children.  Note that due to
		different viewpoints, functions excluded by options, etc,
		these numbers will NOT add up to 100%.

     self	This is the total amount of time spent in this function.

     children	This is the total amount of time propagated into this
		function by its children.

     called	This is the number of times the function was called.
		If the function called itself recursively, the number
		only includes non-recursive calls, and is followed by
		a `+' and the number of recursive calls.

     name	The name of the current function.  The index number is
		printed after it.  If the function is a member of a
		cycle, the cycle number is printed between the
		function's name and the index number.


 For the function's parents, the fields have the following meanings:

     self	This is the amount of time that was propagated directly
		from the function into this parent.

     children	This is the amount of time that was propagated from
		the function's children into this parent.

     called	This is the number of times this parent called the
		function `/' the total number of times the function
```

was called. Recursive calls to the function are not
included in the number after the `/'.

name This is the name of the parent. The parent's index
number is printed after it. If the parent is a
member of a cycle, the cycle number is printed between
the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly
from the child into the function.

children This is the amount of time that was propagated from the
child's children to the function.

called This is the number of times the function called
this child `/' the total number of times the child
was called. Recursive calls by the child are not
listed in the number after the `/'.

name This is the name of the child. The child's index
number is printed after it. If the child is a
member of a cycle, the cycle number is printed
between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

Index by function name

[3] acessaVetor	[9] desativaMemLog	[2] inicializaVetorAleatorio
[7] clkDifMemLog	[10] destroiVetor	[5] inicializaVetorNulo

```
  [8] criaVetor            [11] finalizaMemLog             [4] normaVetor
  [6] defineFaseMemLog       [12] iniciaMemLog               [13] parse_args
```

5.1.2.5. 5M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 33.39      0.02     0.02        1    20.04    20.04  acessaVetor
 33.39      0.04     0.02        1    20.04    30.05  inicializaVetorAleatorio
 16.70      0.05     0.01        1    10.02    10.02  inicializaVetorNulo
 16.70      0.06     0.01        1    10.02    10.02  normaVetor
  0.00      0.06     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.06     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.06     0.00        1     0.00     0.00  criaVetor
  0.00      0.06     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.06     0.00        1     0.00     0.00  destroiVetor
  0.00      0.06     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.06     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.06     0.00        1     0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
	   else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
	   function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
```

```
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 16.64% of 0.06 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.06                 main [1]
                0.02    0.01       1/1           inicializaVetorAleatorio [2]
                0.02    0.00       1/1           acessaVetor [3]
                0.01    0.00       1/1           normaVetor [5]
                0.00    0.00       3/3           defineFaseMemLog [6]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [12]
                0.00    0.00       1/1           desativaMemLog [9]
                0.00    0.00       1/1           criaVetor [8]
                0.00    0.00       1/1           destroiVetor [10]
                0.00    0.00       1/1           finalizaMemLog [11]
-----------------------------------------------
                0.02    0.01       1/1           main [1]
[2]     50.0    0.02    0.01       1         inicializaVetorAleatorio [2]
                0.01    0.00       1/1           inicializaVetorNulo [4]
-----------------------------------------------
                0.02    0.00       1/1           main [1]
[3]     33.3    0.02    0.00       1         acessaVetor [3]
-----------------------------------------------
                0.01    0.00       1/1           inicializaVetorAleatorio [2]
[4]     16.7    0.01    0.00       1         inicializaVetorNulo [4]
-----------------------------------------------
                0.01    0.00       1/1           main [1]
[5]     16.7    0.01    0.00       1         normaVetor [5]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[6]      0.0    0.00    0.00       3         defineFaseMemLog [6]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [11]
[7]      0.0    0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
```

```
                0.00    0.00    1/1             main [1]
[8]     0.0     0.00    0.00    1           criaVetor [8]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[9]     0.0     0.00    0.00    1           desativaMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1           destroiVetor [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1           finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1           iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1           parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.
```

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

children   This is the amount of time that was propagated from the
 child's children to the function.

called   This is the number of times the function called
 this child `/' the total number of times the child
 was called. Recursive calls by the child are not
 listed in the number after the `/'.

name   This is the name of the child. The child's index
 number is printed after it. If the child is a
 member of a cycle, the cycle number is printed
 between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the

```
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [3] acessaVetor              [9] desativaMemLog          [2] inicializaVetorAleatorio
   [7] clkDifMemLog             [10] destroiVetor       [4] inicializaVetorNulo
   [8] criaVetor           [11] finalizaMemLog          [5] normaVetor
   [6] defineFaseMemLog      [12] iniciaMemLog            [13] parse_args
```

5.1.3. Soma

5.1.3.1. 1M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 33.39      0.01     0.01        4     2.50     2.50  acessaVetor
 33.39      0.02     0.01        4     2.50     2.50  inicializaVetorNulo
 33.39      0.03     0.01        1    10.02    12.52  somaVetores
  0.00      0.03     0.00        4     0.00     0.00  criaVetor
  0.00      0.03     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.03     0.00        3     0.00     0.00  destroiVetor
  0.00      0.03     0.00        2     0.00     2.50  inicializaVetorAleatorio
  0.00      0.03     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.03     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.03     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.03     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.03     0.00        1     0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.
 self     the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
          listing.

calls     the number of times this function was invoked, if
          this function is profiled, else blank.

 self     the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
          else blank.

 total    the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
          function is profiled, else blank.

name             the name of the function.  This is the minor sort
          for this listing. The index shows the location of
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 33.27% of 0.03 seconds

index % time    self  children    called   name
                                           <spontaneous>
[1]     100.0    0.00    0.03               main [1]
                 0.01    0.00     1/1           somaVetores [2]
                 0.01    0.00     4/4           acessaVetor [3]
                 0.00    0.01     2/2           inicializaVetorAleatorio [5]
                 0.00    0.00     1/4           inicializaVetorNulo [4]
                 0.00    0.00     3/3           defineFaseMemLog [7]
                 0.00    0.00     3/4           criaVetor [6]
                 0.00    0.00     3/3           destroiVetor [8]
                 0.00    0.00     1/1           parse_args [13]
                 0.00    0.00     1/1           iniciaMemLog [12]
                 0.00    0.00     1/1           desativaMemLog [10]
                 0.00    0.00     1/1           finalizaMemLog [11]
-----------------------------------------------
                 0.01    0.00     1/1           main [1]
```

```
[2]     41.7    0.01    0.00    1       somaVetores [2]
                0.00    0.00    1/4             inicializaVetorNulo [4]
                0.00    0.00    1/4             criaVetor [6]
-----------------------------------------------
                0.01    0.00    4/4             main [1]
[3]     33.3    0.01    0.00    4       acessaVetor [3]
-----------------------------------------------
                0.00    0.00    1/4             main [1]
                0.00    0.00    1/4             somaVetores [2]
                0.01    0.00    2/4             inicializaVetorAleatorio [5]
[4]     33.3    0.01    0.00    4       inicializaVetorNulo [4]
-----------------------------------------------
                0.00    0.01    2/2             main [1]
[5]     16.7    0.00    0.01    2       inicializaVetorAleatorio [5]
                0.01    0.00    2/4             inicializaVetorNulo [4]
-----------------------------------------------
                0.00    0.00    1/4             somaVetores [2]
                0.00    0.00    3/4             main [1]
[6]     0.0     0.00    0.00    4       criaVetor [6]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[7]     0.0     0.00    0.00    3       defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[8]     0.0     0.00    0.00    3       destroiVetor [8]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [11]
[9]     0.0     0.00    0.00    1       clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1       desativaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1       finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1       iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1       parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
```

The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:

index   A unique number given to each element of the table.
Index numbers are sorted numerically.
The index number is printed next to every function name so
it is easier to look up where the function is in the table.

% time   This is the percentage of the `total' time that was spent
in this function and its children. Note that due to
different viewpoints, functions excluded by options, etc,
these numbers will NOT add up to 100%.

self   This is the total amount of time spent in this function.

children   This is the total amount of time propagated into this
function by its children.

called   This is the number of times the function was called.
If the function called itself recursively, the number
only includes non-recursive calls, and is followed by
a `+' and the number of recursive calls.

name   The name of the current function. The index number is
printed after it. If the function is a member of a
cycle, the cycle number is printed between the
function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
from the function into this parent.

children   This is the amount of time that was propagated from
the function's children into this parent.

called   This is the number of times this parent called the
function `/' the total number of times the function
was called. Recursive calls to the function are not
included in the number after the `/'.

name   This is the name of the parent. The parent's index
number is printed after it. If the parent is a
member of a cycle, the cycle number is printed between
the name and the index number.

If the parents of the function cannot be determined, the word

```
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

     self    This is the amount of time that was propagated directly
      from the child into the function.

     children    This is the amount of time that was propagated from the
      child's children to the function.

     called    This is the number of times the function called
      this child `/' the total number of times the child
      was called.  Recursive calls by the child are not
      listed in the number after the `/'.

