Input 10 numbers

Execution time avg. 0.003s

Function Name	Number of Calls	Elapsed Inclusive Time % ▼	Elapsed Exclusive Time %
⊿ python.exe	0	100.00%	0.00%
⊿ profiling (module)	1	100.00%	7.41%
⊿ <frozen importlib._fin...	1	53.88%	1.45%
▷ <frozen importlib._fin...	1	50.44%	0.23%
▷ <frozen importlib._M...	1	0.94%	0.13%
▷ <frozen importlib._M...	1	0.93%	0.48%
▷ <frozen importlib.cb	1	0.07%	0.04%
<frozen importlib._M...	1	0.03%	0.03%
dict.get	1	0.01%	0.01%
▷ io.open	1	19.56%	3.28%
print	2	9.10%	9.10%
▷ re.findall	2	8.50%	0.14%
▷ mathlib.exp	11	0.60%	0.58%
▷ encodings.cp1250.Incre...	3	0.43%	0.26%
round	1	0.36%	0.36%
mathlib.sum	30	0.08%	0.08%
▷ mathlib.root	1	0.04%	0.03%
mathlib.div	2	0.03%	0.03%
mathlib.mul	1	0.01%	0.01%
mathlib.sub	2	0.01%	0.01%

Input 100 numbers

Execution time avg. 0.003s

Function Name	Number of Calls	Elapsed Inclusive Time % ▼	Elapsed Exclusive Time %
⊿ python.exe	0	100.00%	0.00%
⊿ profiling (module)	1	100.00%	12.99%
▷ <frozen impor...	1	37.35%	0.99%
▷ io.open	1	24.54%	3.99%
▷ re.findall	1	11.21%	0.14%
print	2	9.45%	9.45%
▷ mathlib.exp	101	2.48%	2.37%
mathlib.sum	300	0.81%	0.81%
▷ encodings.cp125...	3	0.50%	0.25%
round	1	0.39%	0.39%
mathlib.sub	2	0.19%	0.19%
▷ mathlib.root	1	0.04%	0.04%
mathlib.div	2	0.03%	0.03%
mathlib.mul	1	0.01%	0.01%

Input 1000 numbers

Execution time avg. 0.0045s

Function Name	Number of Calls	Elapsed Inclusive Time % ▼	Elapsed Exclusive Time %
⊿ 🔥python.exe	0	100.00%	0.00%
⊿ 🔥profiling (module)	1	100.00%	32.52%
▷ 🔥<frozen impor...	1	21.68%	0.56%
▷ 🔥io.open	1	13.53%	2.43%
▷ re.findall	1	10.81%	0.07%
▷ mathlib.exp	1,001	9.55%	8.99%
print	2	6.22%	6.22%
mathlib.sum	3,000	4.91%	4.91%
▷ encodings.cp125...	3	0.42%	0.17%
round	1	0.22%	0.22%
mathlib.sub	2	0.10%	0.10%
▷ mathlib.root	1	0.02%	0.02%
mathlib.div	2	0.02%	0.02%
mathlib.mul	1	0.01%	0.01%

Input 10 000 numbers

Execution time avg. 0.025

Function Name	Number of Calls	Elapsed Inclusive Time % ▼	Elapsed Exclusive Time %
⊿ 🔥python.exe	0	100.00%	0.00%
⊿ 🔥profiling (module)	1	100.00%	53.90%
▷ 🔥mathlib.exp	10,001	19.25%	18.17%
mathlib.sum	30,000	9.84%	9.84%
▷ re.findall	1	9.25%	0.03%
▷ <frozen importli...	1	3.75%	0.10%
▷ io.open	1	2.61%	0.43%
print	2	0.97%	0.97%
▷ encodings.cp125...	6	0.33%	0.03%
round	1	0.06%	0.06%
mathlib.sub	2	0.02%	0.02%
mathlib.div	2	0.01%	0.01%
▷ mathlib.root	1	0.01%	0.01%
mathlib.mul	1	0.00%	0.00%

Summary:

Importing libraries and processing input takes most of the time in input numbers under 100. The execution is low so there is no need for optimization in math_lib. Above 1000 input numbers process takes more time in math_lib especially in exponent and summary. Optimizing these 2 functions could save time.