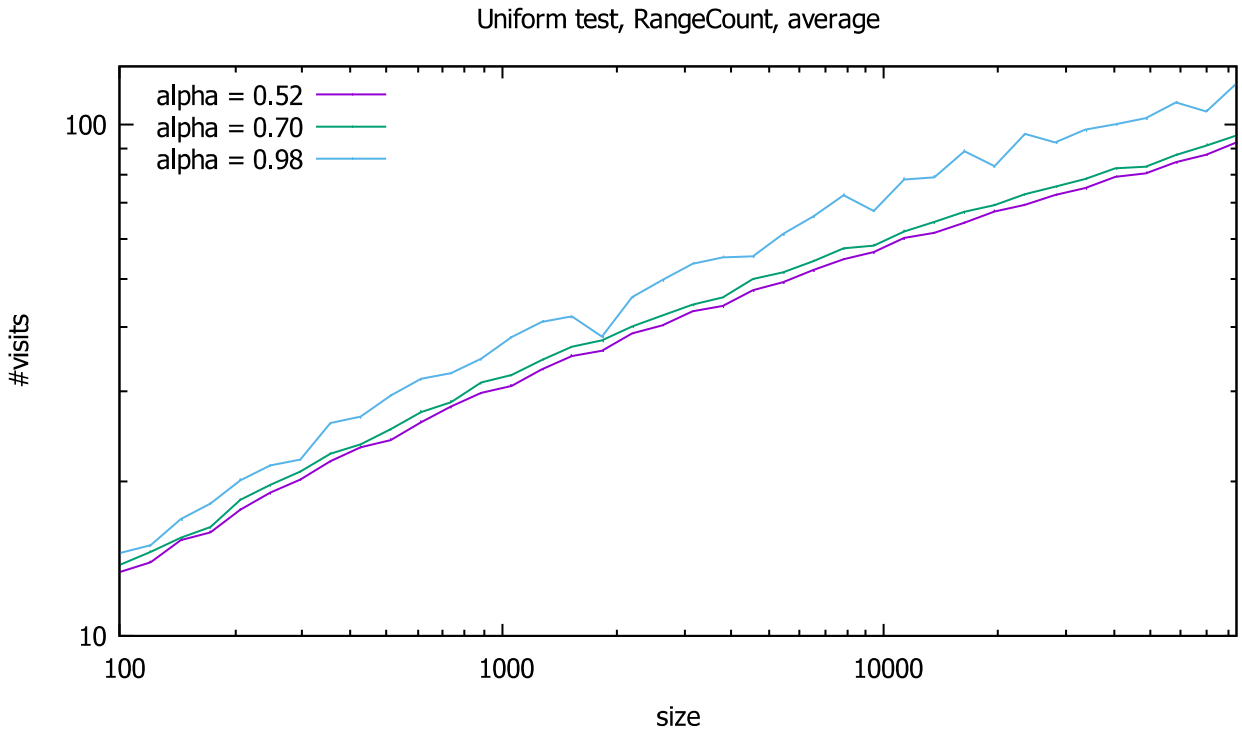
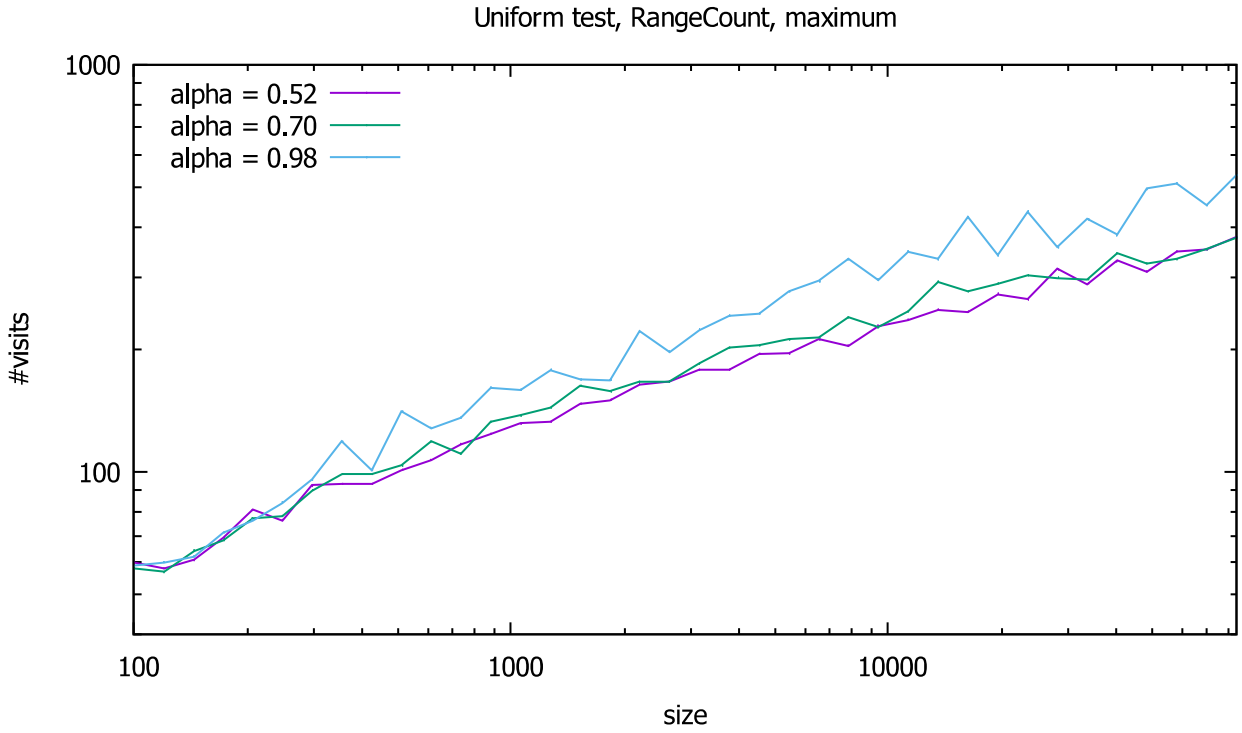
V textu budu hodnotě 0.52 přezdívat “nízká alfa”, hodnotě 0.70 budu přezdívat “střední alfa” a hodnotě 0.98 budu přezdívat “vysoká alfa”. Dále budu skloňovat (česky) alfa, alfy, alfě, …