     name    This is the name of the child.  The child's index
      number is printed after it.  If the child is a
      member of a cycle, the cycle number is printed
      between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole.  This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

  [3] acessaVetor            [10] desativaMemLog         [5] inicializaVetorAleatorio
  [9] clkDifMemLog            [8] destroiVetor           [4] inicializaVetorNulo
  [6] criaVetor              [11] finalizaMemLog        [13] parse_args
  [7] defineFaseMemLog       [12] iniciaMemLog           [2] somaVetores
```

5.1.3.2. 2M

```
Flat profile:
```

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 53.04          0.09     0.09     4        22.54   22.54  acessaVetor
 23.57          0.13     0.04     1        40.07   47.59  somaVetores
 17.68          0.16     0.03     4        7.51     7.51  inicializaVetorNulo
  5.89   0.17     0.01     2        5.01      12.52  inicializaVetorAleatorio
  0.00   0.17     0.00     4        0.00      0.00  criaVetor
  0.00   0.17     0.00     3        0.00      0.00  defineFaseMemLog
  0.00   0.17     0.00     3        0.00      0.00  destroiVetor
  0.00   0.17     0.00     1        0.00      0.00  clkDifMemLog
  0.00   0.17     0.00     1        0.00      0.00  desativaMemLog
  0.00   0.17     0.00     1        0.00      0.00  finalizaMemLog
  0.00   0.17     0.00     1        0.00      0.00  iniciaMemLog
  0.00   0.17     0.00     1        0.00      0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name             the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 5.87% of 0.17 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.17                 main [1]
                0.09    0.00       4/4           acessaVetor [2]
                0.04    0.01       1/1           somaVetores [3]
                0.01    0.02       2/2           inicializaVetorAleatorio [5]
                0.01    0.00       1/4           inicializaVetorNulo [4]
                0.00    0.00       3/3           defineFaseMemLog [7]
                0.00    0.00       3/4           criaVetor [6]
                0.00    0.00       3/3           destroiVetor [8]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [12]
                0.00    0.00       1/1           desativaMemLog [10]
                0.00    0.00       1/1           finalizaMemLog [11]
-----------------------------------------------
                0.09    0.00       4/4           main [1]
[2]     52.9    0.09    0.00       4         acessaVetor [2]
-----------------------------------------------
                0.04    0.01       1/1           main [1]
[3]     27.9    0.04    0.01       1         somaVetores [3]
                0.01    0.00       1/4           inicializaVetorNulo [4]
                0.00    0.00       1/4           criaVetor [6]
-----------------------------------------------
                0.01    0.00       1/4           main [1]
                0.01    0.00       1/4           somaVetores [3]
                0.02    0.00       2/4           inicializaVetorAleatorio [5]
[4]     17.6    0.03    0.00       4         inicializaVetorNulo [4]
-----------------------------------------------
                0.01    0.02       2/2           main [1]
[5]     14.7    0.01    0.02       2         inicializaVetorAleatorio [5]
                0.02    0.00       2/4           inicializaVetorNulo [4]
-----------------------------------------------
                0.00    0.00       1/4           somaVetores [3]
                0.00    0.00       3/4           main [1]
[6]      0.0    0.00    0.00       4         criaVetor [6]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[7]      0.0    0.00    0.00       3         defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00       3/3           main [1]
[8]      0.0    0.00    0.00       3         destroiVetor [8]
-----------------------------------------------
```

```
                0.00    0.00     1/1              finalizaMemLog [11]
[9]     0.0     0.00    0.00     1         clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[10]    0.0     0.00    0.00     1         desativaMemLog [10]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[11]    0.0     0.00    0.00     1         finalizaMemLog [11]
                0.00    0.00     1/1              clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[12]    0.0     0.00    0.00     1         iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00     1/1              main [1]
[13]    0.0     0.00    0.00     1         parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.

     name    The name of the current function.  The index number is
      printed after it.  If the function is a member of a
      cycle, the cycle number is printed between the
```

function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly from the function into this parent.

children   This is the amount of time that was propagated from the function's children into this parent.

called   This is the number of times this parent called the function `/' the total number of times the function was called.  Recursive calls to the function are not included in the number after the `/'.

name   This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly from the child into the function.

children   This is the amount of time that was propagated from the child's children to the function.

called   This is the number of times the function called this child `/' the total number of times the child was called.  Recursive calls by the child are not listed in the number after the `/'.

name   This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,

```
 for that member, how many times it was called from other members of
 the cycle.
```

```
Index by function name

   [2] acessaVetor            [10] desativaMemLog         [5] inicializaVetorAleatorio
   [9] clkDifMemLog            [8] destroiVetor           [4] inicializaVetorNulo
   [6] criaVetor              [11] finalizaMemLog        [13] parse_args
   [7] defineFaseMemLog       [12] iniciaMemLog           [3] somaVetores
```

5.1.3.3. 3M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 40.07      0.04     0.04        4    10.02    10.02  acessaVetor
 30.05      0.07     0.03        4     7.51     7.51  inicializaVetorNulo
 20.04      0.09     0.02        2    10.02    17.53  inicializaVetorAleatorio
 10.02      0.10     0.01        1    10.02    17.53  somaVetores
  0.00      0.10     0.00        4     0.00     0.00  criaVetor
  0.00      0.10     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.10     0.00        3     0.00     0.00  destroiVetor
  0.00      0.10     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.10     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.10     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.10     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.10     0.00        1     0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.
```

```
calls     the number of times this function was invoked, if
          this function is profiled, else blank.

 self     the average number of milliseconds spent in this
ms/call   function per call, if this function is profiled,
          else blank.

 total    the average number of milliseconds spent in this
ms/call   function and its descendents per call, if this
          function is profiled, else blank.

name            the name of the function.  This is the minor sort
          for this listing. The index shows the location of
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 9.98% of 0.10 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.10                 main [1]
                0.04    0.00       4/4           acessaVetor [2]
                0.02    0.02       2/2           inicializaVetorAleatorio [3]
                0.01    0.01       1/1           somaVetores [5]
                0.01    0.00       1/4           inicializaVetorNulo [4]
                0.00    0.00       3/3           defineFaseMemLog [7]
                0.00    0.00       3/4           criaVetor [6]
                0.00    0.00       3/3           destroiVetor [8]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [12]
                0.00    0.00       1/1           desativaMemLog [10]
                0.00    0.00       1/1           finalizaMemLog [11]
-----------------------------------------------
                0.04    0.00       4/4           main [1]
[2]     40.0    0.04    0.00       4         acessaVetor [2]
-----------------------------------------------
                0.02    0.02       2/2           main [1]
[3]     35.0    0.02    0.02       2         inicializaVetorAleatorio [3]
```

```
                0.02    0.00    2/4             inicializaVetorNulo [4]
-----------------------------------------------
                0.01    0.00    1/4             main [1]
                0.01    0.00    1/4             somaVetores [5]
                0.02    0.00    2/4             inicializaVetorAleatorio [3]
[4]     30.0    0.03    0.00    4           inicializaVetorNulo [4]
-----------------------------------------------
                0.01    0.01    1/1             main [1]
[5]     17.5    0.01    0.01    1           somaVetores [5]
                0.01    0.00    1/4             inicializaVetorNulo [4]
                0.00    0.00    1/4             criaVetor [6]
-----------------------------------------------
                0.00    0.00    1/4             somaVetores [5]
                0.00    0.00    3/4             main [1]
[6]     0.0     0.00    0.00    4           criaVetor [6]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[7]     0.0     0.00    0.00    3           defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[8]     0.0     0.00    0.00    3           destroiVetor [8]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [11]
[9]     0.0     0.00    0.00    1           clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1           desativaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1           finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1           iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1           parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index      A unique number given to each element of the table.
```

Index numbers are sorted numerically.
The index number is printed next to every function name so
it is easier to look up where the function is in the table.

% time   This is the percentage of the `total' time that was spent
in this function and its children. Note that due to
different viewpoints, functions excluded by options, etc,
these numbers will NOT add up to 100%.

self   This is the total amount of time spent in this function.

children   This is the total amount of time propagated into this
function by its children.

called   This is the number of times the function was called.
If the function called itself recursively, the number
only includes non-recursive calls, and is followed by
a `+' and the number of recursive calls.

name   The name of the current function. The index number is
printed after it. If the function is a member of a
cycle, the cycle number is printed between the
function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
from the function into this parent.

children   This is the amount of time that was propagated from
the function's children into this parent.

called   This is the number of times this parent called the
function `/' the total number of times the function
was called. Recursive calls to the function are not
included in the number after the `/'.

name   This is the name of the parent. The parent's index
number is printed after it. If the parent is a
member of a cycle, the cycle number is printed between
the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

```
          self    This is the amount of time that was propagated directly
           from the child into the function.

          children    This is the amount of time that was propagated from the
           child's children to the function.

