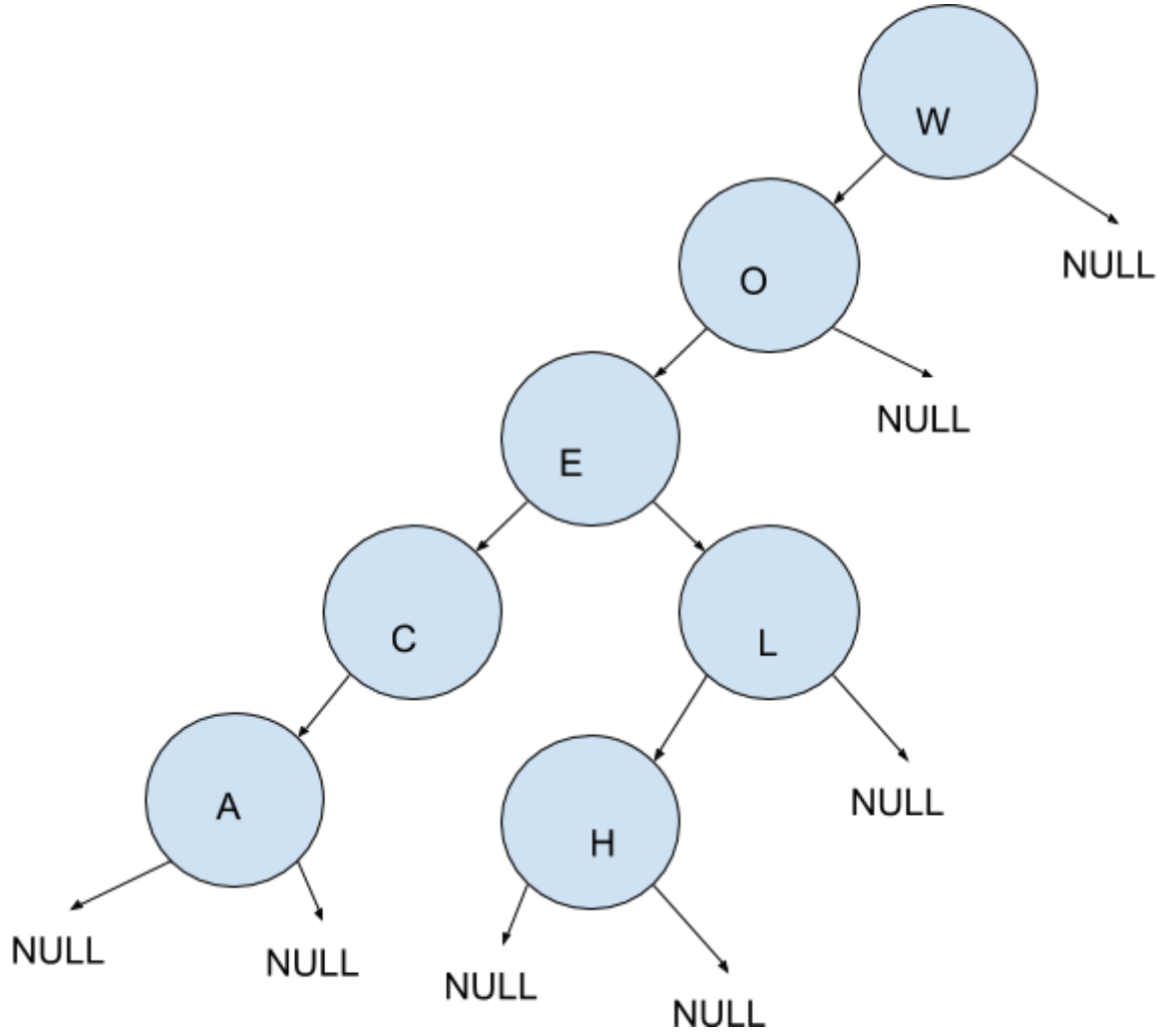
Teori fråga 1:

a)



When building the binary search tree we will start with the first letter that's been entered, in this case 'W', and put it in the beginning of the tree. After that we compare the next letter to it. If it has a key value that is lesser than 'W' it'll go on the left hand side and if it's bigger it'll go on the right hand side. We have the letter 'O' which is lesser than 'W' so it'l go on the left hand side of the tree. We then continue with the next letter, 'E' in our case and we'll compare it first to 'W', and then with 'O'. We see that it's lesser than both of our other keys and will then be on the left side of 'O'. We then move onto our next key and compare it to the keys already in the tree. 'C' is less than all the previous and as such is on the left subtree of 'E'. Our next letter is 'A' which also has the lowest key value that has been entered so it'll be on the left subtree of 'E'. The letter after that is 'L', we compare it with W and it's lesser, then we compare it with O and it's again löesser. After that we compare it with 'E' which has a lesser key value than 'L' so 'L' will be put in the right subtree of the node 'E'. The last letter is H and when we compare it to 'W' and 'O' it has a lesser key value but when compared to 'E' it has a larger key value so it'll be in the right subtree of 'E'. We then compare it to the other node

