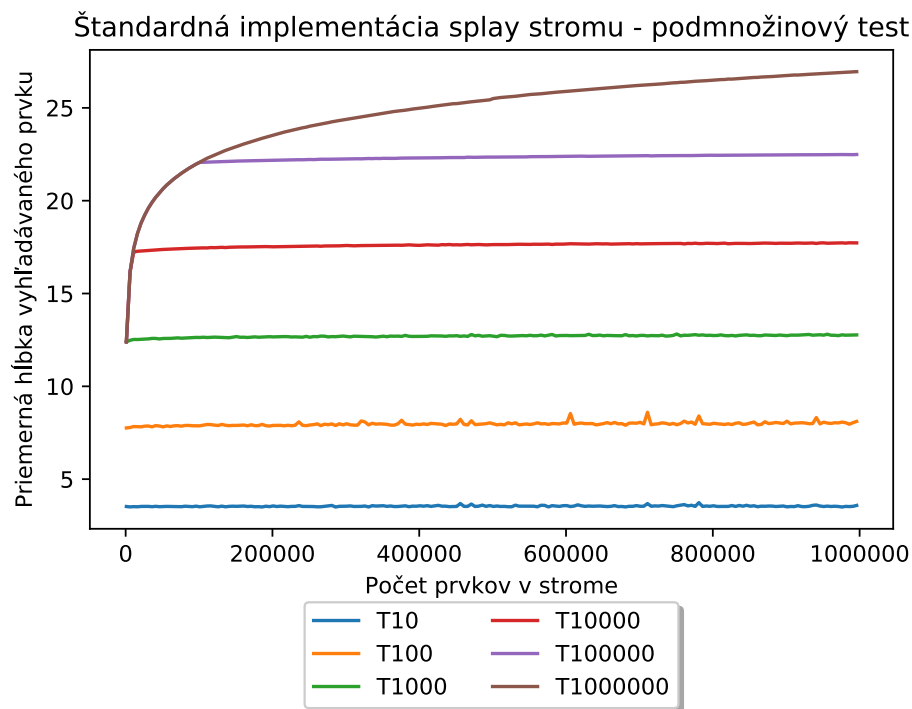


1 Splay

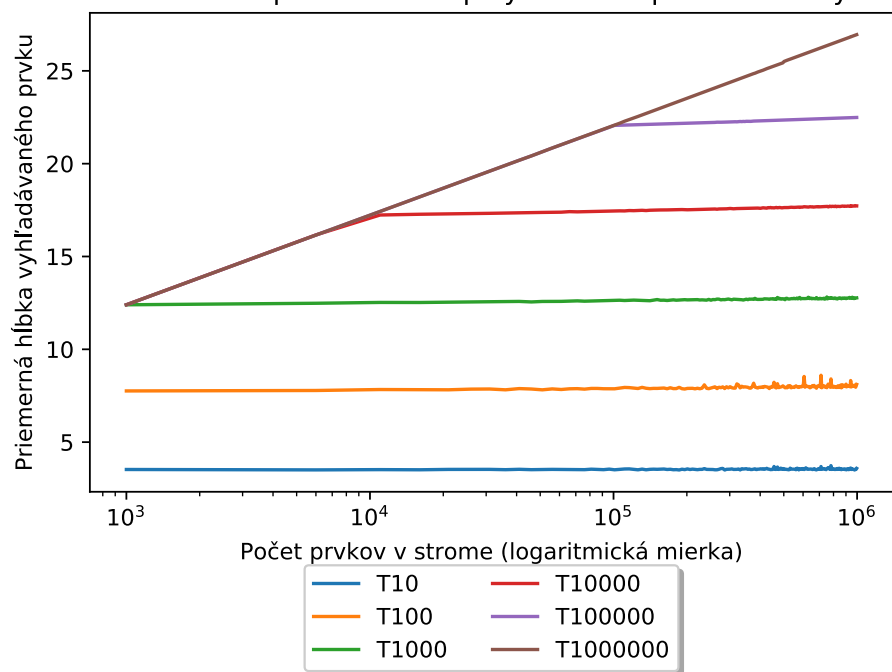
1.1 Štandardná implementácia v podmnožinovom teste



