Programming and Data Structures in Python

Duration: 1 Hour and 15 minutes
Maximum Marks: 20

1. Suppose I provide you with a function `median` which given a list l returns a position i in l containing the median of the values in l. Further assume that it returns this value in time linear in the size of l. Rewrite `Quicksort` to use this function so that its worst case complexity is $O(NlogN)$. [Your algorithm need not be inplace] *(3 marks)*

2. Your task is to maintiain a `set`, i.e. support the operations `insert, delete` and `search`. However, I promise that all the values that we will insert (and hence delete or search) will come from the range $1, 2, \ldots, 10^5$ (the value 10^5 is not critical, it is just a possible reasonable value). Describe the most efficient (in terms of worst case time complexity) implementation of the 3 operations you can provide? Write down a python `class` $MySet$ describing your implementation. *(3 marks)*

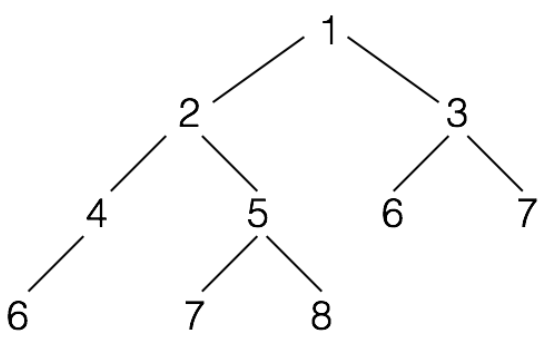
3. For any two sorted (in ascending order) lists x and y let $merge(x, y)$ denote the sorted listed obtained by merging them together. Consider the following code where `L1` and `L2` are lists sorted in ascending orders.

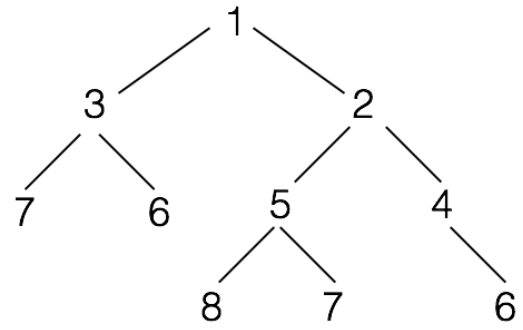
```
l1 = L1+[]
l2 = L2+[]
ans = []
while (l1 != []) and (l2 != []):
  if l1[0] <= l2[0]:
    ans.append(l1[0])
    l1.pop(0)
  else:
    ans.append(l2[0])
    l2.pop(0)
ans = ans + l1 + l2
```

   Demonstrate a loop invariant that enables you to argue that at the end `ans` $= merge($`L1`$,$`L2`$)$ *(5 marks)*

4. Assume you have an implementation of a `heap` as a class. Add another method `add(i,v)` which adds the value v (which may be positive or negative) to the value in the ith node in the heap (and then ensures that the resulting list is still a heap). If the heap has fewer than $i + 1$ elements and hence has no node i then your method should do nothing. What is the complexity of your function? *(5 marks)*

5. Write a python function `nestedFlip` which takes a binary tree `BTree` as defined in class and flips the right and left subtrees at every node. Here is an example. The tree on the right is obtained by applying `nestedFlip` to the tree on the left.





(5 marks)
