

For each function  $f(n)$  and time  $t$  in the following table, determine the largest size  $n$  of a problem that can be solved in time  $t$ , assuming that the algorithm to

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
$\sqrt{n}$							
$n$							
$n \lg n$							
$n^2$							
$n^3$							
$2^n$							
$n!$							

solve the problem takes  $f(n)$  microseconds.

a. I decided to solve this problem programmatically. I wrote a program called `fs.py`. I had to write an approximation function since some of the functions are not obviously invertible. Check the program.