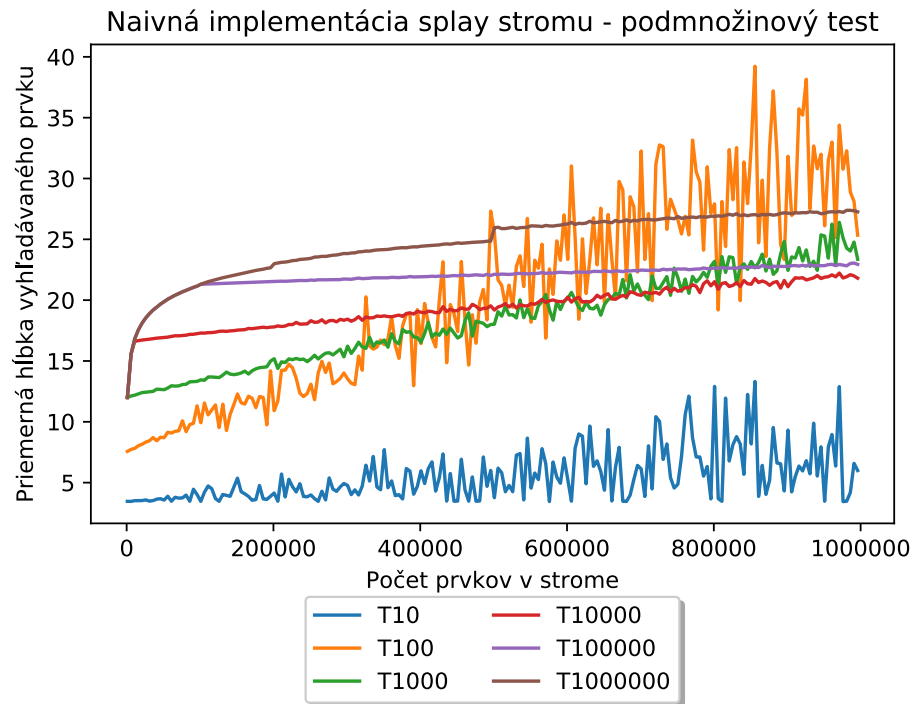
Z grafu vidíme, správanie Splay stromu, keď v ňom vyhľadávame stále tie isté prvky dokola. Vidíme, že zložitosť vyhľadávania nezáleží na celkovom počte prvkov v strome n , ale iba na veľkosti vyhľadávanej množiny. Samozrejme, ak je v strome menej prvkov ako je veľkosť množiny T , čas vyhľadania je menší. To vysvetľuje krivku pre $T = 1\,000\,000$. Všimnime si, že aj krivka pre $T = 100\,000$ je na začiatku zakrivená, a približne od $n = 100\,000$ už zostáva konštantná. Pre n menšie ako T čas stúpa logaritmicky, pre $n \geq T$ zostáva konštantný. To je z toho dôvodu, že všetky vyhľadávané prvky sa operáciou splay posunú blízko koreňa.

Ten istý graf v logaritmickej mierke:

