Dan Kondratyuk
Data Structures I
Charles University
29 October 2017

Assignment 1

This assignment implements a splay tree for analyzing its algorithmic complexity. There are two types of splay trees implemented: (1) a standard splay tree which utilizes double rotations, and (2) a naïve version which utilizes only single rotations up to the root node. The following graphs plot the average path length for a find operation as a function of the tree size.

Uniform Subset Test

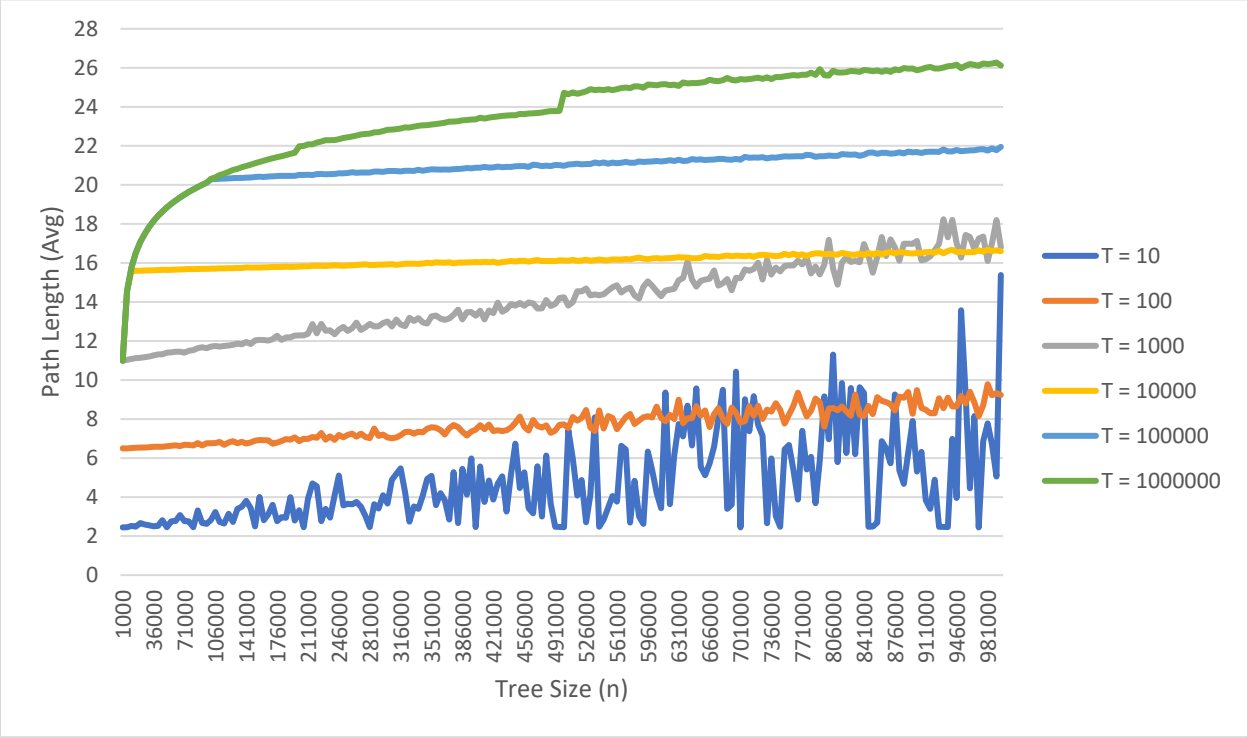
The uniform subset test picks keys uniformly at random from a fixed subset of size T of the inserted keys and issues a find operation for those keys. The graph below plots curves of the average path length needed to find a key in a standard splay tree for a given tree size. Each of the six curves plots a different subset size T.

Uniform Subset Test (Standard)

As T increases, so does the path length. This is simply due to the fact there are more keys being splayed to the root of the tree, further increasing the average number of steps needed to find a key. As standard splay trees have been proved to perform $O(\log n)$ in amortized time, all curves here must be logarithmic up to a constant factor. As T increases, the constant factor increases. The curves for $T \in \{10, 100, 1000\}$ appear linear because the constant factor is very small due to the small subset size (and hence perceived tree size). However, as T approaches the size of the tree n, so too does the curve approach the true asymptotic bound of $O(\log n)$.