          called    This is the number of times the function called
           this child `/' the total number of times the child
           was called.  Recursive calls by the child are not
           listed in the number after the `/'.

          name    This is the name of the child.  The child's index
           number is printed after it.  If the child is a
           member of a cycle, the cycle number is printed
           between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [2] acessaVetor             [10] desativaMemLog          [3] inicializaVetorAleatorio
   [9] clkDifMemLog             [8] destroiVetor            [4] inicializaVetorNulo
   [6] criaVetor               [11] finalizaMemLog         [13] parse_args
   [7] defineFaseMemLog        [12] iniciaMemLog            [5] somaVetores
```

5.1.3.4. 4M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 38.53      0.05     0.05        4    12.52    12.52  acessaVetor
 30.82         0.09   0.04    4      10.02   10.02  inicializaVetorNulo
 15.41         0.11   0.02    2      10.02   20.04  inicializaVetorAleatorio
 15.41         0.13   0.02    1      20.04   30.05  somaVetores
  0.00   0.13  0.00   4      0.00    0.00  criaVetor
  0.00   0.13  0.00   3      0.00    0.00  defineFaseMemLog
  0.00   0.13  0.00   3      0.00    0.00  destroiVetor
  0.00   0.13  0.00   1      0.00    0.00  clkDifMemLog
  0.00   0.13  0.00   1      0.00    0.00  desativaMemLog
  0.00   0.13  0.00   1      0.00    0.00  finalizaMemLog
  0.00   0.13  0.00   1      0.00    0.00  iniciaMemLog
  0.00   0.13  0.00   1      0.00    0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                     Call graph (explanation follows)
```

```
granularity: each sample hit covers 2 byte(s) for 7.68% of 0.13 seconds

index % time    self  children    called     name
                                             <spontaneous>
[1]     100.0    0.00    0.13                main [1]
                 0.05    0.00       4/4          acessaVetor [2]
                 0.02    0.02       2/2          inicializaVetorAleatorio [4]
                 0.02    0.01       1/1          somaVetores [5]
                 0.01    0.00       1/4          inicializaVetorNulo [3]
                 0.00    0.00       3/3          defineFaseMemLog [7]
                 0.00    0.00       3/4          criaVetor [6]
                 0.00    0.00       3/3          destroiVetor [8]
                 0.00    0.00       1/1          parse_args [13]
                 0.00    0.00       1/1          iniciaMemLog [12]
                 0.00    0.00       1/1          desativaMemLog [10]
                 0.00    0.00       1/1          finalizaMemLog [11]
-----------------------------------------------
                 0.05    0.00       4/4          main [1]
[2]      38.5    0.05    0.00       4        acessaVetor [2]
-----------------------------------------------
                 0.01    0.00       1/4          main [1]
                 0.01    0.00       1/4          somaVetores [5]
                 0.02    0.00       2/4          inicializaVetorAleatorio [4]
[3]      30.8    0.04    0.00       4        inicializaVetorNulo [3]
-----------------------------------------------
                 0.02    0.02       2/2          main [1]
[4]      30.8    0.02    0.02       2        inicializaVetorAleatorio [4]
                 0.02    0.00       2/4          inicializaVetorNulo [3]
-----------------------------------------------
                 0.02    0.01       1/1          main [1]
[5]      23.1    0.02    0.01       1        somaVetores [5]
                 0.01    0.00       1/4          inicializaVetorNulo [3]
                 0.00    0.00       1/4          criaVetor [6]
-----------------------------------------------
                 0.00    0.00       1/4          somaVetores [5]
                 0.00    0.00       3/4          main [1]
[6]       0.0    0.00    0.00       4        criaVetor [6]
-----------------------------------------------
                 0.00    0.00       3/3          main [1]
[7]       0.0    0.00    0.00       3        defineFaseMemLog [7]
-----------------------------------------------
                 0.00    0.00       3/3          main [1]
[8]       0.0    0.00    0.00       3        destroiVetor [8]
-----------------------------------------------
                 0.00    0.00       1/1          finalizaMemLog [11]
[9]       0.0    0.00    0.00       1        clkDifMemLog [9]
-----------------------------------------------
                 0.00    0.00       1/1          main [1]
```

```
[10]    0.0     0.00    0.00    1           desativaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1           finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1           iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1           parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index      A unique number given to each element of the table.
                Index numbers are sorted numerically.
                The index number is printed next to every function name so
                it is easier to look up where the function is in the table.

     % time     This is the percentage of the `total' time that was spent
                in this function and its children.  Note that due to
                different viewpoints, functions excluded by options, etc,
                these numbers will NOT add up to 100%.

     self       This is the total amount of time spent in this function.

     children   This is the total amount of time propagated into this
                function by its children.

     called     This is the number of times the function was called.
                If the function called itself recursively, the number
                only includes non-recursive calls, and is followed by
                a `+' and the number of recursive calls.

     name       The name of the current function.  The index number is
                printed after it.  If the function is a member of a
                cycle, the cycle number is printed between the
                function's name and the index number.


 For the function's parents, the fields have the following meanings:
```

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

children   This is the amount of time that was propagated from the
 child's children to the function.

called   This is the number of times the function called
 this child `/' the total number of times the child
 was called. Recursive calls by the child are not
 listed in the number after the `/'.

name   This is the name of the child. The child's index
 number is printed after it. If the child is a
 member of a cycle, the cycle number is printed
 between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

```
Index by function name

   [2] acessaVetor              [10] desativaMemLog          [4] inicializaVetorAleatorio
   [9] clkDifMemLog              [8] destroiVetor         [3] inicializaVetorNulo
   [6] criaVetor            [11] finalizaMemLog           [13] parse_args
   [7] defineFaseMemLog      [12] iniciaMemLog              [5] somaVetores
```

5.1.3.5. 5M

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total           
 time   seconds   seconds    calls  ms/call  ms/call  name    
 40.98            0.09     0.09     4         22.54   22.54  inicializaVetorNulo
 36.43            0.17     0.08     4         20.04   20.04  acessaVetor
 13.66            0.20     0.03     2         15.03   37.57  inicializaVetorAleatorio
  9.11   0.22     0.02     1        20.04    42.58  somaVetores
  0.00   0.22     0.00     4        0.00     0.00  criaVetor
  0.00   0.22     0.00     3        0.00     0.00  defineFaseMemLog
  0.00   0.22     0.00     3        0.00     0.00  destroiVetor
  0.00   0.22     0.00     1        0.00     0.00  clkDifMemLog
  0.00   0.22     0.00     1        0.00     0.00  desativaMemLog
  0.00   0.22     0.00     1        0.00     0.00  finalizaMemLog
  0.00   0.22     0.00     1        0.00     0.00  iniciaMemLog
  0.00   0.22     0.00     1        0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds           function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
```

```
ms/call function per call, if this function is profiled,
        else blank.

 total  the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
        function is profiled, else blank.

name            the name of the function.  This is the minor sort
        for this listing. The index shows the location of
        the function in the gprof listing. If the index is
        in parenthesis it shows where it would appear in
        the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) for 4.54% of 0.22 seconds

index % time    self  children    called     name
                                                 <spontaneous>
[1]    100.0    0.00    0.22                 main [1]
                0.08    0.00       4/4           acessaVetor [3]
                0.03    0.05       2/2           inicializaVetorAleatorio [4]
                0.02    0.02       1/1           somaVetores [5]
                0.02    0.00       1/4           inicializaVetorNulo [2]
                0.00    0.00       3/3           defineFaseMemLog [7]
                0.00    0.00       3/4           criaVetor [6]
                0.00    0.00       3/3           destroiVetor [8]
                0.00    0.00       1/1           parse_args [13]
                0.00    0.00       1/1           iniciaMemLog [12]
                0.00    0.00       1/1           desativaMemLog [10]
                0.00    0.00       1/1           finalizaMemLog [11]
-----------------------------------------------
                0.02    0.00       1/4           main [1]
                0.02    0.00       1/4           somaVetores [5]
                0.05    0.00       2/4           inicializaVetorAleatorio [4]
[2]     40.9    0.09    0.00       4         inicializaVetorNulo [2]
-----------------------------------------------
                0.08    0.00       4/4           main [1]
[3]     36.4    0.08    0.00       4         acessaVetor [3]
-----------------------------------------------
                0.03    0.05       2/2           main [1]
```

