

Profiling - flamegraph

Při spuštění se vstupem o miliónu čísel dostaneme tento výstup:



Tyto data nejsou nijak časově seřazena, jsou pouze seřazena podle hloubky volání ve stacku. Navíc je zde spousta volání okolo, které nás nezajímá.

Když si najdeme část kódu, která volá naši matematickou funkci a rozklikneme si ji, dostaneme následující graf:



Jde vidět, že v `to_postfix` funkci se tráví nejvíce času, a to hlavně kvůli alokacím paměti vektoru při volání funkce `push`. Nejjednodušší zlepšení by bylo zřejmě předem naalokovat aspoň nějakou paměť, nebo pokud by to bylo možné přesně vypočítat kolik paměti bude potřeba. Další funkce, kde se tráví hodně času je `process_current_number`, a je to opět kvůli alokacím vektoru, tudíž stejná optimalizace je potřeba.

Při volání profilingu pro tisíc čísel, i přes zvětšené vzorkování některé funkce nejsou vůbec v grafu vidět:

