

# 1. Úvod

Jelikož jsme psali kalkulačku v jazyce C++ pomocí nástroje Qt zvolili jsme jej i pro naprogramování výběrové směrodatné odchylky. Nejedná se o vyloženě nízkoúrovňový jazyk, ale i tak je C++ velmi rychlý a z toho důvodu byl průběh programu velmi rychlý. Pro měření doby trvání jsme použili nástroj gprof z balíku překladače gcc.

## 2. Výsledky

Testovali jsme program na celkově třech řádech čísel. Tes pro  $10^3$  čísel byl úplně zanedbatelný a z toho důvodu ho zde nebudeme ani uvádět. Pro větší čísla už výsledky stojí za zmínku, jak můžete vidět v tabulce. Z tabulek je na první pohled jasné vidět, jak v případě nutnosti zefektivnit kód.

### 2.1. $10^6$

Celkový čas [%]	Celkový čas [s]	Strávený čas na funkci [s]	Název funkce
33,33	0,01	0,01	std::vector::at
33,33	0,02	0,01	std::forward(std::remove_reference)
16,67	0,03	0,01	secti
16,67	0,03	0,01	odecitani

### 2.2. $10^7$

Celkový čas [%]	Celkový čas [s]	Strávený čas na funkci [s]	Název funkce
22,86	0,08	0,08	Math_lib::secti
17,14	0,14	0,06	main
14,29	0,19	0,05	std::vector::pushback
11,43	0,23	0,04	new_allocator::construct
11,43	0,27	0,04	std::vector::operator[]
8,57	0,30	0,03	std::vector::M_range_check
2,86	0,31	0,01	std::vector::size
2,86	0,32	0,01	std::vector::at
2,86	0,33	0,01	_init
1,43	0,34	0,01	Math_lib::na_x
1,43	0,34	0,01	Math_lib::odecitani
1,43	0,34	0,01	Math_lib::odmocnina

## 3. Závěr

Jak můžeme vidět v tabulkách program pro výpočet výběrové směrodatné odchylky má poměrně dost prostoru pro optimalizaci. Avšak většina stráveného času je na funkcích mimo naši knihovnu Math\_lib a většinu optimalizace by pokrylo použití přímo dynamicky alokované paměti namísto vektorů. Tedy přímo tento program, naši knihovnu jako takovou optimalizovat netřeba.