```
[4]     34.1    0.03    0.05    2       inicializaVetorAleatorio [4]
                0.05    0.00    2/4             inicializaVetorNulo [2]
-----------------------------------------------
                0.02    0.02    1/1             main [1]
[5]     19.3    0.02    0.02    1       somaVetores [5]
                0.02    0.00    1/4             inicializaVetorNulo [2]
                0.00    0.00    1/4             criaVetor [6]
-----------------------------------------------
                0.00    0.00    1/4             somaVetores [5]
                0.00    0.00    3/4             main [1]
[6]     0.0     0.00    0.00    4       criaVetor [6]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[7]     0.0     0.00    0.00    3       defineFaseMemLog [7]
-----------------------------------------------
                0.00    0.00    3/3             main [1]
[8]     0.0     0.00    0.00    3       destroiVetor [8]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [11]
[9]     0.0     0.00    0.00    1       clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[10]    0.0     0.00    0.00    1       desativaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[11]    0.0     0.00    0.00    1       finalizaMemLog [11]
                0.00    0.00    1/1             clkDifMemLog [9]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[12]    0.0     0.00    0.00    1       iniciaMemLog [12]
-----------------------------------------------
                0.00    0.00    1/1             main [1]
[13]    0.0     0.00    0.00    1       parse_args [13]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.
```

% time   This is the percentage of the `total' time that was spent
 in this function and its children. Note that due to
 different viewpoints, functions excluded by options, etc,
 these numbers will NOT add up to 100%.

self   This is the total amount of time spent in this function.

children   This is the total amount of time propagated into this
 function by its children.

called   This is the number of times the function was called.
 If the function called itself recursively, the number
 only includes non-recursive calls, and is followed by
 a `+' and the number of recursive calls.

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

```
           children    This is the amount of time that was propagated from the
            child's children to the function.

           called    This is the number of times the function called
            this child `/' the total number of times the child
            was called.  Recursive calls by the child are not
            listed in the number after the `/'.

           name    This is the name of the child.  The child's index
            number is printed after it.  If the child is a
            member of a cycle, the cycle number is printed
            between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.
```

```
Index by function name

   [3] acessaVetor            [10] desativaMemLog          [4] inicializaVetorAleatorio
   [9] clkDifMemLog            [8] destroiVetor            [2] inicializaVetorNulo
   [6] criaVetor              [11] finalizaMemLog         [13] parse_args
   [7] defineFaseMemLog       [12] iniciaMemLog            [5] somaVetores
```

5.2. Vetor Estático

5.2.1. Interno

5.2.1.1. 100

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total           
 time   seconds   seconds    calls  Ts/call  Ts/call  name    
```

```
  0.00   0.00     0.00     3         0.00     0.00  acessaVetor
  0.00   0.00     0.00     3         0.00     0.00  criaVetor
  0.00   0.00     0.00     3         0.00     0.00  defineFaseMemLog
  0.00   0.00     0.00     3         0.00     0.00  destroiVetor
  0.00   0.00     0.00     3         0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00     2         0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00     1         0.00     0.00  clkDifMemLog
  0.00   0.00     0.00     1         0.00     0.00  desativaMemLog
  0.00   0.00     0.00     1         0.00     0.00  finalizaMemLog
  0.00   0.00     0.00     1         0.00     0.00  iniciaMemLog
  0.00   0.00     0.00     1         0.00     0.00  parse_args
  0.00   0.00     0.00     1         0.00     0.00  produtoInternoVetores

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                     Call graph (explanation follows)
```

```
granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]     0.0     0.00    0.00       3         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[2]     0.0     0.00    0.00       3         criaVetor [2]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[3]     0.0     0.00    0.00       3         defineFaseMemLog [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]     0.0     0.00    0.00       3         destroiVetor [4]
-----------------------------------------------
                0.00    0.00       1/3           main [25]
                0.00    0.00       2/3           inicializaVetorAleatorio [6]
[5]     0.0     0.00    0.00       3         inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]     0.0     0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/3           inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[7]     0.0     0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]     0.0     0.00    0.00       1         desativaMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[9]     0.0     0.00    0.00       1         finalizaMemLog [9]
                0.00    0.00       1/1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[10]    0.0     0.00    0.00       1         iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[11]    0.0     0.00    0.00       1         parse_args [11]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[12]    0.0     0.00    0.00       1         produtoInternoVetores [12]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
```

```
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.

     name    The name of the current function.  The index number is
      printed after it.  If the function is a member of a
      cycle, the cycle number is printed between the
      function's name and the index number.


For the function's parents, the fields have the following meanings:

     self    This is the amount of time that was propagated directly
      from the function into this parent.

     children    This is the amount of time that was propagated from
      the function's children into this parent.

     called    This is the number of times this parent called the
      function `/' the total number of times the function
      was called.  Recursive calls to the function are not
      included in the number after the `/'.

     name    This is the name of the parent.  The parent's index
      number is printed after it.  If the parent is a
      member of a cycle, the cycle number is printed between
      the name and the index number.
```

```
 If the parents of the function cannot be determined, the word
 `<spontaneous>' is printed in the `name' field, and all the other
 fields are blank.

 For the function's children, the fields have the following meanings:

           self    This is the amount of time that was propagated directly
            from the child into the function.

           children    This is the amount of time that was propagated from the
            child's children to the function.

           called    This is the number of times the function called
            this child `/' the total number of times the child
            was called.  Recursive calls by the child are not
            listed in the number after the `/'.

           name    This is the name of the child.  The child's index
            number is printed after it.  If the child is a
            member of a cycle, the cycle number is printed
            between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

  [1] acessaVetor                  [8] desativaMemLog              [6] inicializaVetorAleatorio
  [7] clkDifMemLog                 [4] destroiVetor           [5] inicializaVetorNulo
  [2] criaVetor             [9] finalizaMemLog     [11] parse_args
  [3] defineFaseMemLog        [10] iniciaMemLog                [12] produtoInternoVetores
```

5.2.1.2. 200

```
Flat profile:
```

```
Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total           
 time   seconds   seconds    calls  Ts/call  Ts/call  name    
  0.00   0.00     0.00     3        0.00     0.00  acessaVetor
  0.00   0.00     0.00     3        0.00     0.00  criaVetor
  0.00   0.00     0.00     3        0.00     0.00  defineFaseMemLog
  0.00   0.00     0.00     3        0.00     0.00  destroiVetor
  0.00   0.00     0.00     3        0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00     2        0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00     1        0.00     0.00  clkDifMemLog
  0.00   0.00     0.00     1        0.00     0.00  desativaMemLog
  0.00   0.00     0.00     1        0.00     0.00  finalizaMemLog
  0.00   0.00     0.00     1        0.00     0.00  iniciaMemLog
  0.00   0.00     0.00     1        0.00     0.00  parse_args
  0.00   0.00     0.00     1        0.00     0.00  produtoInternoVetores

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]      0.0    0.00    0.00       3         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[2]      0.0    0.00    0.00       3         criaVetor [2]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[3]      0.0    0.00    0.00       3         defineFaseMemLog [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]      0.0    0.00    0.00       3         destroiVetor [4]
-----------------------------------------------
                0.00    0.00       1/3           main [25]
                0.00    0.00       2/3           inicializaVetorAleatorio [6]
[5]      0.0    0.00    0.00       3         inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]      0.0    0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/3           inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[7]      0.0    0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]      0.0    0.00    0.00       1         desativaMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[9]      0.0    0.00    0.00       1         finalizaMemLog [9]
                0.00    0.00       1/1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[10]     0.0    0.00    0.00       1         iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[11]     0.0    0.00    0.00       1         parse_args [11]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[12]     0.0    0.00    0.00       1         produtoInternoVetores [12]
```

```
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.

     called    This is the number of times the function was called.
      If the function called itself recursively, the number
      only includes non-recursive calls, and is followed by
      a `+' and the number of recursive calls.

     name    The name of the current function.  The index number is
      printed after it.  If the function is a member of a
      cycle, the cycle number is printed between the
      function's name and the index number.


 For the function's parents, the fields have the following meanings:

     self    This is the amount of time that was propagated directly
      from the function into this parent.

     children    This is the amount of time that was propagated from
      the function's children into this parent.

     called    This is the number of times this parent called the
      function `/' the total number of times the function
      was called.  Recursive calls to the function are not
      included in the number after the `/'.
```

```
          name    This is the name of the parent.  The parent's index
           number is printed after it.  If the parent is a
           member of a cycle, the cycle number is printed between
           the name and the index number.

 If the parents of the function cannot be determined, the word
 `<spontaneous>' is printed in the `name' field, and all the other
 fields are blank.

 For the function's children, the fields have the following meanings:

          self    This is the amount of time that was propagated directly
           from the child into the function.

          children    This is the amount of time that was propagated from the
           child's children to the function.

          called    This is the number of times the function called
           this child `/' the total number of times the child
           was called.  Recursive calls by the child are not
           listed in the number after the `/'.

          name    This is the name of the child.  The child's index
           number is printed after it.  If the child is a
           member of a cycle, the cycle number is printed
           between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

  [1] acessaVetor             [8] desativaMemLog          [6] inicializaVetorAleatorio
  [7] clkDifMemLog            [4] destroiVetor            [5] inicializaVetorNulo
  [2] criaVetor               [9] finalizaMemLog         [11] parse_args
  [3] defineFaseMemLog       [10] iniciaMemLog           [12] produtoInternoVetores
```

