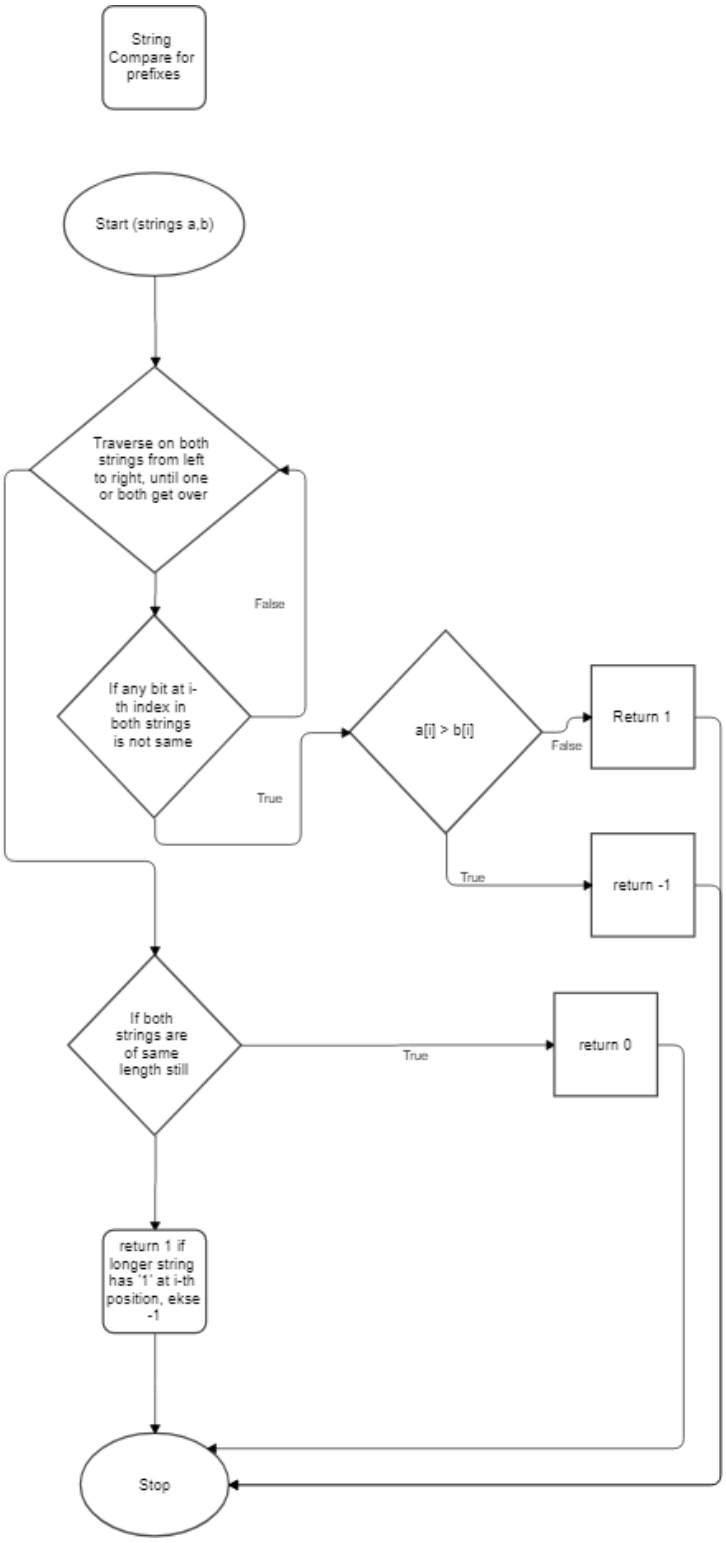
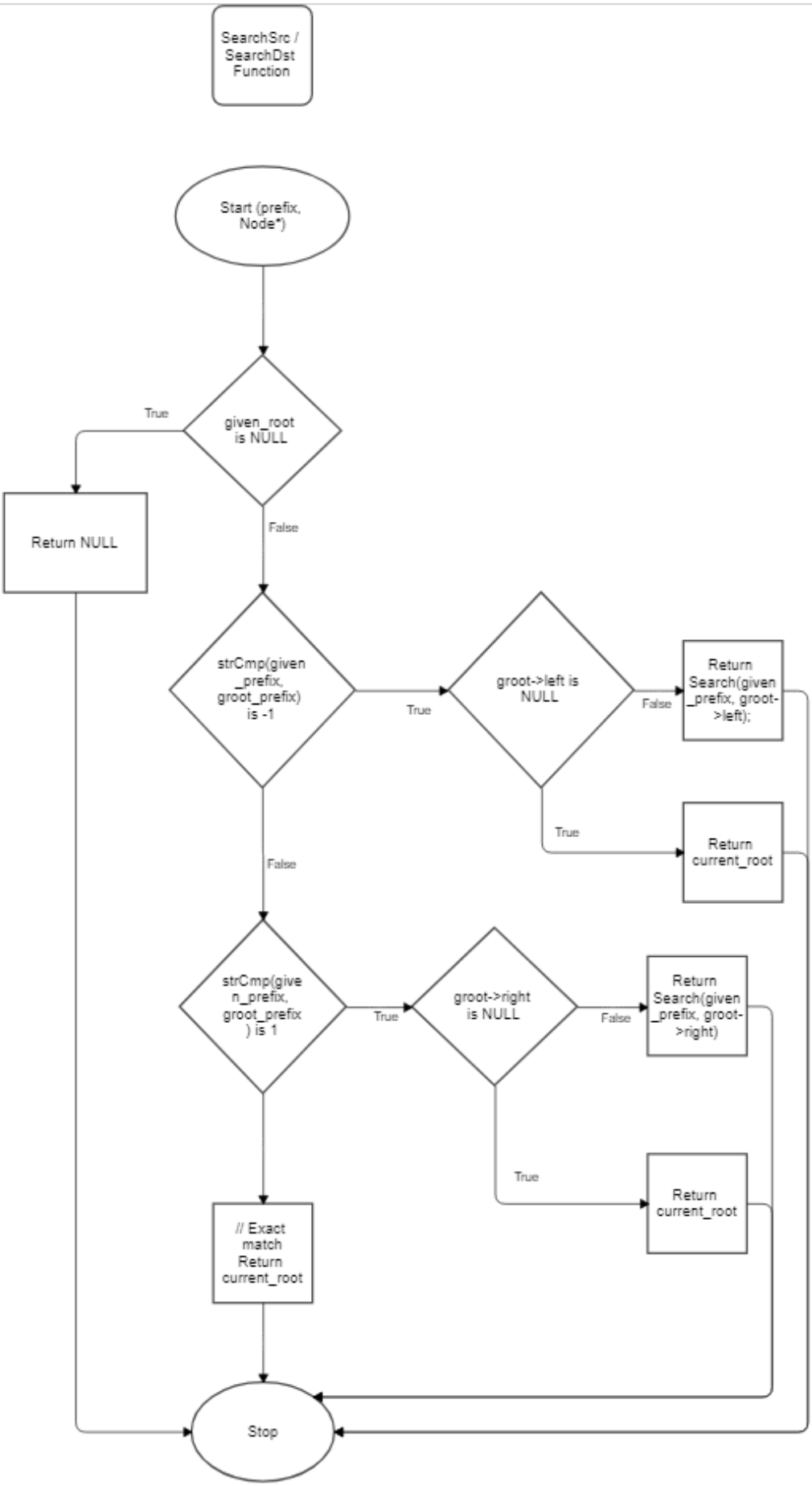
Workflow diagram:




SearchSrc / SearchDst Function
Start (prefix, Node*)
given_root is NULL
True
False
Return NULL
strCmp(given_prefix, groot_prefix) is -1
True
groot->left is NULL
False
Return Search(given_prefix, groot->left);
True
Return current_root
False
strCmp(given_prefix, groot_prefix) is 1
True
groot->right is NULL
False
Return Search(given_prefix, groot->right)
True
Return current_root
// Exact match Return current_root
Stop

Simulation Env:

- ➔ OS: Windows 10 – 64 bit
- ➔ 8GB RAM, 2.27GHz
- ➔ Mingw64 GCC 11

Results:

Time is in micro seconds

DATA STRUCT	T1	T2	T3	T AVG	NO. OF NODES
LINEAR SEARCH	468997	468806	467992	468598	1
H-TRIE	253999	245996	243385	247793	43
H-BSTREE	116002	124242	116983	119075	20

Therefore, in terms of overall performance(time and memory) Hierarchical Binary search Tree works better.

Possible Extension:

- ➔ The binary tree as of now is not self balancing, that can be implemented.
- ➔ Additionally rules should be possible to change, such features can be added.