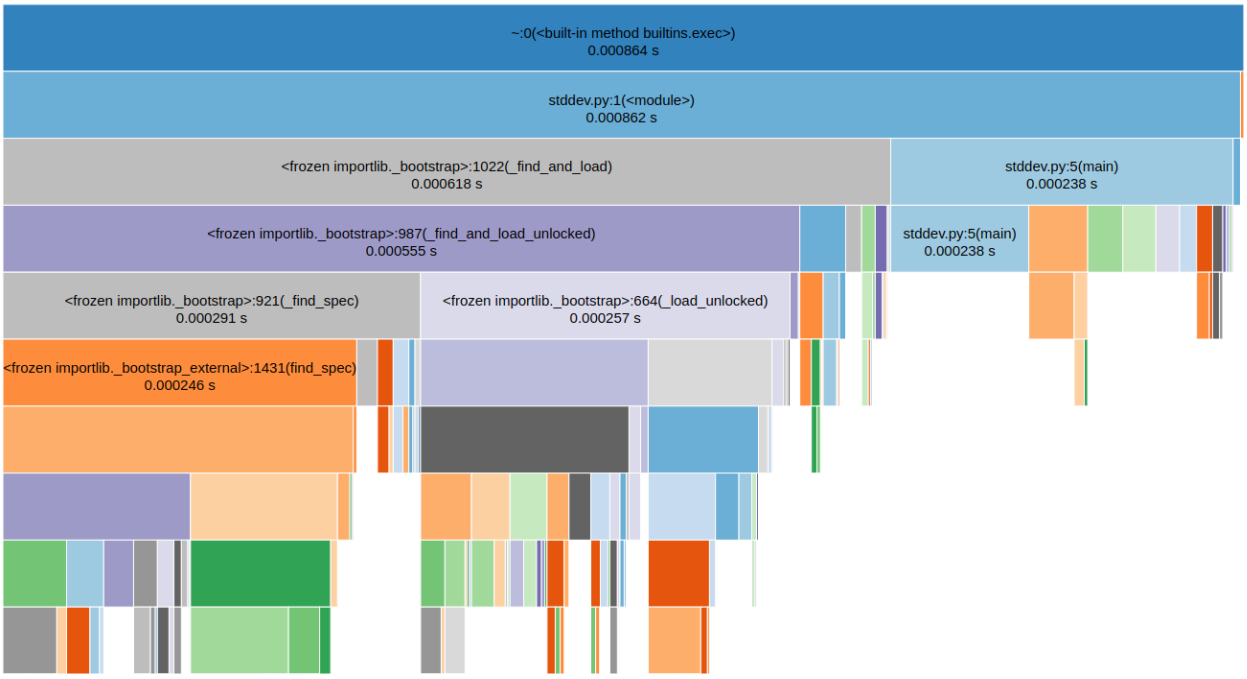
# Profilování - zpráva

Profilování bylo prováděno pomocí modulu cProfile, který je součástí standardní knihovny Python. Program stddev.py byl profilován posloupnostmi čísel 1-10, 1-1000 a 1-1000000. Výstup ze všech běhů profilování je uložen v souboru output.txt.