5.2.1.3. 300

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00   0.00     0.00    3          0.00     0.00  acessaVetor
  0.00   0.00     0.00    3          0.00     0.00  criaVetor
  0.00   0.00     0.00    3          0.00     0.00  defineFaseMemLog
  0.00   0.00     0.00    3          0.00     0.00  destroiVetor
  0.00   0.00     0.00    3          0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00    2          0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00    1          0.00     0.00  clkDifMemLog
  0.00   0.00     0.00    1          0.00     0.00  desativaMemLog
  0.00   0.00     0.00    1          0.00     0.00  finalizaMemLog
  0.00   0.00     0.00    1          0.00     0.00  iniciaMemLog
  0.00   0.00     0.00    1          0.00     0.00  parse_args
  0.00   0.00     0.00    1          0.00     0.00  produtoInternoVetores

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
```

```
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.
```

```
                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]      0.0    0.00    0.00       3         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[2]      0.0    0.00    0.00       3         criaVetor [2]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[3]      0.0    0.00    0.00       3         defineFaseMemLog [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]      0.0    0.00    0.00       3         destroiVetor [4]
-----------------------------------------------
                0.00    0.00       1/3           main [25]
                0.00    0.00       2/3           inicializaVetorAleatorio [6]
[5]      0.0    0.00    0.00       3         inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]      0.0    0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/3           inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[7]      0.0    0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]      0.0    0.00    0.00       1         desativaMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[9]      0.0    0.00    0.00       1         finalizaMemLog [9]
                0.00    0.00       1/1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[10]     0.0    0.00    0.00       1         iniciaMemLog [10]
```

```
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1           parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1           produtoInternoVetores [12]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index      A unique number given to each element of the table.
                Index numbers are sorted numerically.
                The index number is printed next to every function name so
                it is easier to look up where the function is in the table.

     % time     This is the percentage of the `total' time that was spent
                in this function and its children.  Note that due to
                different viewpoints, functions excluded by options, etc,
                these numbers will NOT add up to 100%.

     self       This is the total amount of time spent in this function.

     children   This is the total amount of time propagated into this
                function by its children.

     called     This is the number of times the function was called.
                If the function called itself recursively, the number
                only includes non-recursive calls, and is followed by
                a `+' and the number of recursive calls.

     name       The name of the current function.  The index number is
                printed after it.  If the function is a member of a
                cycle, the cycle number is printed between the
                function's name and the index number.


 For the function's parents, the fields have the following meanings:

     self       This is the amount of time that was propagated directly
                from the function into this parent.

     children   This is the amount of time that was propagated from
```

the function's children into this parent.

called   This is the number of times this parent called the function `/' the total number of times the function was called.  Recursive calls to the function are not included in the number after the `/'.

name   This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>' is printed in the `name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly from the child into the function.

children   This is the amount of time that was propagated from the child's children to the function.

called   This is the number of times the function called this child `/' the total number of times the child was called.  Recursive calls by the child are not listed in the number after the `/'.

name   This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The `+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows, for that member, how many times it was called from other members of the cycle.

```
Index by function name

   [1] acessaVetor               [8] desativaMemLog            [6] inicializaVetorAleatorio
   [7] clkDifMemLog              [4] destroiVetor          [5] inicializaVetorNulo
   [2] criaVetor           [9] finalizaMemLog     [11] parse_args
   [3] defineFaseMemLog       [10] iniciaMemLog              [12] produtoInternoVetores
```

5.2.1.4. 400

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00   0.00     0.00     3      0.00     0.00  acessaVetor
  0.00   0.00     0.00     3      0.00     0.00  criaVetor
  0.00   0.00     0.00     3      0.00     0.00  defineFaseMemLog
  0.00   0.00     0.00     3      0.00     0.00  destroiVetor
  0.00   0.00     0.00     3      0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00     2      0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00     1      0.00     0.00  clkDifMemLog
  0.00   0.00     0.00     1      0.00     0.00  desativaMemLog
  0.00   0.00     0.00     1      0.00     0.00  finalizaMemLog
  0.00   0.00     0.00     1      0.00     0.00  iniciaMemLog
  0.00   0.00     0.00     1      0.00     0.00  parse_args
  0.00   0.00     0.00     1      0.00     0.00  produtoInternoVetores

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.
 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name                 the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.
```

```
                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]      0.0    0.00    0.00       3         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[2]      0.0    0.00    0.00       3         criaVetor [2]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[3]      0.0    0.00    0.00       3         defineFaseMemLog [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]      0.0    0.00    0.00       3         destroiVetor [4]
-----------------------------------------------
                0.00    0.00       1/3           main [25]
                0.00    0.00       2/3           inicializaVetorAleatorio [6]
[5]      0.0    0.00    0.00       3         inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]      0.0    0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/3           inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[7]      0.0    0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]      0.0    0.00    0.00       1         desativaMemLog [8]
-----------------------------------------------
```

```
                0.00    0.00    1/1           main [25]
[9]     0.0     0.00    0.00    1         finalizaMemLog [9]
                0.00    0.00    1/1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[10]    0.0     0.00    0.00    1         iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[11]    0.0     0.00    0.00    1         parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[12]    0.0     0.00    0.00    1         produtoInternoVetores [12]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
              Index numbers are sorted numerically.
              The index number is printed next to every function name so
              it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
              in this function and its children.  Note that due to
              different viewpoints, functions excluded by options, etc,
              these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
              function by its children.

     called    This is the number of times the function was called.
              If the function called itself recursively, the number
              only includes non-recursive calls, and is followed by
              a `+' and the number of recursive calls.

     name    The name of the current function.  The index number is
              printed after it.  If the function is a member of a
              cycle, the cycle number is printed between the
              function's name and the index number.
```

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

children   This is the amount of time that was propagated from the
 child's children to the function.

called   This is the number of times the function called
 this child `/' the total number of times the child
 was called. Recursive calls by the child are not
 listed in the number after the `/'.

name   This is the name of the child. The child's index
 number is printed after it. If the child is a
 member of a cycle, the cycle number is printed
 between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.

```
Index by function name

   [1] acessaVetor          [8] desativaMemLog         [6] inicializaVetorAleatorio
   [7] clkDifMemLog         [4] destroiVetor      [5] inicializaVetorNulo
   [2] criaVetor       [9] finalizaMemLog    [11] parse_args
   [3] defineFaseMemLog     [10] iniciaMemLog          [12] produtoInternoVetores
```

5.2.1.5. 500

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total           
 time   seconds   seconds    calls  Ts/call  Ts/call  name    
  0.00      0.00     0.00        3     0.00     0.00  acessaVetor
  0.00      0.00     0.00        3     0.00     0.00  criaVetor
  0.00      0.00     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.00     0.00        3     0.00     0.00  destroiVetor
  0.00      0.00     0.00        3     0.00     0.00  inicializaVetorNulo
  0.00      0.00     0.00        2     0.00     0.00  inicializaVetorAleatorio
  0.00      0.00     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.00     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.00     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.00     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.00     0.00        1     0.00     0.00  parse_args
  0.00      0.00     0.00        1     0.00     0.00  produtoInternoVetores

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
```

```
          this function is profiled, else blank.

 self     the average number of milliseconds spent in this
ms/call   function per call, if this function is profiled,
          else blank.

 total    the average number of milliseconds spent in this
ms/call   function and its descendents per call, if this
          function is profiled, else blank.

name      the name of the function.  This is the minor sort
          for this listing. The index shows the location of
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.
```

```
                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]      0.0    0.00    0.00       3         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[2]      0.0    0.00    0.00       3         criaVetor [2]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[3]      0.0    0.00    0.00       3         defineFaseMemLog [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]      0.0    0.00    0.00       3         destroiVetor [4]
-----------------------------------------------
                0.00    0.00       1/3           main [25]
                0.00    0.00       2/3           inicializaVetorAleatorio [6]
[5]      0.0    0.00    0.00       3         inicializaVetorNulo [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]      0.0    0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/3           inicializaVetorNulo [5]
-----------------------------------------------
```

```
                0.00    0.00    1/1             finalizaMemLog [9]
[7]     0.0     0.00    0.00    1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[8]     0.0     0.00    0.00    1           desativaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[9]     0.0     0.00    0.00    1           finalizaMemLog [9]
                0.00    0.00    1/1             clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[10]    0.0     0.00    0.00    1           iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1           parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1           produtoInternoVetores [12]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index	A unique number given to each element of the table.
		Index numbers are sorted numerically.
		The index number is printed next to every function name so
		it is easier to look up where the function is in the table.

