it@it-OptiPlex-3020: ~/Documents

File   Edit   View   Search   Terminal   Help

```
it@it-OptiPlex-3020:~/Documents$ g++ -Wall -pg quicksort.cpp -o quicksort
it@it-OptiPlex-3020:~/Documents$ ./quicksort
Enter number:
10
60927 12462 75458 67948 23176 58184 67065 59948 92846 69354

Sorted array:
12462 23176 58184 59948 60927 67065 67948 69354 75458 92846
it@it-OptiPlex-3020:~/Documents$ gprof quicksort gmon.out>output
it@it-OptiPlex-3020:~/Documents$ 
```

```
1 Flat profile:
2
3 Each sample counts as 0.01 seconds.
4  no time accumulated
5
6   %   cumulative   self              self    total
7  time   seconds   seconds    calls  Ts/call  Ts/call  name
8  0.00      0.00      0.00       21    0.00     0.00  swapp(int*, int*)
9  0.00      0.00      0.00       11    0.00     0.00  printRandoms(int, int, int)
10  0.00      0.00      0.00        7    0.00     0.00  part(int*, int, int, int&)
11  0.00      0.00      0.00        2    0.00     0.00  printArr(int*, int)
12  0.00      0.00      0.00        1    0.00     0.00  _GLOBAL__sub_I__Z12printRandomsiii
13  0.00      0.00      0.00        1    0.00     0.00  __static_initialization_and_destruction_0(int, int)
14  0.00      0.00      0.00        1    0.00     0.00  quickSort(int*, int, int, int&)
15
16  %          the percentage of the total running time of the
17 time        program used by this function.
18
19 cumulative a running sum of the number of seconds accounted
20  seconds    for by this function and those listed above it.
21
22  self       the number of seconds accounted for by this
23 seconds     function alone.  This is the major sort for this
24             listing.
25
26 calls       the number of times this function was invoked, if
27             this function is profiled, else blank.
28
29  self       the average number of milliseconds spent in this
30 ms/call     function per call, if this function is profiled,
31             else blank.
32
33  total      the average number of milliseconds spent in this
34 ms/call     function and its descendents per call, if this
35             function is profiled, else blank.
36
```

```
                    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       21/21          part(int*, int, int, int&) [10]
[8]     0.0     0.00    0.00       21         swapp(int*, int*) [8]
-----------------------------------------------
                0.00    0.00       11/11          main [6]
[9]     0.0     0.00    0.00       11         printRandoms(int, int, int) [9]
-----------------------------------------------
                0.00    0.00       7/7            quickSort(int*, int, int, int&) [14]
[10]    0.0     0.00    0.00       7          part(int*, int, int, int&) [10]
                0.00    0.00       21/21          swapp(int*, int*) [8]
-----------------------------------------------
                0.00    0.00       2/2            main [6]
[11]    0.0     0.00    0.00       2          printArr(int*, int) [11]
-----------------------------------------------
                0.00    0.00       1/1            __libc_csu_init [20]
[12]    0.0     0.00    0.00       1          _GLOBAL__sub_I__Z12printRandomsiii [12]
                0.00    0.00       1/1              __static_initialization_and_destruction_0(int, int) [13]
-----------------------------------------------
                0.00    0.00       1/1            _GLOBAL__sub_I__Z12printRandomsiii [12]
[13]    0.0     0.00    0.00       1          __static_initialization_and_destruction_0(int, int) [13]
-----------------------------------------------
                                   14             quickSort(int*, int, int, int&) [14]
                0.00    0.00       1/1            main [6]
[14]    0.0     0.00    0.00       1+14       quickSort(int*, int, int, int&) [14]
                0.00    0.00       7/7              part(int*, int, int, int&) [10]
                                   14             quickSort(int*, int, int, int&) [14]
-----------------------------------------------
```

