

Algorithm Assignment 1

Seshagiri Prabhu

August 25, 2013

seshagiriprabhu@gmail.com

1 Time complexity of program

1. Derive the time complexity of the algorithm using Asymptotic analysis.

```
# /usr/bin/python

def bst(self, search, choice):
    ''' Performs binary search based on the choice '''

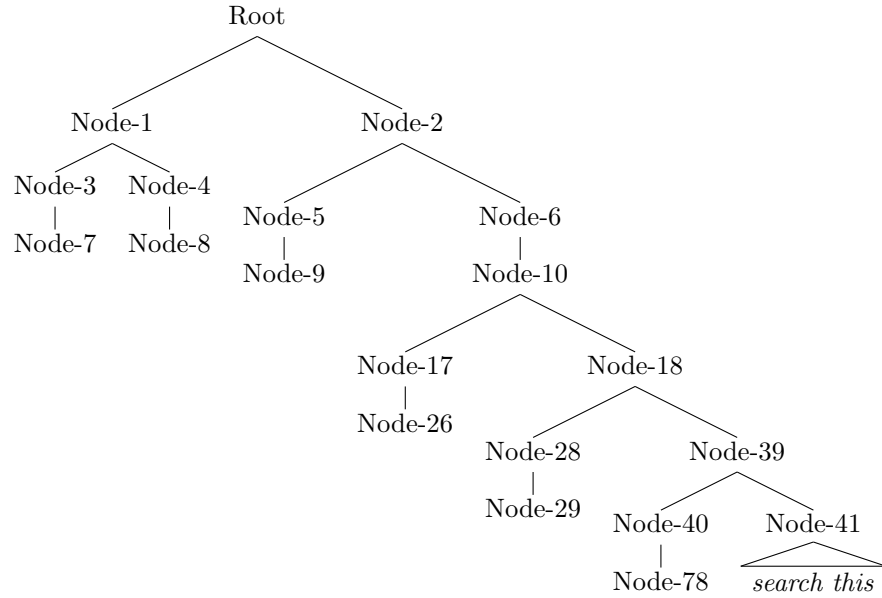
    current = self.root
    while True:
        if search < current.trainNumber: # C * n / 2
            if current.left == None:
                break
            else:
                current = current.left

        elif search > current.trainNumber: # C * n / 2
            if current.right == None:
                break
            else:
                current = current.right

        elif search == current.trainNumber: # C * 1
            break

    else: # C * 1
        break
```

Lets take a simple example in which we search for node-41 from a huge BST.



The peculiarity of this searching technique is that at each levels half of the nodes are eliminated.

This is the recursive equation of the searching algorithm we have used. $T(n) = T(n/2) + 1$

According to 2nd master's theorem:

$$f(n) = \theta(n^{\log_2(1)})$$

$$(f(n) = \theta(1))$$

$$\text{Hence } T(n) = \theta(f(n)) = \theta(\log(n))$$