     % time	This is the percentage of the `total' time that was spent
		in this function and its children.  Note that due to
		different viewpoints, functions excluded by options, etc,
		these numbers will NOT add up to 100%.

     self	This is the total amount of time spent in this function.

     children	This is the total amount of time propagated into this
		function by its children.

     called	This is the number of times the function was called.
		If the function called itself recursively, the number
		only includes non-recursive calls, and is followed by
		a `+' and the number of recursive calls.
```

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

children   This is the amount of time that was propagated from the
 child's children to the function.

called   This is the number of times the function called
 this child `/' the total number of times the child
 was called. Recursive calls by the child are not
 listed in the number after the `/'.

name   This is the name of the child. The child's index
 number is printed after it. If the child is a
 member of a cycle, the cycle number is printed
 between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole. This entry shows who called the

```
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [1] acessaVetor              [8] desativaMemLog            [6] inicializaVetorAleatorio
   [7] clkDifMemLog             [4] destroiVetor          [5] inicializaVetorNulo
   [2] criaVetor           [9] finalizaMemLog     [11] parse_args
   [3] defineFaseMemLog      [10] iniciaMemLog               [12] produtoInternoVetores
```

5.2.2. Norma

5.2.2.1. 100

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00   0.00     0.00     3        0.00     0.00  defineFaseMemLog
  0.00   0.00     0.00     1        0.00     0.00  acessaVetor
  0.00   0.00     0.00     1        0.00     0.00  clkDifMemLog
  0.00   0.00     0.00     1        0.00     0.00  criaVetor
  0.00   0.00     0.00     1        0.00     0.00  desativaMemLog
  0.00   0.00     0.00     1        0.00     0.00  destroiVetor
  0.00   0.00     0.00     1        0.00     0.00  finalizaMemLog
  0.00   0.00     0.00     1        0.00     0.00  iniciaMemLog
  0.00   0.00     0.00     1        0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00     1        0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00     1        0.00     0.00  normaVetor
  0.00   0.00     0.00     1        0.00     0.00  parse_args

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
```

```
 seconds   for by this function and those listed above it.

 self     the number of seconds accounted for by this
seconds          function alone.  This is the major sort for this
          listing.

calls     the number of times this function was invoked, if
          this function is profiled, else blank.

 self     the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
          else blank.

 total    the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
          function is profiled, else blank.

name             the name of the function.  This is the minor sort
          for this listing. The index shows the location of
          the function in the gprof listing. If the index is
          in parenthesis it shows where it would appear in
          the gprof listing if it were to be printed.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

                    Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called    name
                0.00    0.00      3/3           main [25]
[1]     0.0     0.00    0.00      3         defineFaseMemLog [1]
-----------------------------------------------
                0.00    0.00      1/1           main [25]
[2]     0.0     0.00    0.00      1         acessaVetor [2]
-----------------------------------------------
                0.00    0.00      1/1           finalizaMemLog [7]
[3]     0.0     0.00    0.00      1         clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00      1/1           main [25]
[4]     0.0     0.00    0.00      1         criaVetor [4]
-----------------------------------------------
                0.00    0.00      1/1           main [25]
```

```
[5]     0.0     0.00    0.00    1       desativaMemLog [5]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[6]     0.0     0.00    0.00    1       destroiVetor [6]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[7]     0.0     0.00    0.00    1       finalizaMemLog [7]
                0.00    0.00    1/1             clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[8]     0.0     0.00    0.00    1       iniciaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[9]     0.0     0.00    0.00    1       inicializaVetorAleatorio [9]
                0.00    0.00    1/1             inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00    1/1             inicializaVetorAleatorio [9]
[10]    0.0     0.00    0.00    1       inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1       normaVetor [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1       parse_args [12]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index    A unique number given to each element of the table.
      Index numbers are sorted numerically.
      The index number is printed next to every function name so
      it is easier to look up where the function is in the table.

     % time    This is the percentage of the `total' time that was spent
      in this function and its children.  Note that due to
      different viewpoints, functions excluded by options, etc,
      these numbers will NOT add up to 100%.

     self    This is the total amount of time spent in this function.

     children    This is the total amount of time propagated into this
      function by its children.
```

called   This is the number of times the function was called.
 If the function called itself recursively, the number
 only includes non-recursive calls, and is followed by
 a `+' and the number of recursive calls.

name   The name of the current function. The index number is
 printed after it. If the function is a member of a
 cycle, the cycle number is printed between the
 function's name and the index number.

For the function's parents, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the function into this parent.

children   This is the amount of time that was propagated from
 the function's children into this parent.

called   This is the number of times this parent called the
 function `/' the total number of times the function
 was called. Recursive calls to the function are not
 included in the number after the `/'.

name   This is the name of the parent. The parent's index
 number is printed after it. If the parent is a
 member of a cycle, the cycle number is printed between
 the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

self   This is the amount of time that was propagated directly
 from the child into the function.

children   This is the amount of time that was propagated from the
 child's children to the function.

called   This is the number of times the function called
 this child `/' the total number of times the child
 was called. Recursive calls by the child are not
 listed in the number after the `/'.

name   This is the name of the child. The child's index

```
            number is printed after it.  If the child is a
            member of a cycle, the cycle number is printed
            between the name and the index number.

 If there are any cycles (circles) in the call graph, there is an
 entry for the cycle-as-a-whole.  This entry shows who called the
 cycle (as parents) and the members of the cycle (as children.)
 The `+' recursive calls entry shows the number of function calls that
 were internal to the cycle, and the calls entry for each member shows,
 for that member, how many times it was called from other members of
 the cycle.

Copyright (C) 2012-2020 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

Index by function name

   [2] acessaVetor              [5] desativaMemLog          [9] inicializaVetorAleatorio
   [3] clkDifMemLog             [6] destroiVetor           [10] inicializaVetorNulo
   [4] criaVetor                [7] finalizaMemLog         [11] normaVetor
   [1] defineFaseMemLog         [8] iniciaMemLog           [12] parse_args
```

5.2.2.2. 200

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00        3     0.00     0.00  defineFaseMemLog
  0.00      0.00     0.00        1     0.00     0.00  acessaVetor
  0.00      0.00     0.00        1     0.00     0.00  clkDifMemLog
  0.00      0.00     0.00        1     0.00     0.00  criaVetor
  0.00      0.00     0.00        1     0.00     0.00  desativaMemLog
  0.00      0.00     0.00        1     0.00     0.00  destroiVetor
  0.00      0.00     0.00        1     0.00     0.00  finalizaMemLog
  0.00      0.00     0.00        1     0.00     0.00  iniciaMemLog
  0.00      0.00     0.00        1     0.00     0.00  inicializaVetorAleatorio
  0.00      0.00     0.00        1     0.00     0.00  inicializaVetorNulo
  0.00      0.00     0.00        1     0.00     0.00  normaVetor
  0.00      0.00     0.00        1     0.00     0.00  parse_args

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]      0.0    0.00    0.00       3         defineFaseMemLog [1]
```

```
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[2]     0.0     0.00    0.00    1           acessaVetor [2]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [7]
[3]     0.0     0.00    0.00    1           clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[4]     0.0     0.00    0.00    1           criaVetor [4]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[5]     0.0     0.00    0.00    1           desativaMemLog [5]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[6]     0.0     0.00    0.00    1           destroiVetor [6]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[7]     0.0     0.00    0.00    1           finalizaMemLog [7]
                0.00    0.00    1/1             clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[8]     0.0     0.00    0.00    1           iniciaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[9]     0.0     0.00    0.00    1           inicializaVetorAleatorio [9]
                0.00    0.00    1/1             inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00    1/1             inicializaVetorAleatorio [9]
[10]    0.0     0.00    0.00    1           inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1           normaVetor [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1           parse_args [12]
-----------------------------------------------
```

5.2.2.3. 300

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00     0.00     0.00        3     0.00     0.00  defineFaseMemLog
  0.00     0.00     0.00        1     0.00     0.00  acessaVetor
  0.00     0.00     0.00        1     0.00     0.00  clkDifMemLog
  0.00     0.00     0.00        1     0.00     0.00  criaVetor
  0.00     0.00     0.00        1     0.00     0.00  desativaMemLog
  0.00   0.00     0.00     1        0.00     0.00  destroiVetor
  0.00   0.00     0.00     1        0.00     0.00  finalizaMemLog
  0.00   0.00     0.00     1        0.00     0.00  iniciaMemLog
  0.00   0.00     0.00     1        0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00     1        0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00     1        0.00     0.00  normaVetor
  0.00   0.00     0.00     1        0.00     0.00  parse_args

index % time    self  children    called     name
                0.00    0.00       3/3           main [25]
[1]      0.0    0.00    0.00       3         defineFaseMemLog [1]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[2]      0.0    0.00    0.00       1         acessaVetor [2]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [7]
[3]      0.0    0.00    0.00       1         clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[4]      0.0    0.00    0.00       1         criaVetor [4]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[5]      0.0    0.00    0.00       1         desativaMemLog [5]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[6]      0.0    0.00    0.00       1         destroiVetor [6]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[7]      0.0    0.00    0.00       1         finalizaMemLog [7]
                0.00    0.00       1/1           clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]      0.0    0.00    0.00       1         iniciaMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[9]      0.0    0.00    0.00       1         inicializaVetorAleatorio [9]
                0.00    0.00       1/1           inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00       1/1           inicializaVetorAleatorio [9]
[10]     0.0    0.00    0.00       1         inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[11]     0.0    0.00    0.00       1         normaVetor [11]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[12]     0.0    0.00    0.00       1         parse_args [12]
-----------------------------------------------
```

