```python
class ChordNode:

  def __succNode(self, key):
    if (key <= self.nodeSet[0] or
        key > self.nodeSet[len(self.nodeSet)-1]): # key is in segment for which
      return self.nodeSet[0]                      # this node is responsible
    for i in range(1,len(self.nodeSet)):
      if (key <= self.nodeSet[i]):                # key is in segment for which
        return self.nodeSet[i]                    # node (i+1) may be responsible

  def __finger(self, i):
    return self.__succNode((self.nodeID + pow(2,i-1)) % self.MAXPROC) # succ(p+2^(i-1))

  def __recomputeFingerTable(self):
    self.FT[0]  = self.nodeSet[(self.nodeInd - 1)%len(self.nodeSet)] # Predecessor
    self.FT[1:] = [self.__finger(i) for i in range(1,self.nBits+1)]  # Successors
    self.FT.append(self.nodeID)                                      # This node

  def __localSuccNode(self, key):
    if self.__inbetween(key, self.FT[0]+1, self.nodeID+1):  # key in (pred,self]
      return self.nodeID                                    # this node is responsible
    elif self.__inbetween(key, self.nodeID+1, self.FT[1]):  # key in (self,FT[1]]
      return self.FT[1]                                     # successor responsible
    for i in range(1, self.nBits+2):                        # go through rest of FT
      if self.__inbetween(key, self.FT[i], self.FT[(i+1)]): # key in [FT[i],FT[i+1])
        return self.FT[i]                                   # FT[i] is responsible
```