RangeCount v uniformním testu

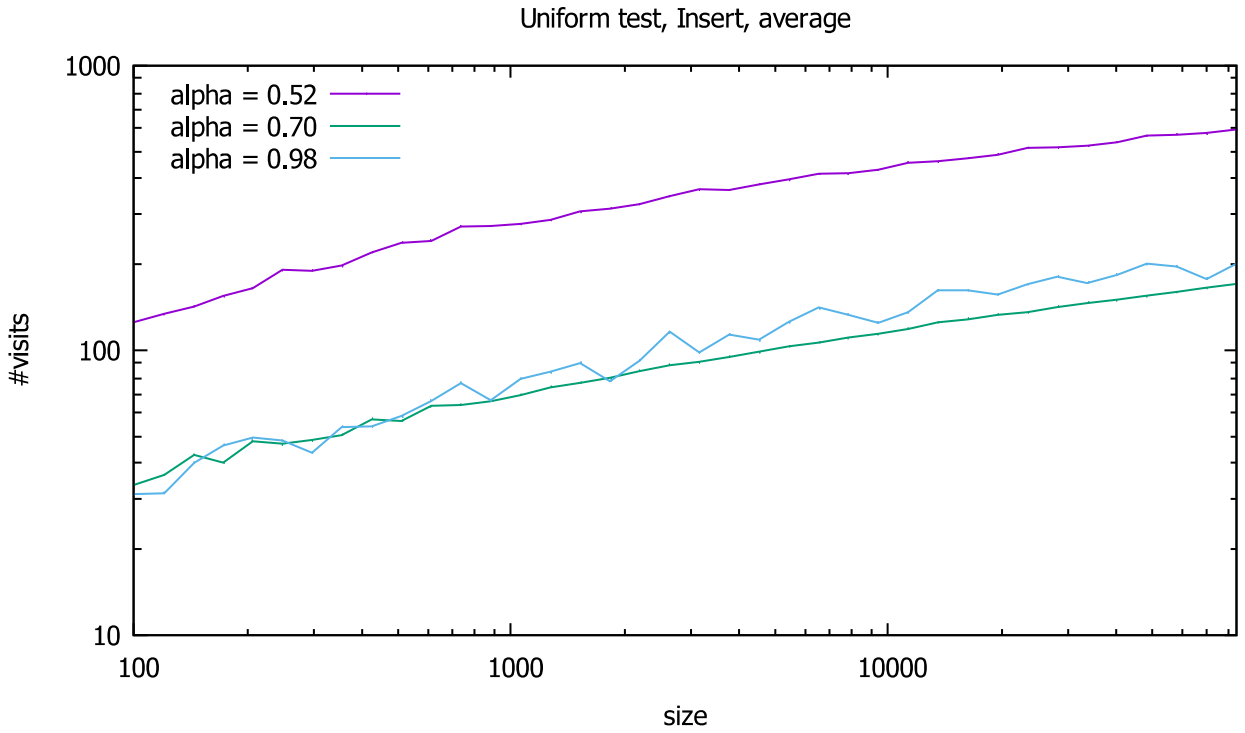




Počet návštěv vrcholů operací RangeCount (průměr i maximum) je v polylog(velikost stromu). Podle křivky pro střední alfa mi vychází exponent pro průměrný počet návštěv vrcholů při operaci RangeCount přibližně 2.14 a pro průměrný počet návštěv vrcholů při operaci Insert (viz obrázky

níže) přibližně 1.8 (exponent je myšlen tak, na jakou konstantu se má umocnit logaritmus počtu vkládaných prvků, abychom získali počet návštěv vrcholů, multiplikativní konstantu zanedbávám). Pro ostatní křivky (nízká alfa, vysoká alfa) je asymptotické chování velmi podobné. Teorie tvrdí, že by tento exponent měl vyjít 2 (přesně). Takže je mé měření v mírném nesouladu s teorií.
Nízká alfa dává rychlejší RangeCount, vysoká alfa dává pomalejší RangeCount. Čím vyšší je alfa, tím horší je vyvážení a tedy i vyšší hloubka stromu.
V uniformním testu je ale RangeCount dost rychlý i pro vysokou alfu. Je to kvůli tomu, že náhodně budovaný binární vyhledávací strom má v průměrném případě logaritmickou hloubku i bez vyvažování. Samozřejmě to vyvažování trochu pomůže i v uniformním testu, protože začneme mít garantovaný lepší základ logaritmu než při "úplné anarchii", ale není to markantní.

