CPE111: Programming with Data Structures

Week6: Sorting Algorithm II

Practice I) Advanced Sorting Algorithm
Mission: Compare these sorting algorithms in **List1** – **List3** with the python's built-in sort function, when n = 100, 200, 300, 400, 500,…, 1000, 5000 and then plot a graph of time operation by compare in 3 cases: best case, average case (random) and worst case.

List 1: merge_sort.py

```
1 def merge(left, right, seq):
2   i = j = 0
3   while i + j < len(seq):
4     if j == len(right) or (i < len(left) and left[i] < right[j]):
5       seq[i+j] = left[i] # copy ith element of left as next item of seq
6       i += 1
7     else:
8       seq[i+j] = right[j] # copy jth element of right as next item of seq
9       j += 1
10
11 def merge_sort(seq):
12   n = len(seq)
13   if n < 2:
14     return # list is already sorted
15   # divide
16   mid = n // 2
17   left = seq[0:mid] # copy of first half
18   right = seq[mid:n] # copy of second half
19   # conquer (with recursion)
20   merge_sort(left) # sort copy of first half
21   merge_sort(right) # sort copy of second half
22   # merge results
23   seq = merge(left, right, seq)
24   return seq
```

List2: quick_sort.py

```
1 def quick_sort(seq):
2   if len(seq) < 2 : return seq
3   mid = len(seq)//2
4   pi = seq[mid]
5   seq = seq[:mid] + seq[mid+1:]
6   lo = [x for x in seq if x <= pi]
7   hi = [x for x in seq if x > pi]
8   return quick_sort(lo) + [pi] + quick_sort(hi)
```

List3: radix_sort.py

```
1  from linklistedQueue import LQueue
2  from array import Array
3  def radixSort( seq, numDigits ):
4    # Create an array of queues to represent the bins.
5     binArray = Array(10)
6     for k in range( 10 ):
7        binArray[k] = LQueue()
8    # The value of the current column.
9    column = 1
10   # Iterate over the number of digits in the largest value.
11   for d in range( numDigits ):
12      # Distribute the keys across the 10 bins.
13      for key in seq :
14         digit = (key // column) % 10
15         binArray[digit].enqueue( key )
16      # Gather the keys from the bins and place them back in intList.
17      i = 0
18      for bin in range(len(binArray)) :
19         while not binArray[bin].isEmpty() :
20            seq[i] = binArray[bin].dequeue()
21            i += 1
22      # Advance to the next column value.
23      column *= 10
24   return seq
```

Reference

Rance D. Necaise. *Data Structures and algorithms using python*. *Chapter2*. John Wiley&Sons,Inc., 2011

Michael T.Goodrich, Roberto Tamassia, Michael H. Goodwasser. *Data Structures and Algorithms in python*. *Chapter5*. John Wiley&Sons,Inc. 2013