Štandardná implementácia splay stromu - podmnožinový test



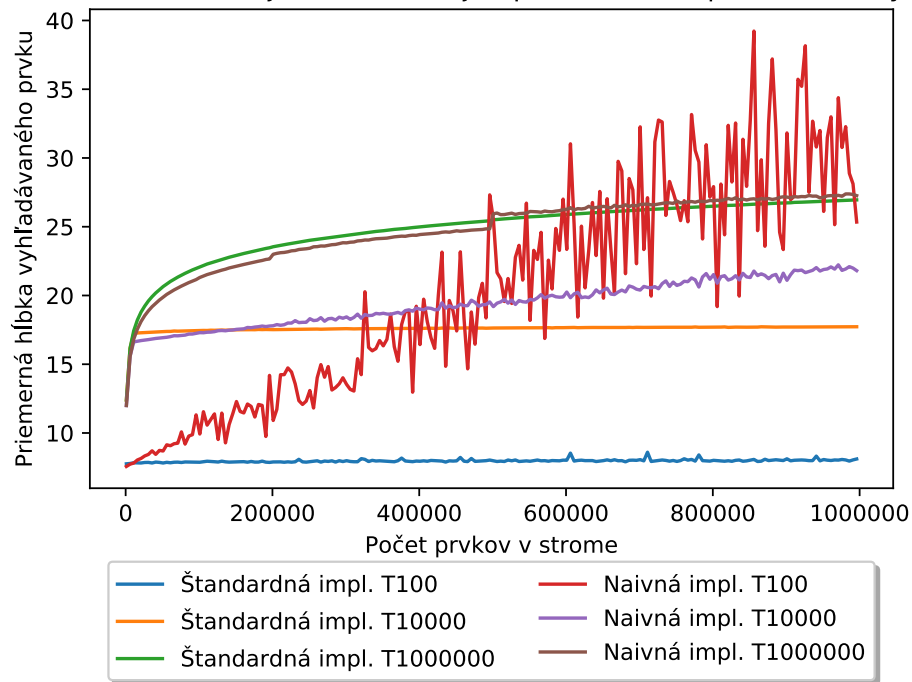
1.2 Naivná implementácia v podmnožinovom teste



Na tých istých vstupných dátach ako v predchodzom prípade sme spustili aj naivnú implementáciu splay stromu. Priemerná hĺbka vyhľadávaného prvku je menej predvídateľná. Okrem toho môžeme odhadnúť, že zložitosť operácie find závisí na celkovom počte prvkov v strome. Je to najviditeľnejšie pre $T = 100$. Naivný splay pravdepodobne nedokáže udržať vyhľadávané prvky blízko koreňa.

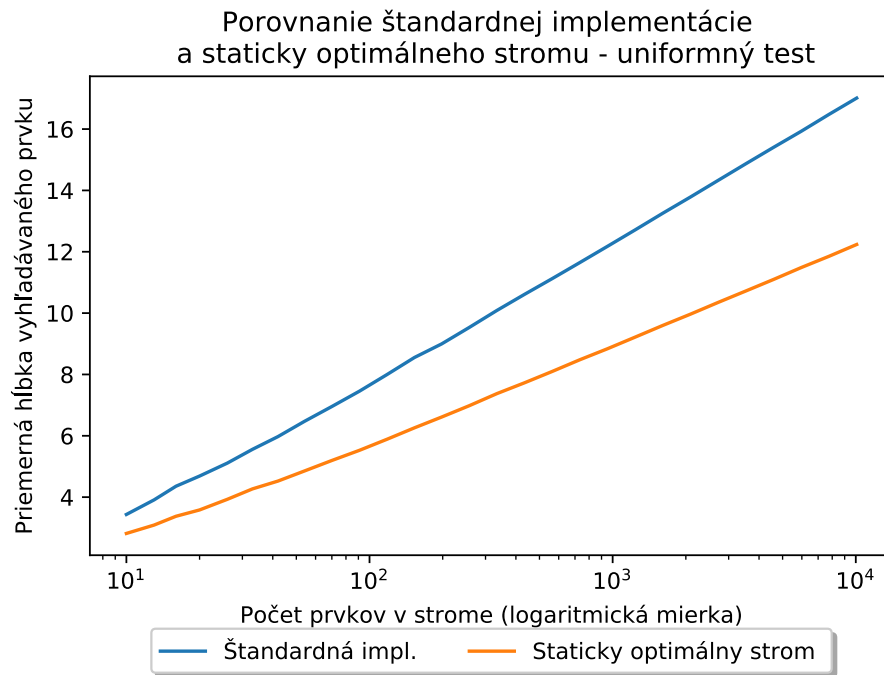
1.3 Porovnanie naivnej a štandardnej implementácie v podmnožinovom teste

Porovnanie naivnej a štandardnej implementácie - podmnožinový test



Porovnanie naivnej a štandardnej implementácie dopadlo lepšie pre štandard. Pre malý počet prvkov v strome pozorujeme mierne lepšiu naivnú implementáciu pri $T = 10\,000$ a $T = 1\,000\,000$.