9162 99162 99163 99164 99164 99167 99168 99168 99168 99168 99169 99170 99170 99172 99176 99176 99177 99178 99179 99180 99183 99183 99183 99183 99184 99187 99188 99189 99189 99189 99190 99190 99190 99190 99193 99194 99195 99196 99196 99198 99198 99199 99200 99202 99204 99205 99205 992 06 99206 99207 99207 99207 99208 99208 99208 99212 99212 99213 99213 99215 99216 99216 99217 99217 99217 99217 99218 99219 99219 99219 99221 9 9221 99222 99222 99223 99223 99224 99224 99225 99225 99226 99226 99227 99228 99229 99230 99231 99232 99233 99235 99237 99237 99238 99239 99239 99240 99240 99240 99241 99241 99242 99242 99243 99243 99245 99246 99247 99249 99249 99250 99250 99250 99251 99257 99260 99260 99261 99261 992 63 99265 99266 99266 99267 99269 99270 99272 99273 99273 99273 99275 99275 99276 99279 99281 99281 99282 99283 99284 99284 99286 99288 99288 9 9289 99289 99292 99293 99294 99296 99296 99298 99298 99299 99299 99301 99302 99303 99303 99304 99306 99306 99306 99308 99308 99308 99313 99314 99314 99316 99318 99318 99318 99319 99319 99322 99323 99323 99323 99325 99328 99328 99328 99329 99330 99330 99331 99331 99331 99333 99335 993 36 99336 99336 99337 99338 99338 99338 99341 99341 99342 99345 99346 99346 99347 99348 99348 99348 99349 99349 99350 99350 99350 99351 99352 9 9353 99354 99355 99361 99362 99363 99364 99364 99365 99365 99365 99366 99368 99368 99369 99369 99370 99371 99372 99372 99373 99375 99377 99378 99378 99379 99381 99381 99383 99386 99386 99386 99389 99390 99390 99391 99392 99392 99393 99393 99395 99397 99397 99397 99399 99399 99400 994 01 99402 99403 99403 99405 99407 99407 99408 99408 99409 99409 99409 99409 99412 99413 99415 99416 99419 99419 99422 99422 99423 99426 99427 9 9428 99432 99433 99433 99433 99434 99435 99437 99437 99437 99439 99439 99440 99440 99443 99444 99448 99450 99450 99451 99452 99452 99453 99453 99453 99454 99455 99455 99457 99459 99464 99466 99467 99467 99468 99469 99469 99470 99471 99472 99473 99473 99474 99475 99475 99475 99480 994 81 99481 99481 99482 99484 99485 99486 99487 99487 99487 99488 99494 99494 99495 99499 99500 99500 99501 99501 99503 99504 99505 99506 99508 9 9511 99511 99513 99515 99517 99517 99518 99518 99518 99519 99520 99520 99522 99523 99524 99525 99529 99530 99530 99531 99532 99534 99536 99537 99538 99540 99540 99544 99544 99544 99547 99548 99552 99555 99562 99563 99564 99565 99565 99567 99569 99569 99569 99570 99571 99571 99572 995 72 99572 99573 99573 99575 99576 99577 99579 99579 99579 99580 99584 99585 99586 99587 99589 99589 99591 99591 99592 99592 99592 99593 99594 9 9594 99597 99597 99597 99598 99598 99599 99600 99600 99601 99601 99602 99603 99603 99604 99605 99605 99606 99606 99608 99608 99609 99610 99611 99611 99612 99616 99616 99616 99618 99618 99623 99623 99624 99624 99626 99626 99627 99627 99629 99631 99632 99632 99633 99634 99634 99635 996 36 99636 99636 99637 99637 99639 99640 99640 99642 99643 99643 99644 99644 99645 99647 99647 99648 99648 99648 99649 99650 99651 99651 99651 9 9651 99652 99652 99654 99655 99656 99656 99656 99656 99656 99658 99659 99659 99660 99662 99664 99666 99667 99667 99668 99668 99669 99669 99670 99671 99671 99671 99673 99673 99675 99676 99677 99679 99679 99679 99680 99681 99681 99683 99683 99686 99686 99687 99688 99688 99688 99689 996 89 99689 99690 99691 99692 99693 99693 99695 99696 99696 99696 99696 99697 99699 99700 99700 99701 99701 99702 99702 99703 99705 99705 99707 9 9707 99708 99708 99708 99711 99712 99712 99713 99713 99715 99715 99716 99718 99720 99724 99724 99726 99726 99728 99731 99732 99732 99733 99736 99743 99746 99746 99747 99747 99748 99748 99748 99749 99749 99750 99751 99752 99754 99756 99756 99757 99757 99758 99758 99759 99760 99761 997 61 99762 99764 99768 99769 99769 99770 99770 99770 99774 99774 99775 99776 99777 99779 99782 99782 99783 99784 99786 99787 99787 99788 99789 9 9790 99790 99792 99793 99796 99797 99798 99798 99798 99798 99801 99802 99802 99803 99806 99809 99812 99812 99813 99813 99814 99814 99815 99818 99818 99819 99819 99819 99820 99820 99821 99822 99823 99824 99824 99825 99827 99827 99828 99828 99829 99833 99834 99834 99835 99836 99836 998 37 99838 99839 99840 99840 99841 99842 99843 99846 99847 99847 99852 99854 99856 99856 99857 99859 99859 99861 99862 99862 99862 99862 99866 9 9866 99867 99869 99871 99872 99875 99880 99880 99883 99883 99885 99885 99888 99888 99890 99890 99891 99892 99892 99892 99893 99893 99894 99894 99895 99895 99896 99896 99896 99897 99897 99898 99898 99898 99898 99900 99900 99901 99901 99902 99903 99903 99905 99906 99908 99909 99909 999 09 99910 99912 99912 99913 99916 99916 99921 99921 99922 99923 99924 99926 99928 99928 99929 99931 99933 99933 99935 99936 99939 99940 99943 9 9944 99944 99944 99944 99944 99945 99947 99947 99949 99950 99951 99952 99952 99953 99954 99956 99958 99958 99959 99959 99960 99960 99961 99962 99962 99963 99963 99965 99965 99966 99966 99968 99969 99969 99970 99971 99972 99972 99973 99973 99973 99977 99978 99978 99982 99985 99986 999 88 99989 99991 99993 99993 99994 99995 99996 99997 99999 100000
it@it-OptiPlex-3020:~/Documents$ gprof quicksort gmon.out>output
it@it-OptiPlex-3020:~/Documents$

