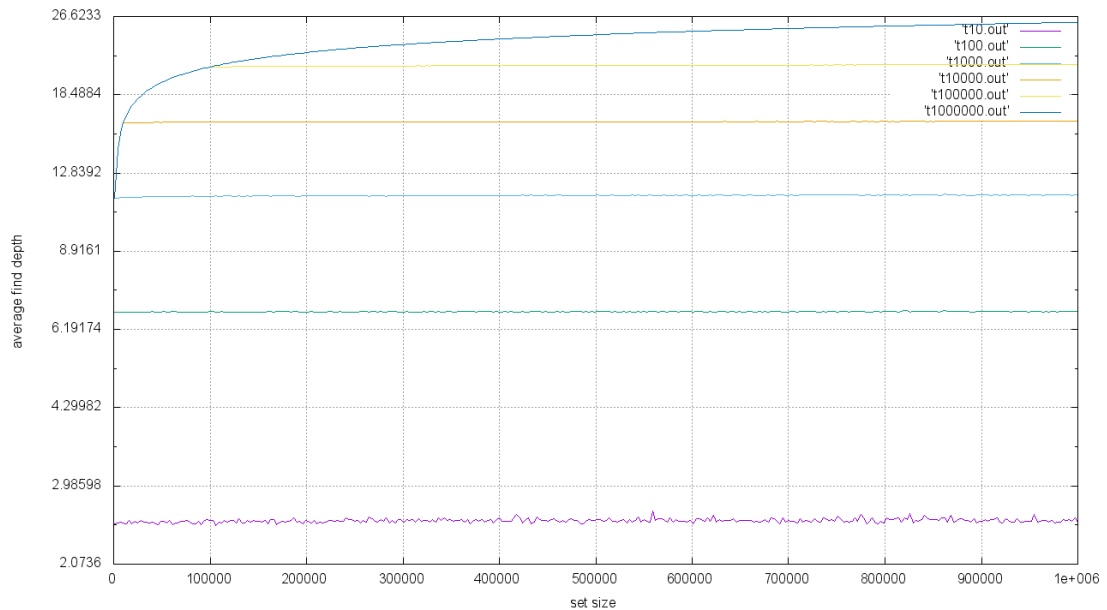
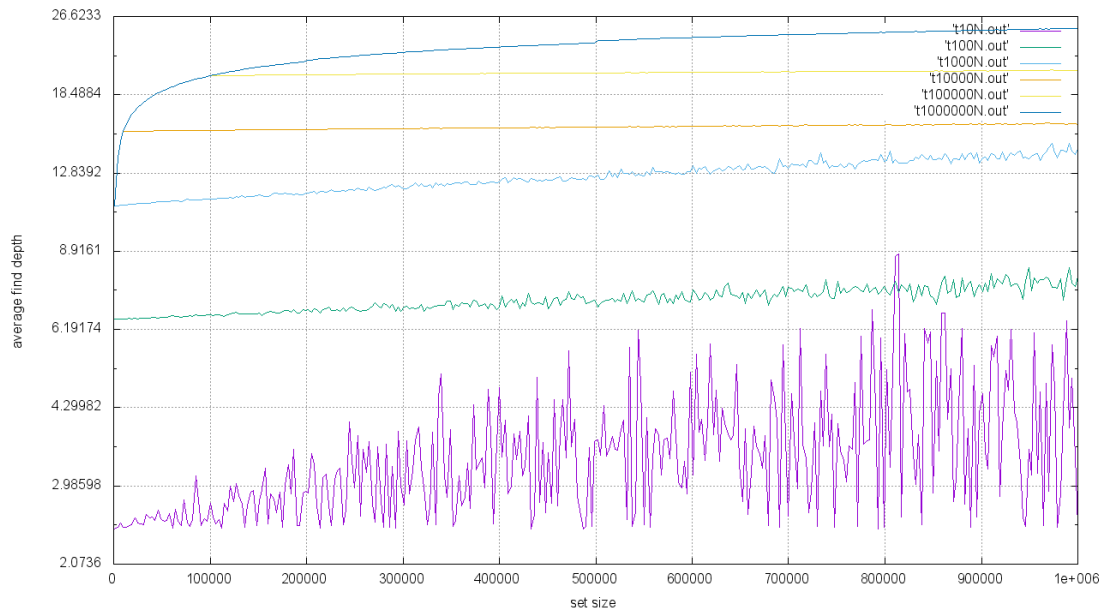