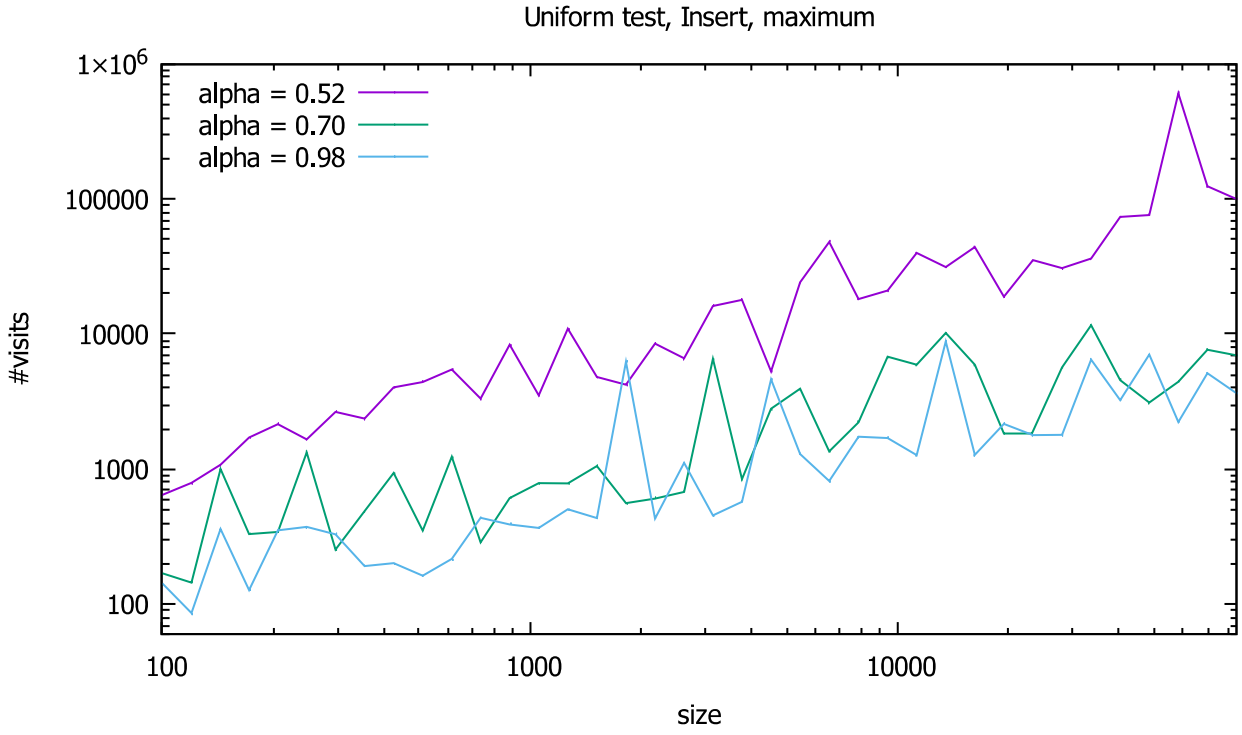
Insert v uniformním testu



Asymptotické chování je pro všechny alfy polylogaritmické, jak už jsem naznačil v předchozí části.

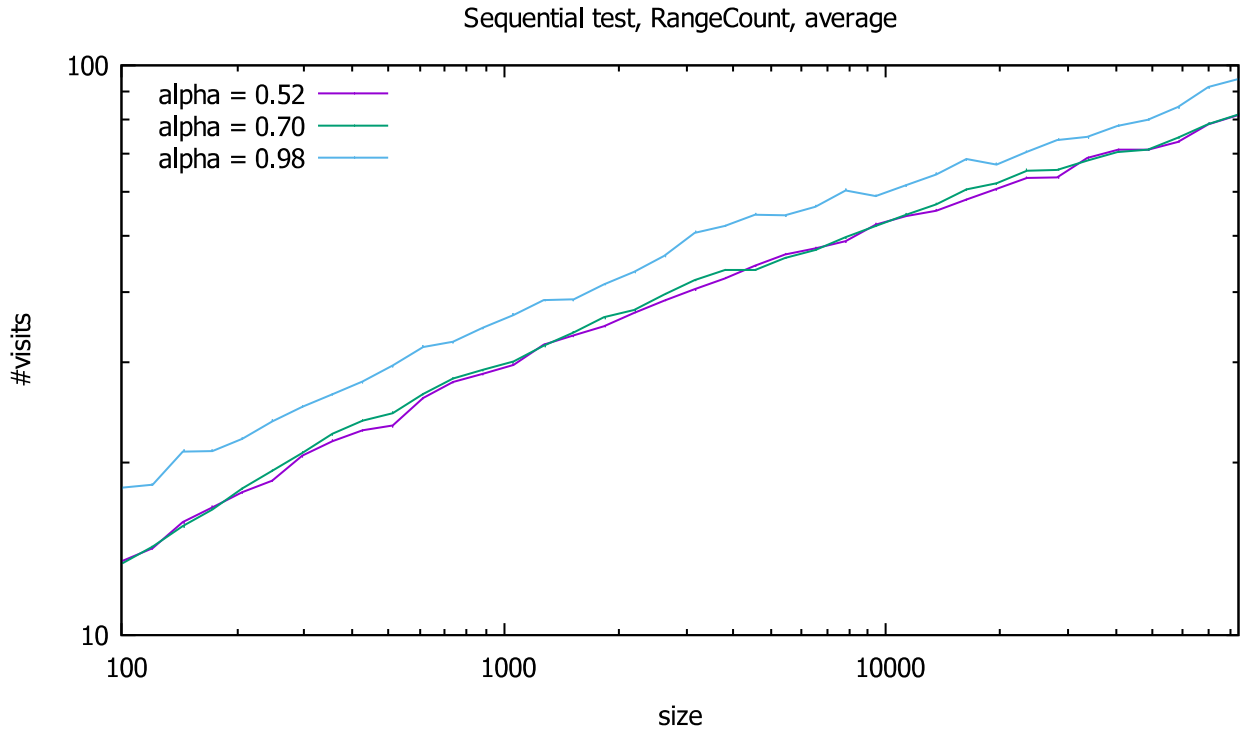
Průměrný počet návštěv při Insertu vychází nejlépe při normální alfě. Nízká alfa vede k tomu, že rebuild je spouštěn velmi často, takže Insert trvá v průměru dlouho. Vysoká alfa je sice výhodná na to, že jsou rebuildy vzácné, ale vede k tomu, že musí Insert občas sestupovat do velké hloubky, takže vychází o trochu hůře než normální alfa. Zhoršení není velké díky náhodnému pořadí vkládání, viz komentář u operace RangeCount. Naopak zhoršení pro malé alfa je velmi markantní, protože rebuildy jsou s ní extrémně časté a lepší vyvážení nemá moc velký význam.

