

# Assignment Submission

**Name** - Nirmitt Chattoo

**PRN** - 2020BTEIT00001

**File number 2:** Profile output for any 'n' value

## Flat profile:

```
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    2069      0.00      0.00  swap
0.00      0.00      0.00     89      0.00      0.00  partition
0.00      0.00      0.00      2      0.00      0.00  printArray
0.00      0.00      0.00      1      0.00      0.00  quickSort
```

## Call Graph:

```
Call graph (explanation follows)

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

index % time   self  children   called      name
[1]    0.0     0.00    0.00  2069/2069    partition [2]
         0.0     0.00    0.00    2069        swap [1]
-----
         0.00    0.00    89/89        quickSort [4]
[2]    0.0     0.00    0.00     89        partition [2]
         0.00    0.00  2069/2069    swap [1]
-----
         0.00    0.00    2/2          main [10]
[3]    0.0     0.00    0.00      2        printArray [3]
-----
         0.00    0.00    178          quickSort [4]
[4]    0.0     0.00    0.00    1/1          main [10]
         0.00    0.00    1+178        quickSort [4]
         0.00    0.00    89/89        partition [2]
         178          quickSort [4]
-----
```

## Complete Output Produced:

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	2069	0.00	0.00	swap
0.00	0.00	0.00	89	0.00	0.00	partition
0.00	0.00	0.00	2	0.00	0.00	printArray
0.00	0.00	0.00	1	0.00	0.00	quickSort

%     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	2069/2069	partition [2]
[1]	0.0	0.00	0.00	2069	swap [1]
-----					
		0.00	0.00	89/89	quickSort [4]
[2]	0.0	0.00	0.00	89	partition [2]
		0.00	0.00	2069/2069	swap [1]
-----					
		0.00	0.00	2/2	main [10]
[3]	0.0	0.00	0.00	2	printArray [3]
-----					
			178		quickSort [4]
		0.00	0.00	1/1	main [10]
[4]	0.0	0.00	0.00	1+178	quickSort [4]
		0.00	0.00	89/89	partition [2]
			178		quickSort [4]
-----					

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] partition	[4] quickSort
[3] printArray	[1] swap