5.2.2.4. 400

```
 %   cumulative  self         self    total
time   seconds   seconds      calls  Ts/call  Ts/call  name
 0.00  0.00     0.00    3       0.00    0.00  defineFaseMemLog
 0.00  0.00     0.00    1       0.00    0.00  acessaVetor
 0.00  0.00     0.00    1       0.00    0.00  clkDifMemLog
 0.00  0.00     0.00    1       0.00    0.00  criaVetor
 0.00  0.00     0.00    1       0.00    0.00  desativaMemLog
 0.00  0.00     0.00    1       0.00    0.00  destroiVetor
 0.00  0.00     0.00    1       0.00    0.00  finalizaMemLog
 0.00  0.00     0.00    1       0.00    0.00  iniciaMemLog
 0.00  0.00     0.00    1       0.00    0.00  inicializaVetorAleatorio
 0.00  0.00     0.00    1       0.00    0.00  inicializaVetorNulo
 0.00  0.00     0.00    1       0.00    0.00  normaVetor
 0.00  0.00     0.00    1       0.00    0.00  parse_args


index % time    self  children    called   name
                0.00    0.00      3/3          main [25]
[1]      0.0    0.00    0.00      3        defineFaseMemLog [1]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[2]      0.0    0.00    0.00      1        acessaVetor [2]
-----------------------------------------------
                0.00    0.00      1/1          finalizaMemLog [7]
[3]      0.0    0.00    0.00      1        clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[4]      0.0    0.00    0.00      1        criaVetor [4]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[5]      0.0    0.00    0.00      1        desativaMemLog [5]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[6]      0.0    0.00    0.00      1        destroiVetor [6]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[7]      0.0    0.00    0.00      1        finalizaMemLog [7]
                0.00    0.00      1/1          clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[8]      0.0    0.00    0.00      1        iniciaMemLog [8]
-----------------------------------------------
                0.00    0.00      1/1          main [25]
[9]      0.0    0.00    0.00      1        inicializaVetorAleatorio [9]
                0.00    0.00      1/1          inicializaVetorNulo [10]
-----------------------------------------------
```

```
                  0.00    0.00    1/1          inicializaVetorAleatorio [9]
[10]     0.0      0.00    0.00    1        inicializaVetorNulo [10]
-----------------------------------------------
                  0.00    0.00    1/1          main [25]
[11]     0.0      0.00    0.00    1        normaVetor [11]
-----------------------------------------------
                  0.00    0.00    1/1          main [25]
[12]     0.0      0.00    0.00    1        parse_args [12]
-----------------------------------------------
```

5.2.2.5. 500

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00    0.00     0.00        3     0.00     0.00  defineFaseMemLog
  0.00    0.00     0.00        1     0.00     0.00  acessaVetor
  0.00    0.00     0.00        1     0.00     0.00  clkDifMemLog
  0.00    0.00     0.00        1     0.00     0.00  criaVetor
  0.00    0.00     0.00        1     0.00     0.00  desativaMemLog
  0.00    0.00     0.00        1     0.00     0.00  destroiVetor
  0.00    0.00     0.00        1     0.00     0.00  finalizaMemLog
  0.00    0.00     0.00        1     0.00     0.00  iniciaMemLog
  0.00    0.00     0.00        1     0.00     0.00  inicializaVetorAleatorio
  0.00    0.00     0.00        1     0.00     0.00  inicializaVetorNulo
  0.00    0.00     0.00        1     0.00     0.00  normaVetor
  0.00    0.00     0.00        1     0.00     0.00  parse_args


index % time    self  children    called     name
                  0.00    0.00    3/3          main [25]
[1]      0.0      0.00    0.00    3        defineFaseMemLog [1]
-----------------------------------------------
                  0.00    0.00    1/1          main [25]
[2]      0.0      0.00    0.00    1        acessaVetor [2]
-----------------------------------------------
                  0.00    0.00    1/1          finalizaMemLog [7]
[3]      0.0      0.00    0.00    1        clkDifMemLog [3]
-----------------------------------------------
                  0.00    0.00    1/1          main [25]
[4]      0.0      0.00    0.00    1        criaVetor [4]
-----------------------------------------------
                  0.00    0.00    1/1          main [25]
[5]      0.0      0.00    0.00    1        desativaMemLog [5]
-----------------------------------------------
                  0.00    0.00    1/1          main [25]
[6]      0.0      0.00    0.00    1        destroiVetor [6]
```

```
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[7]     0.0     0.00    0.00    1         finalizaMemLog [7]
                0.00    0.00    1/1           clkDifMemLog [3]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[8]     0.0     0.00    0.00    1         iniciaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[9]     0.0     0.00    0.00    1         inicializaVetorAleatorio [9]
                0.00    0.00    1/1           inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00    1/1           inicializaVetorAleatorio [9]
[10]    0.0     0.00    0.00    1         inicializaVetorNulo [10]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[11]    0.0     0.00    0.00    1         normaVetor [11]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[12]    0.0     0.00    0.00    1         parse_args [12]
-----------------------------------------------
```

5.2.3. Soma

5.2.3.1. 100

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00    0.00     0.00        4     0.00     0.00  acessaVetor
  0.00    0.00     0.00        4     0.00     0.00  criaVetor
  0.00    0.00     0.00        4     0.00     0.00  inicializaVetorNulo
  0.00    0.00     0.00        3     0.00     0.00  defineFaseMemLog
  0.00    0.00     0.00        3     0.00     0.00  destroiVetor
  0.00    0.00     0.00        2     0.00     0.00  inicializaVetorAleatorio
  0.00    0.00     0.00        1     0.00     0.00  clkDifMemLog
  0.00    0.00     0.00        1     0.00     0.00  desativaMemLog
  0.00    0.00     0.00        1     0.00     0.00  finalizaMemLog
  0.00    0.00     0.00        1     0.00     0.00  iniciaMemLog
  0.00    0.00     0.00        1     0.00     0.00  parse_args
  0.00    0.00     0.00        1     0.00     0.00  somaVetores

index % time    self  children    called     name
                0.00    0.00    4/4           main [25]
[1]     0.0     0.00    0.00    4         acessaVetor [1]
-----------------------------------------------
                0.00    0.00    1/4           somaVetores [12]
                0.00    0.00    3/4           main [25]
[2]     0.0     0.00    0.00    4         criaVetor [2]
-----------------------------------------------
```

```
                0.00    0.00    1/4             main [25]
                0.00    0.00    1/4             somaVetores [12]
                0.00    0.00    2/4             inicializaVetorAleatorio [6]
[3]     0.0     0.00    0.00    4           inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00    3/3             main [25]
[4]     0.0     0.00    0.00    3           defineFaseMemLog [4]
-----------------------------------------------
                0.00    0.00    3/3             main [25]
[5]     0.0     0.00    0.00    3           destroiVetor [5]
-----------------------------------------------
                0.00    0.00    2/2             main [25]
[6]     0.0     0.00    0.00    2           inicializaVetorAleatorio [6]
                0.00    0.00    2/4             inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [9]
[7]     0.0     0.00    0.00    1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[8]     0.0     0.00    0.00    1           desativaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[9]     0.0     0.00    0.00    1           finalizaMemLog [9]
                0.00    0.00    1/1             clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[10]    0.0     0.00    0.00    1           iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1           parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1           somaVetores [12]
                0.00    0.00    1/4             criaVetor [2]
                0.00    0.00    1/4             inicializaVetorNulo [3]
-----------------------------------------------
```

5.2.3.2. 200

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00    0.00     0.00       4     0.00     0.00  acessaVetor
  0.00    0.00     0.00       4     0.00     0.00  criaVetor
  0.00    0.00     0.00       4     0.00     0.00  inicializaVetorNulo
  0.00   0.00    0.00    3        0.00    0.00  defineFaseMemLog
  0.00   0.00    0.00    3        0.00    0.00  destroiVetor
  0.00   0.00    0.00    2        0.00    0.00  inicializaVetorAleatorio
  0.00   0.00    0.00    1        0.00    0.00  clkDifMemLog
  0.00   0.00    0.00    1        0.00    0.00  desativaMemLog
  0.00   0.00    0.00    1        0.00    0.00  finalizaMemLog
  0.00   0.00    0.00    1        0.00    0.00  iniciaMemLog
  0.00   0.00    0.00    1        0.00    0.00  parse_args
  0.00   0.00    0.00    1        0.00    0.00  somaVetores

index % time    self  children    called     name
                0.00    0.00       4/4           main [25]
[1]      0.0    0.00    0.00       4         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       1/4           somaVetores [12]
                0.00    0.00       3/4           main [25]
[2]      0.0    0.00    0.00       4         criaVetor [2]
-----------------------------------------------
                0.00    0.00       1/4           main [25]
                0.00    0.00       1/4           somaVetores [12]
                0.00    0.00       2/4           inicializaVetorAleatorio [6]
[3]      0.0    0.00    0.00       4         inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]      0.0    0.00    0.00       3         defineFaseMemLog [4]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[5]      0.0    0.00    0.00       3         destroiVetor [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]      0.0    0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/4           inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[7]      0.0    0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]      0.0    0.00    0.00       1         desativaMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[9]      0.0    0.00    0.00       1         finalizaMemLog [9]
                0.00    0.00       1/1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[10]     0.0    0.00    0.00       1         iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[11]     0.0    0.00    0.00       1         parse_args [11]
```

