CSCI 3104 Assignment 5

10:00 - 10:50 Wanshan

Jackson Chen

March 7 2016

1. (a) Assume root is the root node

```
countNodes(root, x.key)

def countNodes(node, val):
  if (node != null):
    if (node.key < val):
      return 1 + countNodes(node.left_node, val)
               + countNodes(node.right_node, val)
    else:
      return countNodes(node.left_node, val)
  else:
    return 0
```

   (b) Assume root is the root node

```
countNode(root, x.key, 0)

def countNode(node, target, total):
  if (node != null):
    if (node.key < target.key):
      total += node.left_node.s
      total += 1 # for the node itself
      return countNode(node.right_node, target, total)
    elsif (node.key == target.key):
      total += node.left_node.s
      return total
    else:
      return countNode(node.left_node, target, total)
  else:
    return 0
```

2. High level steps:

   (a) Sort the intervals by end times.

(b) Find the start time for the last interval

(c) Find the closest end time that is before that start time with binary search

(d) Check if there are any start times between that end and start time with binary search

(e) If not, the interval between that end and start time is a free interval

(f) If so, then shorten the free interval to that end time and the new start

(g) Repeat steps b-f for that end time

Pseudocode:

```
free = []
Iarr = [I1, I2, ..., In]
quickSort(Iarr) // Sorts the ending time in ascending order
tmp = len(Iarr)-1
while (Iarr[tmp][0] > 0):
  freeStart = binarySearch(Iarr, Iarr[tmp][0])
  freeEnd = binarySearch(Iarr, Iarr[freeStart][1])
  free.push([Iarr[freeStart][1], Iarr[freeEnd][0]])
  tmp = freeStart
```

3. Assume the binary search function finds the latest starting time in a time interval

```
colissions = []
Iarr = [I1, I2, ..., In]
quickSort(Iarr) // Sorts starting time in ascending order
for (i in range(len(Iarr))):
  n = binarySearch(Iarr, Iarr[i][1])
  colissions.push(n - i)
return findMax(colissions)
```