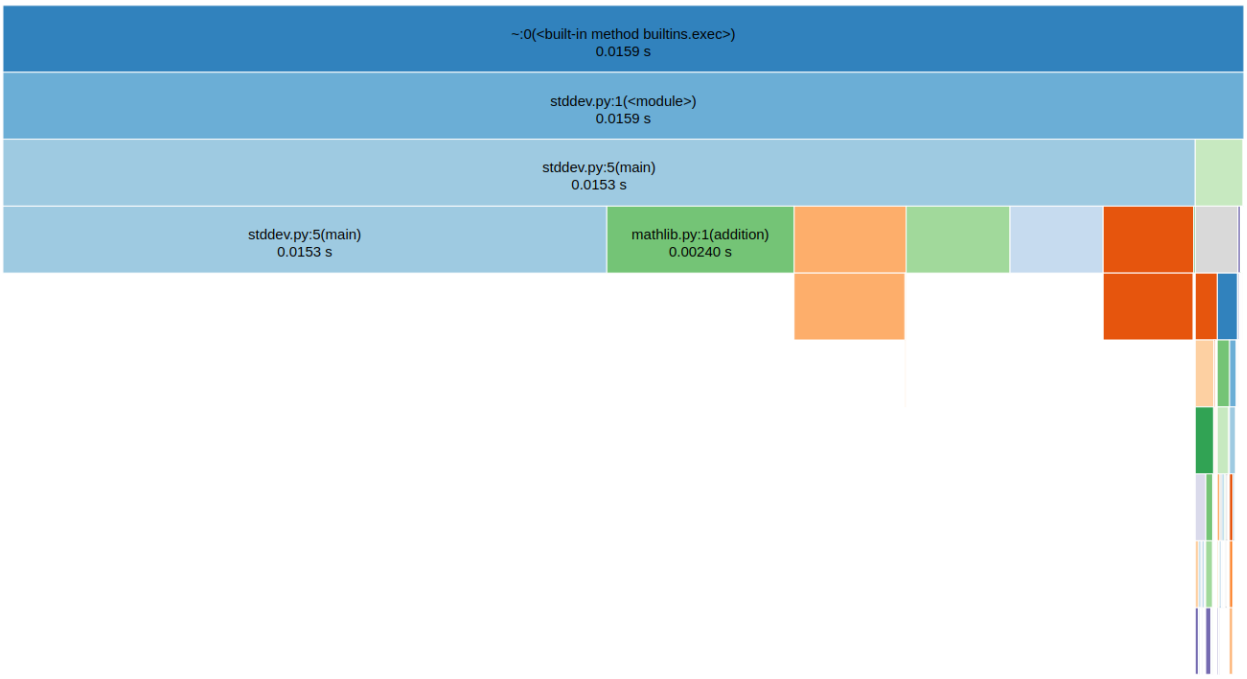
Pro snadnější analýzu je také možné data vizualizovat například pomocí programu snakeviz. Pro vizualizaci je ale nutné vygenerovat binární soubor při profilování s možností -o. Při pouzití snakeviz se pod grafem celého běhu nachází take tabulka statistik běhu, jako při normálním běhu, ale tato tabulka obsahuje přesnější hodnoty, hlavně pro krátké běhy funkcí.

## Vizualizace 1-10

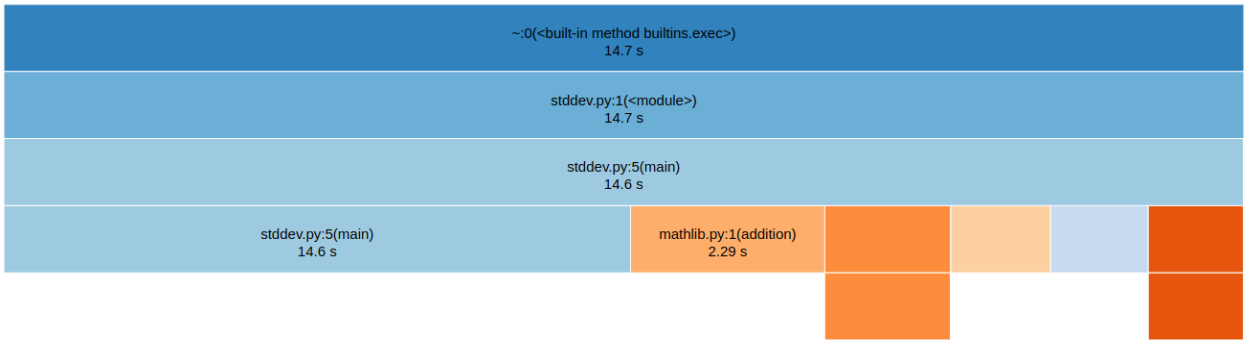



Při pohledu na první obrázek je vidět, že vetšinu běhu programu spotřebuje importlib._bootstrap. Tato část zajišťuje načtení knihoven pro stddev.py, a mezi jednotlivými běhy se téměř neměnila doba po kterou běžela. Z matematické knihovny zabrala největší část programu funkce sčítání, byla volána 20 krát, po ní to byla funkce pro umocnení, která byla volána 11 krát.

## Vizualizace 1-1000




## Vizualizace 1-1000000




Naopak na posledním obrázku, mimo main funkci, zabere nejvíce času funkce sčítání, která je celkem volána  2000000 krát, další z matematické knihovny je mocnina, ta je volána celkem 1000001 krát. Zbylé tři funkce, které jsou na obrázku videt, jsou input, split a extend, pomocí kterých jsou načítánz vstupní hodnoty.

# Závěr

Pro optimalizaci tohoto programu by bylo vhodné se zaměřit na funkci sčítání a umocnení. Pokud máme n vstupů sčítání je voláno 2*n krát a umocnení n+1 krát. Popřípadě změnit způsob, kterým jsou vstupní data čtena ze vstupu. Jelikož tyto operace mají největší podíl na délce běhu programu.