Všimněte si, že světle modrá křivka (vysoká alfa) je hodně zašuměna, protože při vysoké alfě nastává jen málo rebuildů (takže průměrujeme respektive maximujeme malý počet hodnot).



Maximální počet návštěv vrcholů při Insertu je opravdu vysoký, protože občas dojde k rebuildu, který sahá na opravdu hodně vrcholů (potenciálně až na všechny kromě největšího Y-stromu, když dojde k rozvážení kořene X-stromu). Pokud vás zajímá, která z těch vysokých hodnot je “nejméně špatná”, pak mohu říct, že maximální Insert vychází nejlépe při vysoké alfě (nejnižší počet rebuildů zároveň dává nejnižší riziko nějakého opravdu velkého rebuildu, hlavně je při vysoké alfě nízké riziko rebuildu celého stromu a to je tím nižší, čím větší počet prvků je již vložen do stromu (ale dílem náhody může někdy ten maximální rebuild vyjít pro vysokou alfu hůře než pro ostatní alfy; zde to nastalo při testu o velikosti 1826)). Při studování nejhoršího případu (maxima) se neprojevily Inserty bez rebuildu, tudíž střední alfa nevyhrála nad vysokou alfou (v globálním trendu), zatímco při studiu průměrného případu měla střední alfa výhodu v mělčích Inserted bez rebuildu.