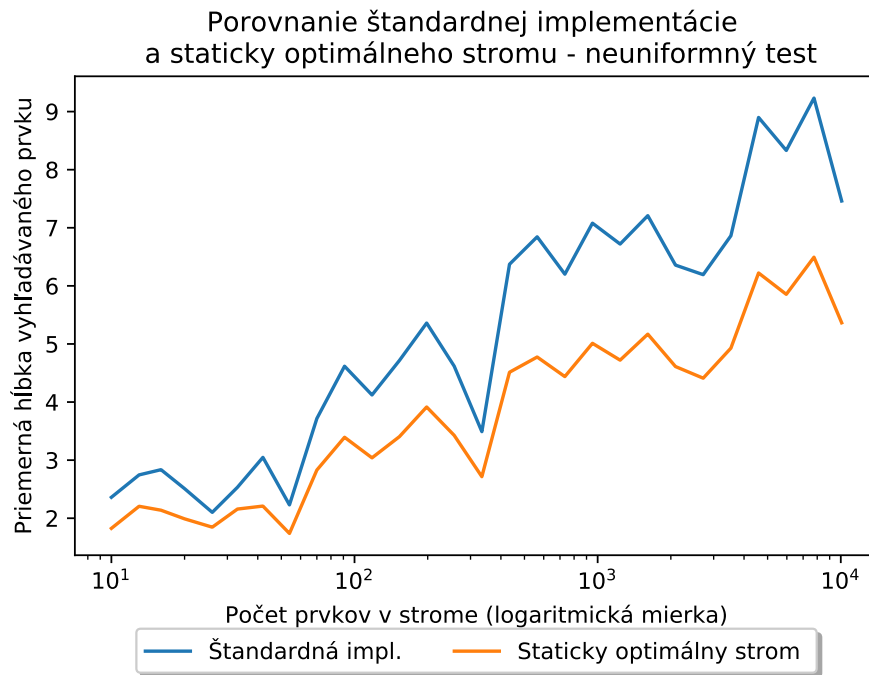
1.4 Porovnanie štandardnej implementácie a staticky optimálneho stromu v uniformnom teste



Z grafu pozorujeme, že s počtom prvkov v strome rastie hĺbka logaritmicky pri oboch stromoch, pri Splay strome rastie rýchlejšie. Odhadol som multiplikatívne konštanty jednoducho tým, že v každom nameranom bode som výslednú hĺbku podelil logaritmom počtu prvkov v strome a následne spriemeroval:

- Konštanta pri štandardnej implementácii: 1.18
- Konštanta pri staticky optimálnom strome: 1.03

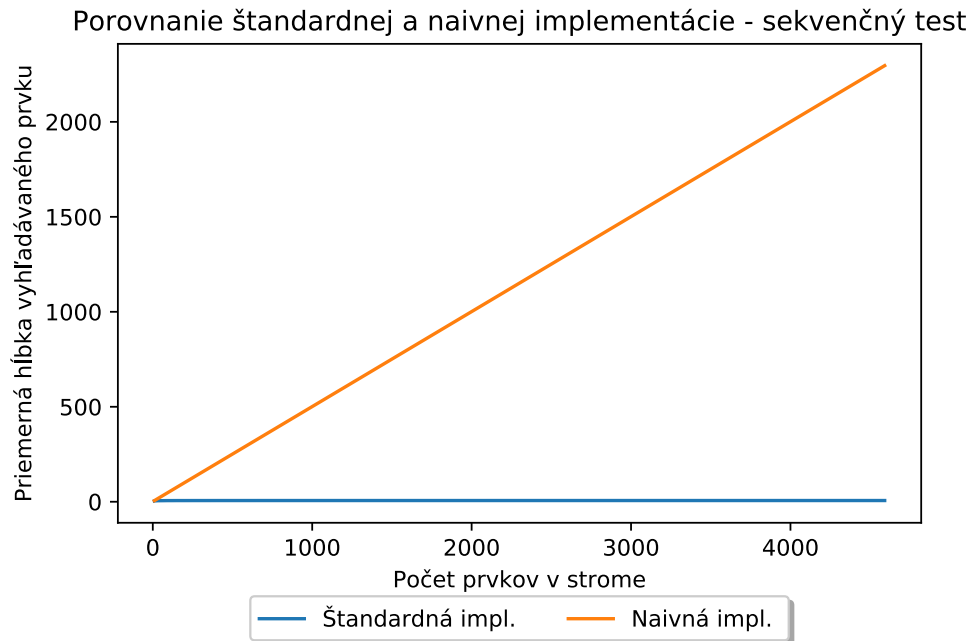
1.5 Porovnanie štandardnej implementácie a staticky optimálneho stromu v neuniformnom teste