```
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1           somaVetores [12]
                0.00    0.00    1/4             criaVetor [2]
                0.00    0.00    1/4             inicializaVetorNulo [3]
-----------------------------------------------
```

5.2.3.3. 300

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00   0.00     0.00    4      0.00     0.00  acessaVetor
  0.00   0.00     0.00    4      0.00     0.00  criaVetor
  0.00   0.00     0.00    4      0.00     0.00  inicializaVetorNulo
  0.00   0.00     0.00    3      0.00     0.00  defineFaseMemLog
  0.00   0.00     0.00    3      0.00     0.00  destroiVetor
  0.00   0.00     0.00    2      0.00     0.00  inicializaVetorAleatorio
  0.00   0.00     0.00    1      0.00     0.00  clkDifMemLog
  0.00   0.00     0.00    1      0.00     0.00  desativaMemLog
  0.00   0.00     0.00    1      0.00     0.00  finalizaMemLog
  0.00   0.00     0.00    1      0.00     0.00  iniciaMemLog
  0.00   0.00     0.00    1      0.00     0.00  parse_args
  0.00   0.00     0.00    1      0.00     0.00  somaVetores

index % time    self  children    called     name
                0.00    0.00    4/4             main [25]
[1]     0.0     0.00    0.00    4           acessaVetor [1]
-----------------------------------------------
                0.00    0.00    1/4             somaVetores [12]
                0.00    0.00    3/4             main [25]
[2]     0.0     0.00    0.00    4           criaVetor [2]
-----------------------------------------------
                0.00    0.00    1/4             main [25]
                0.00    0.00    1/4             somaVetores [12]
                0.00    0.00    2/4             inicializaVetorAleatorio [6]
[3]     0.0     0.00    0.00    4           inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00    3/3             main [25]
[4]     0.0     0.00    0.00    3           defineFaseMemLog [4]
-----------------------------------------------
                0.00    0.00    3/3             main [25]
[5]     0.0     0.00    0.00    3           destroiVetor [5]
-----------------------------------------------
```

```
                0.00    0.00    2/2             main [25]
[6]     0.0     0.00    0.00    2       inicializaVetorAleatorio [6]
                0.00    0.00    2/4             inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [9]
[7]     0.0     0.00    0.00    1       clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[8]     0.0     0.00    0.00    1       desativaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[9]     0.0     0.00    0.00    1       finalizaMemLog [9]
                0.00    0.00    1/1             clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[10]    0.0     0.00    0.00    1       iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1       parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1       somaVetores [12]
                0.00    0.00    1/4             criaVetor [2]
                0.00    0.00    1/4             inicializaVetorNulo [3]
-----------------------------------------------
```

5.2.3.4. 400

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00    0.00     0.00        4     0.00     0.00  acessaVetor
  0.00    0.00     0.00        4     0.00     0.00  criaVetor
  0.00    0.00     0.00        4     0.00     0.00  inicializaVetorNulo
  0.00    0.00     0.00        3     0.00     0.00  defineFaseMemLog
  0.00    0.00     0.00        3     0.00     0.00  destroiVetor
  0.00    0.00     0.00        2     0.00     0.00  inicializaVetorAleatorio
  0.00    0.00     0.00        1     0.00     0.00  clkDifMemLog
  0.00    0.00     0.00        1     0.00     0.00  desativaMemLog
  0.00    0.00     0.00        1     0.00     0.00  finalizaMemLog
  0.00    0.00     0.00        1     0.00     0.00  iniciaMemLog
  0.00    0.00     0.00        1     0.00     0.00  parse_args
  0.00    0.00     0.00        1     0.00     0.00  somaVetores

index % time    self  children    called     name
                0.00    0.00    4/4             main [25]
[1]     0.0     0.00    0.00    4       acessaVetor [1]
```

```
-----------------------------------------------
                0.00    0.00    1/4             somaVetores [12]
                0.00    0.00    3/4             main [25]
[2]     0.0     0.00    0.00    4          criaVetor [2]
-----------------------------------------------
                0.00    0.00    1/4             main [25]
                0.00    0.00    1/4             somaVetores [12]
                0.00    0.00    2/4             inicializaVetorAleatorio [6]
[3]     0.0     0.00    0.00    4          inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00    3/3             main [25]
[4]     0.0     0.00    0.00    3          defineFaseMemLog [4]
-----------------------------------------------
                0.00    0.00    3/3             main [25]
[5]     0.0     0.00    0.00    3          destroiVetor [5]
-----------------------------------------------
                0.00    0.00    2/2             main [25]
[6]     0.0     0.00    0.00    2          inicializaVetorAleatorio [6]
                0.00    0.00    2/4             inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00    1/1             finalizaMemLog [9]
[7]     0.0     0.00    0.00    1          clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[8]     0.0     0.00    0.00    1          desativaMemLog [8]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[9]     0.0     0.00    0.00    1          finalizaMemLog [9]
                0.00    0.00    1/1             clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[10]    0.0     0.00    0.00    1          iniciaMemLog [10]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[11]    0.0     0.00    0.00    1          parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1             main [25]
[12]    0.0     0.00    0.00    1          somaVetores [12]
                0.00    0.00    1/4             criaVetor [2]
                0.00    0.00    1/4             inicializaVetorNulo [3]
-----------------------------------------------
```

5.2.3.5. 500

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00        4     0.00     0.00  acessaVetor
  0.00  0.00    0.00    4        0.00    0.00  criaVetor
  0.00  0.00    0.00    4        0.00    0.00  inicializaVetorNulo
  0.00  0.00    0.00    3        0.00    0.00  defineFaseMemLog
  0.00  0.00    0.00    3        0.00    0.00  destroiVetor
  0.00  0.00    0.00    2        0.00    0.00  inicializaVetorAleatorio
  0.00  0.00    0.00    1        0.00    0.00  clkDifMemLog
  0.00  0.00    0.00    1        0.00    0.00  desativaMemLog
  0.00  0.00    0.00    1        0.00    0.00  finalizaMemLog
  0.00  0.00    0.00    1        0.00    0.00  iniciaMemLog
  0.00  0.00    0.00    1        0.00    0.00  parse_args
  0.00  0.00    0.00    1        0.00    0.00  somaVetores

index % time    self  children    called     name
                0.00    0.00       4/4           main [25]
[1]      0.0    0.00    0.00       4         acessaVetor [1]
-----------------------------------------------
                0.00    0.00       1/4           somaVetores [12]
                0.00    0.00       3/4           main [25]
[2]      0.0    0.00    0.00       4         criaVetor [2]
-----------------------------------------------
                0.00    0.00       1/4           main [25]
                0.00    0.00       1/4           somaVetores [12]
                0.00    0.00       2/4           inicializaVetorAleatorio [6]
[3]      0.0    0.00    0.00       4         inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[4]      0.0    0.00    0.00       3         defineFaseMemLog [4]
-----------------------------------------------
                0.00    0.00       3/3           main [25]
[5]      0.0    0.00    0.00       3         destroiVetor [5]
-----------------------------------------------
                0.00    0.00       2/2           main [25]
[6]      0.0    0.00    0.00       2         inicializaVetorAleatorio [6]
                0.00    0.00       2/4           inicializaVetorNulo [3]
-----------------------------------------------
                0.00    0.00       1/1           finalizaMemLog [9]
[7]      0.0    0.00    0.00       1         clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[8]      0.0    0.00    0.00       1         desativaMemLog [8]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[9]      0.0    0.00    0.00       1         finalizaMemLog [9]
                0.00    0.00       1/1           clkDifMemLog [7]
-----------------------------------------------
                0.00    0.00       1/1           main [25]
[10]     0.0    0.00    0.00       1         iniciaMemLog [10]
-----------------------------------------------
```

```
                0.00    0.00    1/1           main [25]
[11]    0.0     0.00    0.00    1         parse_args [11]
-----------------------------------------------
                0.00    0.00    1/1           main [25]
[12]    0.0     0.00    0.00    1         somaVetores [12]
                0.00    0.00    1/4           criaVetor [2]
                0.00    0.00    1/4           inicializaVetorNulo [3]
-----------------------------------------------
```