**Profile output for n value as 10000:**

```
Copyright (C) 2012-2015 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

  [11] _GLOBAL__sub_I__Z4swapPiS_ [4] swap(int*, int*)     [5] frame_dummy
  [12] __static_initialization_and_destruction_0(int, int) [13] print(int*, int)
   [2] part(int*, int, int)    [3] quick(int*, int, int)
it@it-OptiPlex-3050:~/Desktop$ gprof -a caa gmon.out> can.txt
it@it-OptiPlex-3050:~/Desktop$ cat can.txt
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
 55.90     0.10     0.10 50004999    0.00     0.00  swap(int*, int*)
 39.13     0.17     0.07     9999    0.01     0.02  part(int*, int, int)
  5.59     0.18     0.01                            __gmon_start__
  0.00     0.18     0.00        1    0.00     0.00  print(int*, int)
  0.00     0.18     0.00        1    0.00   171.06  quick(int*, int, int)
  0.00     0.18     0.00        1    0.00   171.06  main

  %           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.
```

```
                  Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 5.52% of 0.18 seconds

index % time    self  children    called     name
                                       1          main [1]
                0.00    0.17       1/1          __libc_csu_init [4]
[1]     94.4    0.00    0.17       1+1      main [1]
                0.00    0.17       1/1          quick(int*, int, int) [3]
                0.00    0.00       1/1          print(int*, int) [10]
                                       1          main [1]
----------------------------------------------
                0.07    0.10    9999/9999        quick(int*, int, int) [3]
[2]     94.4    0.07    0.10    9999         part(int*, int, int) [2]
                0.10    0.00 50004999/50004999   swap(int*, int*) [5]
----------------------------------------------
                                   19998          quick(int*, int, int) [3]
                0.00    0.17       1/1          main [1]
[3]     94.4    0.00    0.17   1+19998    quick(int*, int, int) [3]
                0.07    0.10    9999/9999        part(int*, int, int) [2]
                                   19998          quick(int*, int, int) [3]
----------------------------------------------
                                              <spontaneous>
[4]     94.4    0.00    0.17                 __libc_csu_init [4]
                0.00    0.17       1/1          main [1]
----------------------------------------------
                0.10    0.00 50004999/50004999   part(int*, int, int) [2]
[5]     55.6    0.10    0.00 50004999       swap(int*, int*) [5]
----------------------------------------------
                                              <spontaneous>
[6]      5.6    0.01    0.00                 __gmon_start__ [6]
----------------------------------------------
                0.00    0.00       1/1          main [1]
[10]     0.0    0.00    0.00       1       print(int*, int) [10]
----------------------------------------------

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.
```