Pri neuniformnom teste je ťažké odhadnúť akýkoľvek trend, pretože priemerná hĺbka vyhľadávaného prvku príliš závisí na rozdelení vstupných dát a to zahmlieva závislosť na počte prvkov v strome.

Experiment potvrdzuje vetu z prednášky, že splay strom urobí asymptoticky rovnako veľa operácií ako staticky optimálny strom. Optimálny strom má síce vždy menší priemer, ale vždy keď sme namerali menšiu hodnotu pre staticky optimálny strom, namerali sme umerne menšiu hodnotu aj pri splay strome.

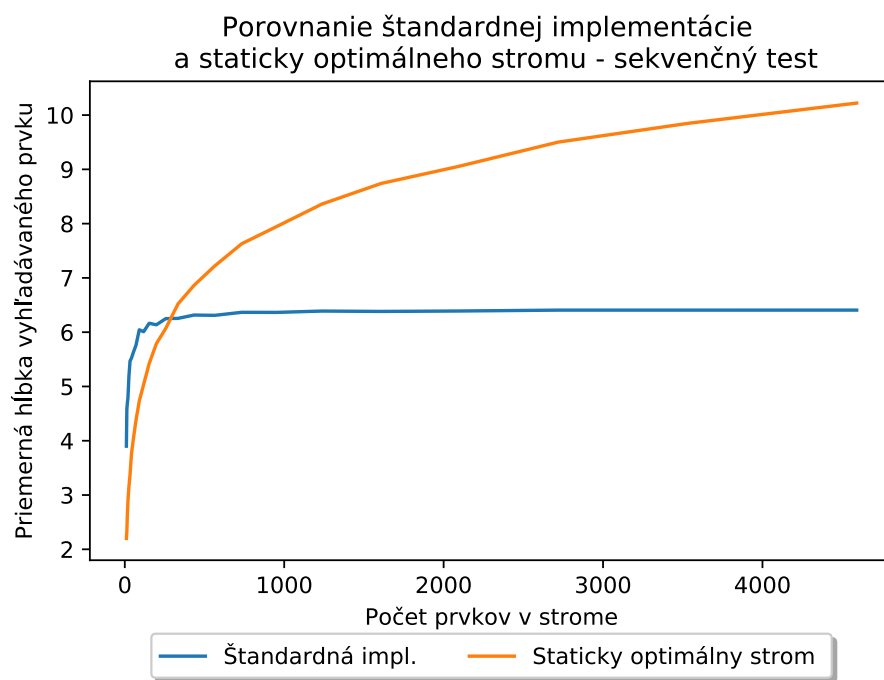
1.6 Porovnanie štandardnej a naivnej implementácie stromu v sekvenčnom teste



Namerané hodnoty štandardnej implementácie korešpondujú s vetou z prednášky, že ak postupnosť vyhľadávanií obsahuje rastúce prvky tak celkový čas na vyhľadanie je konštantný.

Priemerná hĺbka vyhľadania naivnej implementácie rastie lineárne.

1.7 Porovnanie štandardnej implementácie a staticky optimálneho stromu v sekvenčnom teste



Splay strom má v tomto porovnaní “výhodu” v tom, že sa môže dynamicky upravovať počas vyhľadávania. Priemerná hĺbka vyhľadania v staticky optimálnom strome rastie aj pri sekvenčnom teste logaritmicky.