The next graph displays a naïve version of the same test, where the splay tree can only perform single rotations on the path to the root.

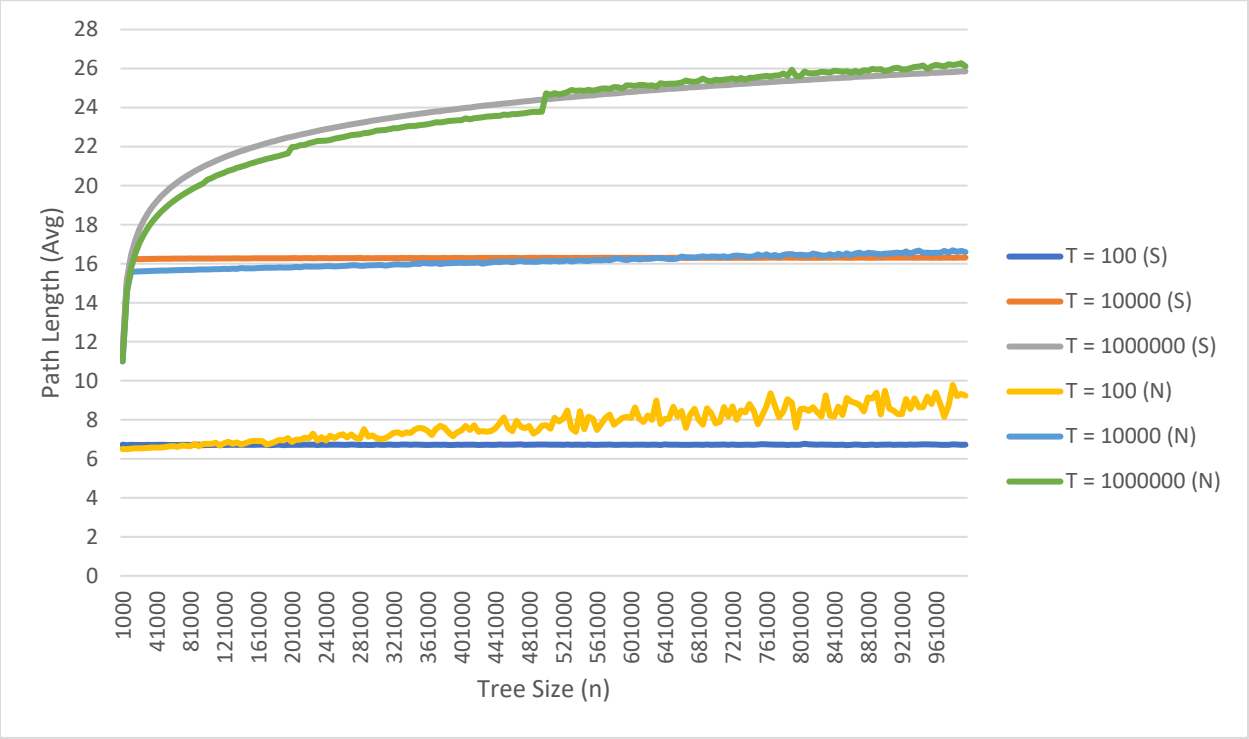
Uniform Subset Test (Naïve)



Like in the previous graph, this graph shows that naïve trees have similar asymptotic complexity, approaching $O(\log n)$ as n and T increase. However, it is immediately visible that naïve splay trees have much more variance in their average path length. The only difference between a standard and naïve splay tree is in the zig-zig step, i.e., whether or not to perform a double rotation in the left-left or right-right case.

The graph below joint plots the standard and naïve curves of the previous two graphs.

