

IVS – 2.projekt

Profiling

xgallo06

xpaska05

xhalas14

1 Spustenie Profileru

Profiler sa nachádza v adresári src. Jeho spustenie je možné pomocou `python3 ./profiler` alebo `python3 ./profiler <data.txt` (presmerovanie štandardného vstupu)

2 Výstup profileru

Program na výpočet smerodajnej odchýlky bol profilovaný s 3 rôznymi testovacími vstupnými hodnotami.

Poznámka: 1.riadok každého výstupu profileru predstavuje výslednú smerodajnú odchýlku pre daný súbor vstupných hodnôt.

1. Vstup č.1

Veľkosť vstupu: 10 číselných hodnôt

Výstup profileru:

```
1060.3213139840625
    36 function calls in 0.004 seconds

Ordered by: internal time
List reduced from 15 to 3 due to restriction <'ivsmath.py'>

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    20    0.000    0.000    0.000    0.000 ivsmath.py:162(power)
     2    0.000    0.000    0.000    0.000 ivsmath.py:114(division)
     1    0.000    0.000    0.000    0.000 ivsmath.py:128(radical)
```

2. Vstup č.2

Veľkosť vstupu: 1 000 číselných hodnôt

Výstup profileru:

```
43455.69105725018
   2016 function calls in 0.002 seconds

Ordered by: internal time
List reduced from 15 to 3 due to restriction <'ivsmath.py'>

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
   2000    0.001    0.000    0.001    0.000 ivsmath.py:162(power)
     2    0.000    0.000    0.000    0.000 ivsmath.py:114(division)
     1    0.000    0.000    0.000    0.000 ivsmath.py:128(radical)
```

3. Vstup č.3

Veľkosť vstupu:

1 000 000 číselných hodnôt

Výstup profileru:

```
865.5006365188475
2000016 function calls in 1.553 seconds

Ordered by: internal time
List reduced from 15 to 3 due to restriction <'ivsmath.py'>

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
2000000  0.365    0.000    0.365    0.000 ivsmath.py:162(power)
2        0.000    0.000    0.000    0.000 ivsmath.py:114(division)
1        0.000    0.000    0.000    0.000 ivsmath.py:128(radical)
```

3 Záver

Na základe našich výstupov profilingu môžeme usúdiť, že funkcia **power()** z matematickej knižnice **ivsmath.py** zaberá najviac času na vykonanie výpočtu, je najčastejšie volaná. Pri vstupoch pozostávajúcich z veľkého počtu číselných hodnôt to môže spôsobiť výraznejšie predĺženie vykonania výpočtu, avšak pri malých vstupoch je toto časové oneskorenie takmer nespozorovateľné.