## 0.1 Standartní implementace



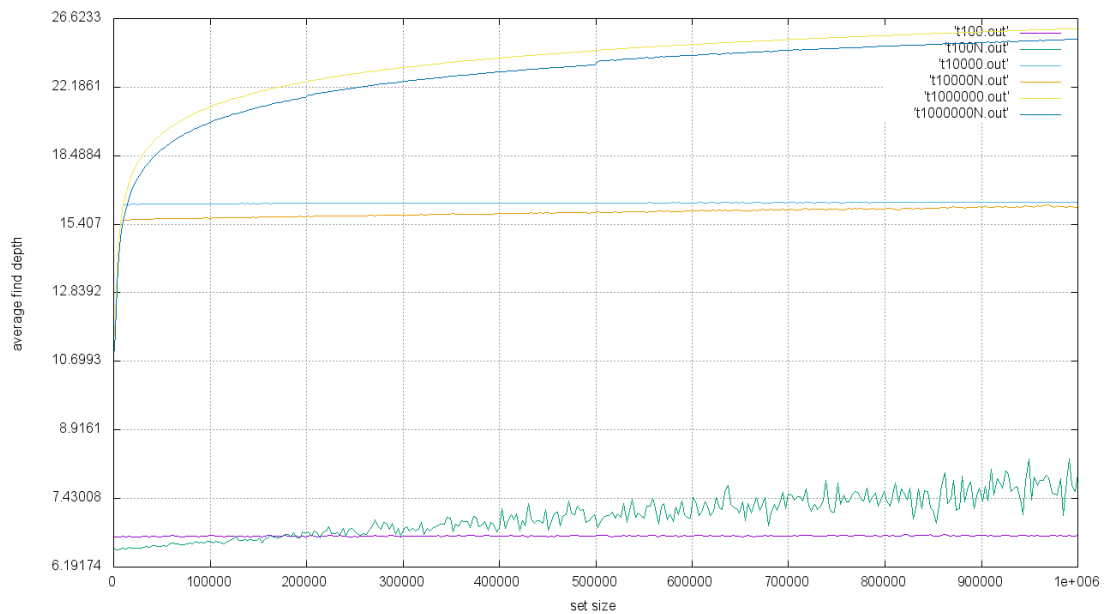
Na tomto grafu vidíme, že hloubky findu se pohybují kolem stálých hodnot jen s jemným rozptylem. Počáteční nárůst, který lze na tomto grafu sledovat pouze u t1000000, se vyskytuje u všech grafů. (viz ./report/graphs) Tento nárůst probíhá vždy do velikosti vyhledávané množiny a pak se ustálí. Je také vidět, že čím menší je vyhledávaná množina, tím menší je průměrná délka nalezení. To lze usuzovat z toho, že tato množina se dostane blíže ke kořeni, a tak menší vyhledávané množiny musí mít i menší hloubku vyhledání.

## 0.2 Naivní implementace



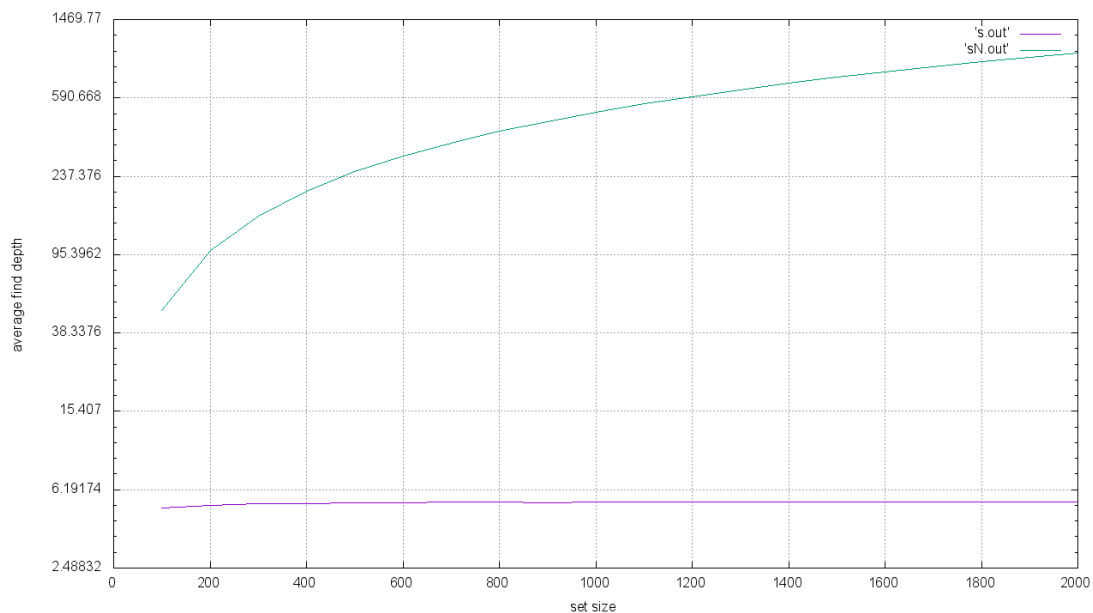
Pro naivní implementaci vidíme, že značně zvýšila rozptyl při vyhledávání. Z čehož lze usuzovat, že si tato verze při splay odsouvá často použité vrcholy příliš rychle do hloubky.

## 0.3 Porovnání standartní a naivní implementace



Je vidět, že již pro  $t=10000$  je hloubka naivních vyhledání o něco menší než u standartního. Nicméně narozdíl od standartní verze je délka cesty mírně rostoucí i po dosažení počtu nalezení rovné velikosti hledané množiny. Z toho lze usuzovat, že strom degraduje.

## 0.4 Sekvenční test



Zde je vidět, že naivní verze potřebuje přibližně dvakrát delší hloubku než standartní. Také je vidět, že jednoduché rotace „rozbíjejí” strom a dělají ho vyžším. Takže stálé vyhledávání nových klíčů naivní strom rozbíjí, nicméně standartní implementace zůstává stabilní.