

Matthew Ryan and Garrett Gilliom

a)

Searching once for a random value in a BST

Number of Items	Time (in seconds)
409	0.06
205	0.0141
102	0.00498
4	1.525×10^{-5}

As the value of n decreases, the time to find the word also decreases because there are less nodes to search. However, it is not linear, as 100x less items results in a time which is about 4000x times faster. This means the efficiency is likely around $O(\log_2 n)$.

b)

Searching 500 times for a random value in a BST

Number of Items	Time (in seconds)
409	.152
205	.0686
102	.0334
4	.0023

While there is still a direct relationship between length of list and time taken searching, this relationship is closer to linear, as 100x less items is about 65x faster of a time.

c) This would mean that the function for insertion is likely already at its maximum efficiency and I would guess the efficiency is around $O(\log_2 n)$. There may be some cases where you could search faster if you knew a value didn't exist, or if you could start from a point further into the BST.