Uniform Subset Test (Joint)

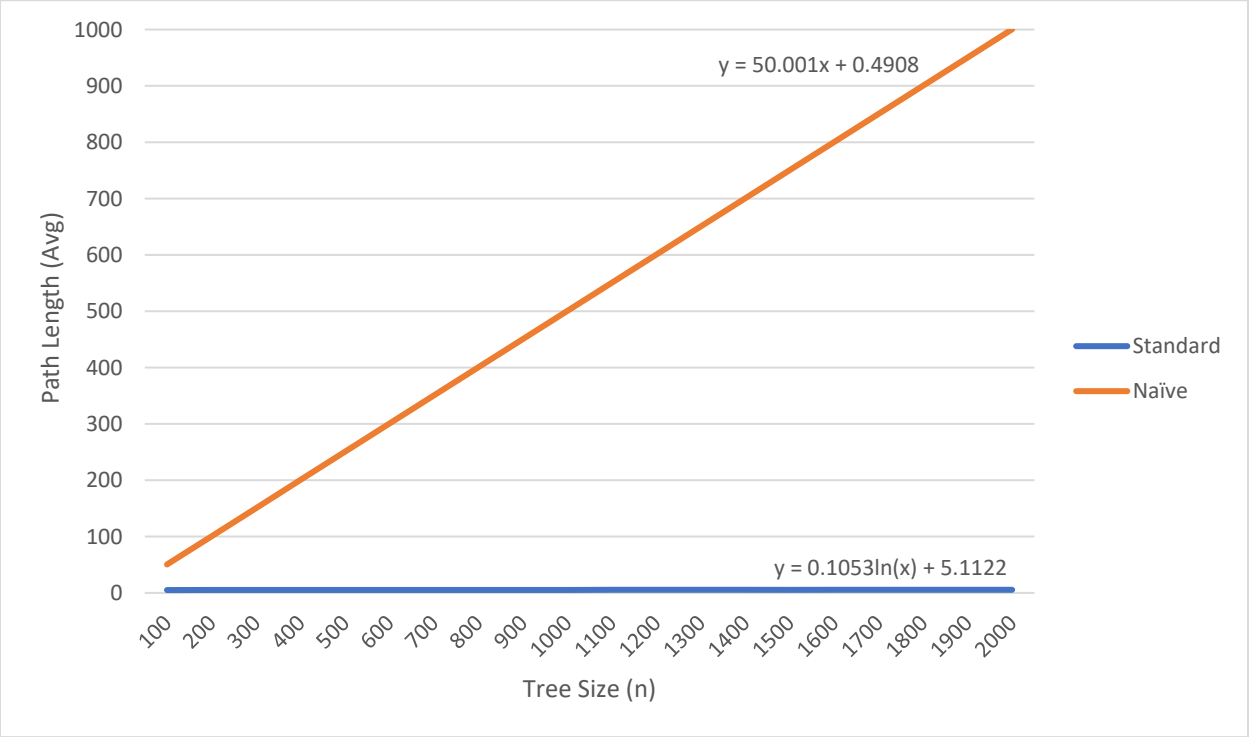


For all values of T, the naïve tree performs asymptotically worse than the standard tree, i.e., the average path length is greater in naïve trees for large values of n. As will be demonstrated by the sequential test in the next section, there exists a case when the average path length is $O(n)$ in the naïve case. Therefore, it is not guaranteed that naïve trees will have $O(\log n)$ performance and thus naïve trees will have asymptotically worse performance than standard trees. The naïve tree appears to have a slight advantage over the standard tree for small values of n, but that advantage disappears as n increases.

Sequential Test

The sequential test inserts keys in consecutive order from the set $\{1, \dots, n\}$, and performs two rounds of issuing a find operation for consecutive keys in the set $\left\{1, \dots, \frac{n}{2}\right\}$. The final graph below plots the curves of the average path length needed to find a key in a standard and naïve splay tree for a given tree size.

Sequential Test (Joint)



The sequential test is where the standard splay tree vastly outperforms the naïve splay tree. The standard tree has a nearly constant average path length of ~5.2, while the naïve tree increases its path length by a factor of 0.5 for each additional searched key.

When the keys are inserted (in either tree), the tree will be linear in height, since an inserted key will always be greater than the root and hence will always be rotated left to form a long chain of left children. When a find operation is performed in the naïve tree, simple rotations will bubble the element from the bottom of the tree up to the top without changing the linear structure. Therefore, a sequence of naïve find operations on consecutive keys will always search through $O(n)$ keys before finding the desired key. When a find operation is performed in the standard tree, the tree is guaranteed not to be linear, as the double rotations will form right children on every left child. As standard splay trees have $O(\log n)$ amortized cost in the worst case, the graph confirms that path length has an asymptotically better bound in the standard tree than the naïve tree.