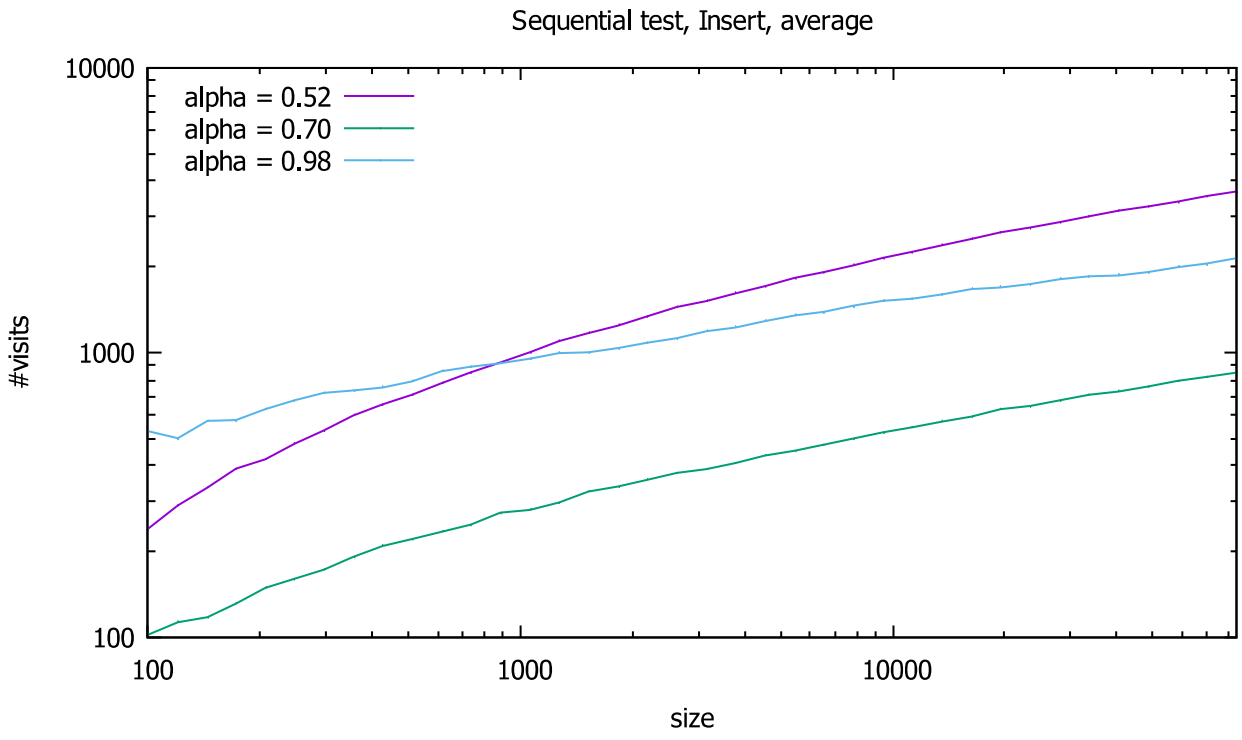
RangeCount v sekvenčním testu

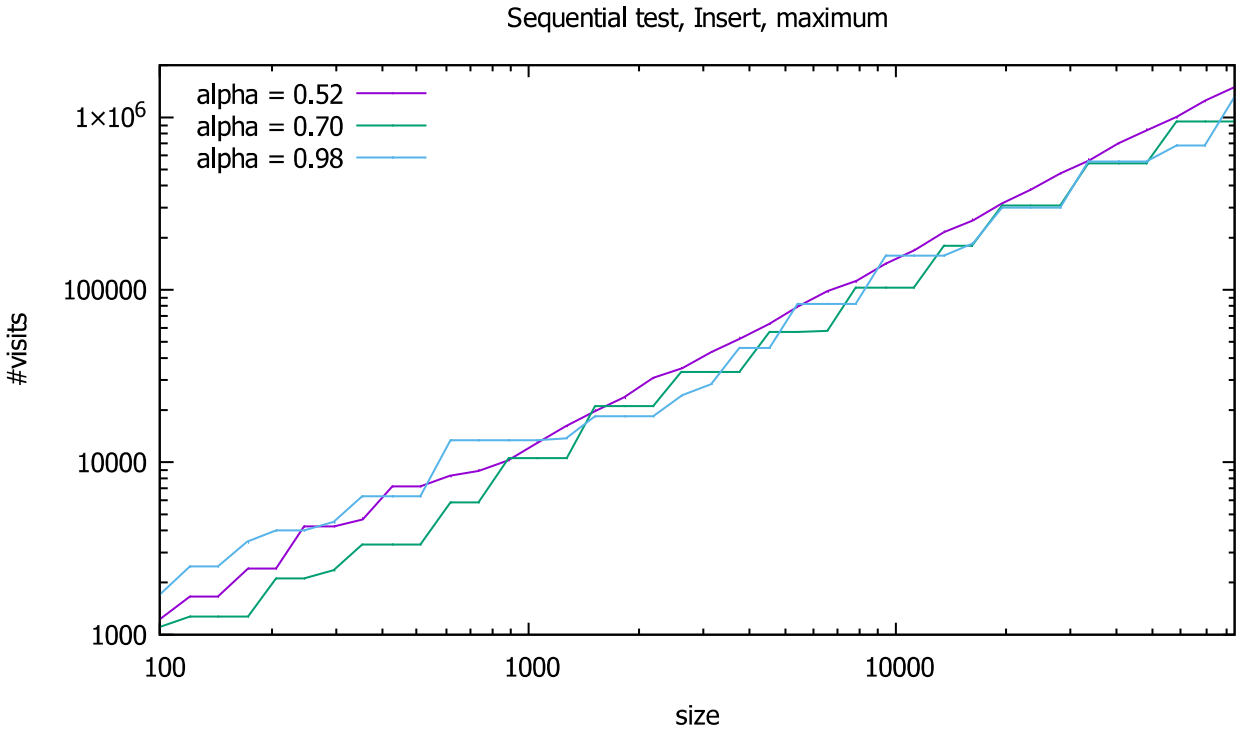


Při sekvenčním testu je složitost operace RangeCount téměř stejná jako při uniformním testu, akorát jsou výsledky méně zašuměné (neboť ve vstupních datech není element náhody).

