Report for assignment 9

Manne Naga Nithin (14EE10026)

March 21, 2016

We take a variable which keeps track of the time points for each node. Whenever a loop is run, the time for the loop is taken from the nodes which are involved. Then the times are examined.

Amortised Cost of Operations

1. **insertHeap**
   We allot 2 time units. When the extractMin operation happens, the combining operation happens using one time operation from one of the combining elements.
   Amortised Cost : $O(1)$

2. **decreasekeyHeap**
   We allot 3 time units. One time unit is given to the parent when we mark it, the other to itself when it goes back to the root list. When the parent gets two time units, or double marked, it uses one to go back to the list and another to its parent. When the extractMin operation happens, the combining operation happens using one time operation from one of the combining elements.
   Amortised Cost : $O(1)$

3. **makeHeap**
   We just allocate memory to the pointer.
   Time Complexity : $O(1)$

4. **unionHeap**
   We just add one list tho the end of another, also updating the minimum to whichever minimum is smaller.
   Time Complexity : $O(1)$

5. **extractMinHeap**
   Firstly, we remove the minimum element from the list and adds its children to the list, which takes a $log(n)$ time. Then we combine the elements into Fibonacci trees, which is done by using time gotten from the insert operation.
   Time Complexity : $O(log(n))$