(размер входни данни – броя вкарвания и търсения, които правя)

Направих три вида тестове за елементи в обхвата [0, 100].

1. Вкарване на случаен елемент и последващо негово търсене.
2. Вкарване на случаен елемент и последващо търсене на: някой от предходните 8 елемента(с вероятност 80%) или случаен елемент от обхвата (с вероятност 20%)
3. Вкарване на случаен елемент и последващо търсене: на него (с вероятност 50%) или на елемент извън обхвата (с вероятност 50%)

Очаквано, Splay дървото работи най-бързо при тест „а“, след което има леко забавяне при тест „б“.

За тест „с“ - при Splay дървото - ако търся извън обхвата и е над дясната граница, трябва да мине през най-десния път на дървото и понеже там вкарвам повтарящите се елементи(ако се повтаря- в дясното поддърво), то би било и потенциално много много дълъг път, затова и при търсене на елемент извън обхвата (по-големи елементи от дясната граница) нещата се забавят драстично (явно дори ротациите, когато го връщам към корена не успяват да го пребалансират достатъчно). (съответно за по-голям набор от входни данни ~2^21 + )

AVL се държи по сходен начин – при тест „а“ работи най-бързо(явно все още са „близко“ в паметта възлите нужни за търсене на елемента, който преид малко е бил вкаран). При тест „б“ и „с“ има леко забавяне(при по-голям набор от входни данни ~2^21+). Разликите между „б“ и „с“ са малки.

Във всички тестове AVL дървото се държи по-добре от Splay дървото(Може би се дължи на това, че имплементацията ми на splay дървото не е много близка до оптимална), но разликата е ~<20%.

По отношение на използваната памет, AVL използва повече, има допълнителна променлива за баланса на всеки възел, при мен е char тип(1 байт), но са нужни 3 бита. Подравнява се на 16 байта памет за всеки възел, за разлика от Splay-а, където всеки възел е 12 байта. Реално разликата в паметта може да се направи доста по-малка(ако не се подравнява), но с подравняване е 25%.

Направих и тестове със случайни висла в интервала [0, размера\_на\_входните\_данни) и при 2^23 входнни данни Splay дървото започва да работи по-бързо, но това вероятно се дължи и на факта, че случайните входни данни го „балансират“ донякъде и по-добре.

(Същи и при дървете, в които не вкарвам дибликати, но за 100 елемента(макс 100 възли) данните не са много адекватни и не ги разглеждам).