V případě zkoumání maximálního počtu navštívených vrcholů operací RangeCount jsou opět výsledky sekvenčního testu skoro stejné jako výsledky uniformního testu, akorát je v sekvenčním testu o trochu větší nevýhoda vysoké alfy (protože tu s jistotou nastává nejhorší případ hloubky).

Insert v sekvenčním testu





Insert je v sekvenčním testu opravdu zajímavý. Prvně se musíme smířit s tím, že nám při všech variantách RangeTrees budou vycházet hodně vysoké hodnoty počtu návštěv, protože rebuildy jsou nevyhnutelné a to včetně těch úplně největších.

Pokud jde o maximální počet návštěv vrcholů, ten je při všech alfách přibližně stejný a roste ještě hůře než lineárně. Na přednášce nemáme teoretický výsledek týkající se maximálního Insertu

(známe akorát amortizovanou složitost, která odpovídá průměru). Je ovšem jasné, že občas musí dojít k přegenerování všech stromů kromě jednoho (a to ve chvíli, když se rozváží kořen X-stromu). Maximální časová složitost Insertu by tedy měla růst podle rekurentního vztahu:
$T(N) = 2 * T(N/2) + c * N * \log N$
Metodou "tipni a ověř" jsem dospěl k výsledku:
$T(N) = d * N * \log(N) * \log(N)$, kde $d = c/2$
Naměřená maximální složitost Insertu mi vyšla podstatně nižší než tento teoretický odhad, ale stále superlineární.
"Zuby" tu vznikají z toho, že v některých velikostech testu nastal poslední rebuild až při vkládání posledního vrcholu (což je nejhorší) a v jiných velikostech testu nastal poslední rebuild s jistým odstupem od konce (což umožnilo rebuildovat trochu menší strom).

A teď k průměrnému případu Insertu v sekvenčním testu, ten mi připadal nejzajímavější ze všeho... Vidíme že nejlepší je střední alfa a její exponent vychází asi 2.23 (opět vůči logaritmu velikosti testu). Nízká alfa i vysoká alfa byly mnohem horší než střední alfa. Na malých velikostech je nejhorší vysoká alfa (chození do příliš velké hloubky). Na velkých velikostech je nejhorší nízká alfa (extrémně časté rebuildy).

Shrnutí

Střední alfa vyšla ve všech testech dobře až výborně (viz zelené křivky na všech osmi obrázcích). Proto bych si z nabízených hodnot vybral střední alfu v případě, že bych měl svoji datovou strukturu reálně používat. Ještě před finálním rozhodnutím bych si však pohrál s jinými hodnotami, například jestli by hodnota 0.75 nebyla lepší než testovaná hodnota 0.7 (střední alfa).

Výsledky měření mohou být ovlivněny (zvýšeny, tj. zhoršeny) tím, že ve speciálních případech Insertu můj program spouští rebuild podstromu a poté i rebuild většího stromu, čímž anuluje význam menšího rebuildu (a zbytečně se zvyšuje počet operací). Na asymptotickou časovou složitost však toto nemá vliv, protože velikosti podstromů exponenciálně klesají při pohybu směrem dolů (což je zajištěno předchozím vyvažováním).

Situace, ve kterých mi vyšel exponent vyšší než 2, ještě nutně neznamenají, že má moje datová struktura vyšší asymptotickou časovou složitost, než by měla mít. Máme tu omezeně velká data, takže je možné, že akorát pro malé velikosti vycházela nějaká menší multiplikativní konstanta (například proto, že bychom v malých testech nestrefili worst-case pro malé velikosti). A teprve při dlouhém zvyšování velikosti vstupu bychom mohli dosáhnout asymptotického trendu s exponentem 2 přesně.