INFO1105 Assignment Report

This report is for explaining how to do the assignment called “Calendar” through introducing the data structure, implementations and methods that are created and are used for the assignment. Additionally, this report would also analyze the runtime of the methods and the test cases, which are created by the individual.

The implementation:

1. MyAppointment:

Since the Class “Appointment” only has the three methods created without the logic (getDescription, getLocation, getStartTime). Hence we need to create another class implementing the parent class with constructor and methods with logic overriding parent methods. This “MyAppointment” implements the old Object Appointment and also constructs the constructor passing the three variables ((String) description, (Date) when, (String) Location).

2. Node:

This class is for storing structure the node of the tree. There are three variables, which are data, left and right (left is left child and right is right child). There is a method included in the class called AppointmentEqual, which is used for checking whether the passed in appointment equals the comparison appointment.

The data structure:
In order to choose the suitable data structure for decreasing the time complexity to less than O(n), there are lists of data structure to use for storing. According to the binary tree’s time complexity, it’s average time complexity is O(log(n)). Choosing binary tree to complete the assignment is more suitable than the other data structures such as, ArrayList or LinkList.

Methods:

- getAppointments:

  This method calls another method called “traversal” to go through every node to find the matching one. Although traversal is in linear-time complexity. However, list.clear() method apparently the time complexity is in O(n).

- getNextAppointment(when)

  This method calls the method, which is called “inOrderIterFind”. First it traverses from the most left node, which is the earliest date and the traversal order is from left child, parent, right child and pushes into the stack list. So it will start from comparing the node with the earliest date. Hence the time complexity of the method is O(log(n)).

- getNextAppointment (when, Location)
  This method is the same as above method, but with the location parameter, which needs to consider for compare. So the time complexity is the same as above, which is O(log(n)).

- Add
  The add function traverse down from root and compare to the current node if the node is smaller, it goes to the left child and vice versa. So from root traversing down to the bottom node takes O(log(n)).

- remove
  From root, it finds the matching node. If the compare node is smaller than the current node, it traverses from left child and vice versa. After it finds the matching node, it links its parent to its successor. If the matching node's left child it links the parent node to the right child, and if the right child node is null, it links the left child to the parent node. If both child nodes are not null, then we call the "getSuccessor" method to find the next node of the current node. Since this is traversing down to bottom node, the time complexity of this method is O(log(n)).

Test case:

- testGetonespecificAppointment:
  This test case is to check whether the appointment, which we pick with the specific location is the same as the result we expect.
- testGetAppointmentswithsamelocation:
  This test case is to check whether the appointments can find the matching given location.
- testEmptyGetAppointment:
  This case is for testing whether the result is empty or not if we pick empty appointment.
- testAppointmentAfterRemove:
  This case is to check whether the removed appointments are actually removed or not.
- testGetNextAppointment:
  This test case is to check whether the appointments can find the matching given time.
- testGetNextAppointmentLocation:
  This test case is to check whether the appointments can find the matching given time and location.
- testIllegalArgumentExample:
  This test case is to check whether it would throw the "Illegal Argument Exception" or not when getAppointment method gets nothing.