

### 1-1 Comparison of running times

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

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
$\sqrt{n}$	$10^{12}$	$3.6 \cdot 10^{15}$	$1.3 \cdot 10^{19}$	$7.47 \cdot 10^{21}$	$6.91 \cdot 10^{24}$	$9.95 \cdot 10^{26}$	$9.95 \cdot 10^{30}$
$n$	$10^6$	$6 \cdot 10^7$	$3.6 \cdot 10^9$	$8.64 \cdot 10^{10}$	$2.63 \cdot 10^{12}$	$3.16 \cdot 10^{13}$	$3.16 \cdot 10^{15}$
$n \lg n$	62746	$2.8 \cdot 10^6$	$1.33 \cdot 10^8$	$2.76 \cdot 10^9$	$7.29 \cdot 10^{10}$	$7.99 \cdot 10^{11}$	$6.87 \cdot 10^{13}$
$n^2$	1000	7746 (approx)	60000	293939	$1.62 \cdot 10^6$	$5.62 \cdot 10^6$	$5.61 \cdot 10^7$
$n^3$	100	391 (approx)	1532	4420	13803	31601	146679
$2^n$	20 (approx)	26 (approx)	32	36	41	44	51
$n!$	9	11	12	13	15	16	17

There is a code called "problem1" that solve  $n$  for each row, and calculate the values.