Method	**BST**	**Worst case time complexity of BST**	**Hash Table**	**Worst case time complexity of Hash Table**
Search	0.0009	O(log n)	0.0013	O(n)
Insert	0.002	O(log n)	0.003	O(n)
Delete	0.0025	O(log n)	0.0006	O(n)
Sort	0.208734	O(1)	0.572255	O(n log n)
Range Query (n=10)	0.00075	O(10)	0.158161	O(n)
Range Query (n=100)	0.00750	O(100)	0.158161	O(n)
Range Query (n=1000)	0.075	O(1000)	0.158161	O(n)

For search function, BST is more efficient than Hash Table because the worst case time complexity for BST is log n which is better than n. For insert function, BST is better even though the times we got are pretty similar. For delete, hash table is better than bst even though hash table has a worse worst case time complexity. For sort, BST is definitely better hash table because it is already sorted. For range search, it depends on the number of words because worst case time complexity is the same even for different n in hash table, whereas, it depends on number of n for BST.