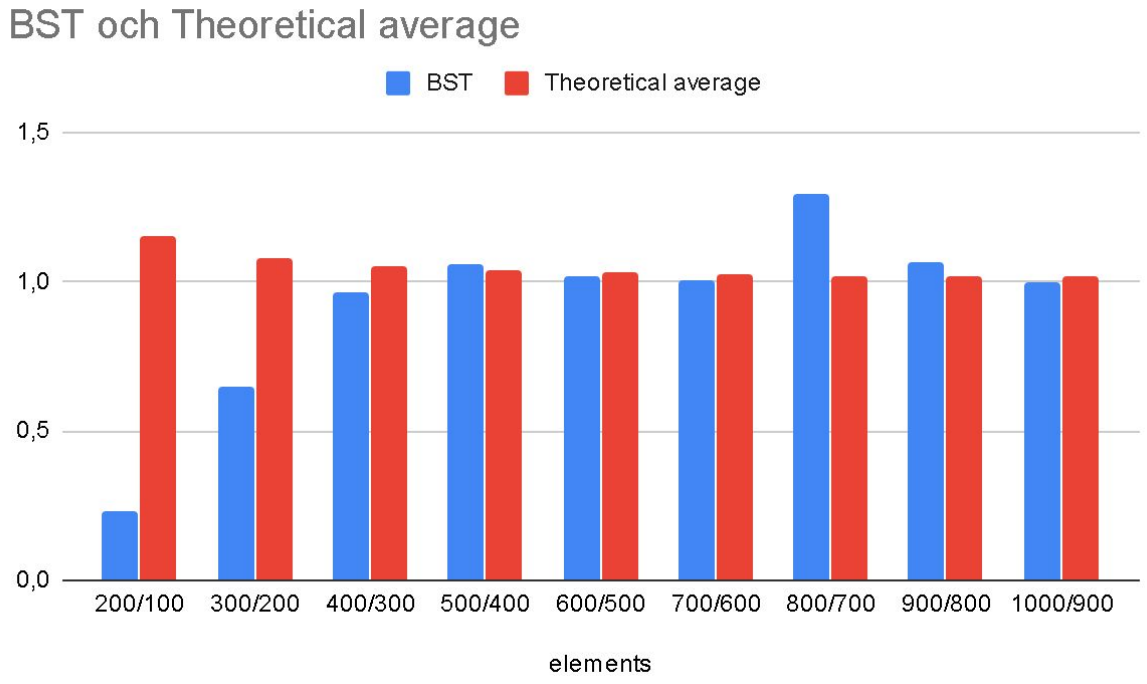
that is in the right subtree of ‘E’ which is ‘L’ which has a bigger key value than ‘H’ so H will be in the left subtree of ‘L’.

b)
When we’re printing it in Postfix order we will first go with the left subtree then the right subtree and lastly we’ll print the root. Using postfix order our result will be:
ACHLEOW

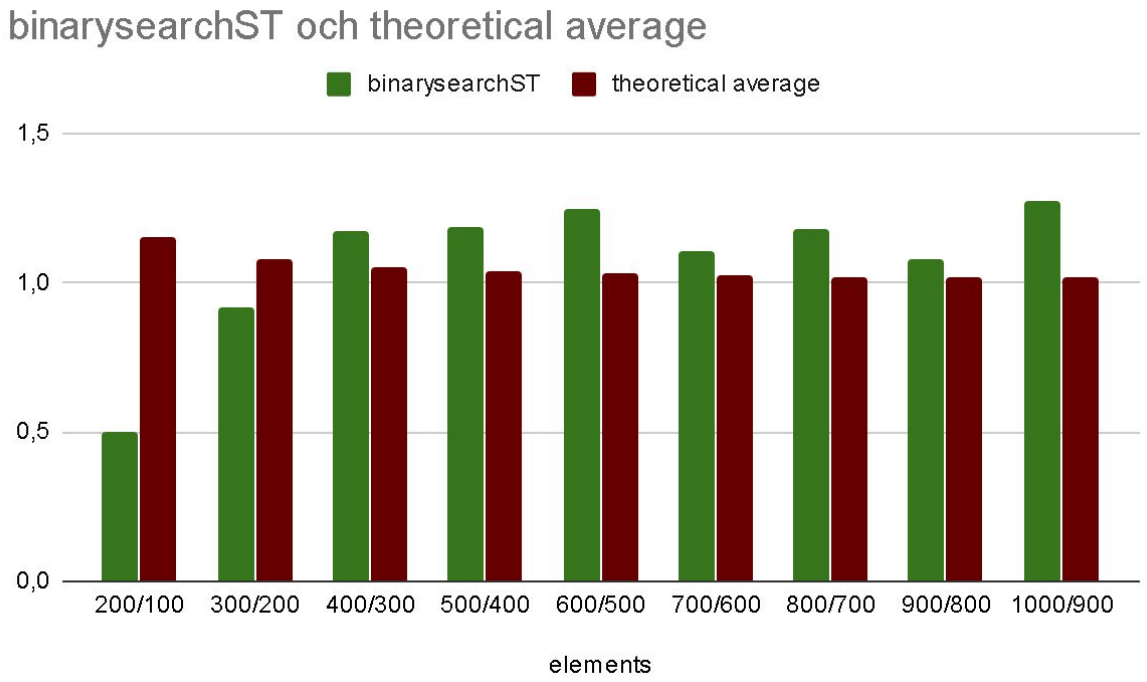
When using in order traversal to print our tree we’ll first print the left subtree then the root and lastly we’ll print the right subtree. That’ll give us the print out of:
ACEHLOW

When traversing the tree using prefix order we’ll first be printing the root then the left subtree and lastly the right subtree. This will give us the output:
WOECALH

2.
The method used to calculate the theoretical time is by calculating the time it’ll take for 100 items and then for 200 items and divide the time it takes for the larger amount of items by the time it takes for the smaller. The same method is used for the real word results. this will enable us to see the improvement in speed.



Doing the same thing with BinarySearchST will give us this chart:


binarysearchST och theoretical average
binarysearchST
theoretical average
1,5
1,0
0,5
0,0
200/100
300/200
400/300
500/400
600/500
700/600
800/700
900/800
1000/900
elements