Documents ▾     Open ▾    output ~/Documents    Save ≡ ⊖ ⊟ ✕

quicksort.cpp ✕       quicksort.cpp ✕       output ✕

output ✕

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
100.23     0.01      0.01    67728     0.00     0.00  part(int*, int, int, int&)
  0.00     0.01      0.00  1049047     0.00     0.00  swapp(int*, int*)
  0.00     0.01      0.00   100001     0.00     0.00  printRandoms(int, int, int)
  0.00     0.01      0.00        2     0.00     0.00  printArr(int*, int)
  0.00     0.01      0.00        1     0.00     0.00  _GLOBAL__sub_I__Z12printRandomsiii
  0.00     0.01      0.00        1     0.00     0.00  __static_initialization_and_destruction_0(int, int)
  0.00     0.01      0.00        1     0.00    10.02  quickSort(int*, int, int, int&)

 %         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
```

Plain Text ▾    Tab Width: 8 ▾      Ln 1, Col 1    ▾    INS

output
~/Documents

Open ▾　⊞　　　　　　　　　　　　　　　　　　Save　≡　⊖ ⊡ ✕

quicksort.cpp　　✕

output　　✕

quicksort.cpp　　　　　　　　　　✕　　　　　　　output　　✕

```
50 granularity: each sample hit covers 2 byte(s) for 99.77% of 0.01 seconds
51
52 index % time    self  children    called     name
53                                              <spontaneous>
54 [1]     100.0    0.00    0.01                 main [1]
55                  0.00    0.01       1/1          quickSort(int*, int, int, int&) [3]
56                  0.00    0.00  100001/100001    printRandoms(int, int, int) [11]
57                  0.00    0.00       2/2          printArr(int*, int) [12]
58 ---------------------------------------------
59                  0.01    0.00   67728/67728     quickSort(int*, int, int, int&) [3]
60 [2]     100.0    0.01    0.00    67728         part(int*, int, int, int&) [2]
61                  0.00    0.00 1049047/1049047   swapp(int*, int*) [10]
62 ---------------------------------------------
63                               135456          quickSort(int*, int, int, int&) [3]
64                  0.00    0.01       1/1          main [1]
65 [3]     100.0    0.00    0.01    1+135456    quickSort(int*, int, int, int&) [3]
66                  0.01    0.00   67728/67728     part(int*, int, int, int&) [2]
67                               135456          quickSort(int*, int, int, int&) [3]
68 ---------------------------------------------
69                  0.00    0.00 1049047/1049047   part(int*, int, int, int&) [2]
70 [10]     0.0    0.00    0.00 1049047         swapp(int*, int*) [10]
71 ---------------------------------------------
72                  0.00    0.00  100001/100001    main [1]
73 [11]     0.0    0.00    0.00  100001         printRandoms(int, int, int) [11]
74 ---------------------------------------------
75                  0.00    0.00       2/2          main [1]
76 [12]     0.0    0.00    0.00       2         printArr(int*, int) [12]
77 ---------------------------------------------
78                  0.00    0.00       1/1          __libc_csu_init [20]
79 [13]     0.0    0.00    0.00       1         _GLOBAL__sub_I__Z12printRandomsiii [13]
80                  0.00    0.00       1/1          __static_initialization_and_destruction_0(int, int) [14]
81 ---------------------------------------------
82                  0.00    0.00       1/1          _GLOBAL__sub_I__Z12printRandomsiii [13]
83 [14]     0.0    0.00    0.00       1         __static_initialization_and_destruction_0(int, int) [14]
84 ---------------------------------------------
85
```

Plain Text ▾　Tab Width: 8 ▾　　　Ln 1, Col 1　　▾　INS