LAB#25

Example#1: Write a code for bucket sort algorithm to sort an unsorted array. The elements in the array should be in between 0 to 9.

Solution:

bucketnew.py > ...

```
1   from array import array
2   def insertion_sort(arr):
3       for i in range(1, len(arr)):
4           key = arr[i]
5           j = i - 1
6           while j >= 0 and key < arr[j]:
7               arr[j + 1] = arr[j]
8               j -= 1
9           arr[j + 1] = key
10
11  def bucket_sort(arr):
12      n = len(arr)
13      print(n)
14      if n == 0:
15          return
16
17      buckets = [[] for j in range(n)]
18
19      # Put array elements in different buckets
20      for i in range(n):
21          index = int(arr[i] * n)
22          buckets[index].append(arr[i])
23
24      for i in range(n):
25          insertion_sort(buckets[i])
26
```

```
    # Concatenate all buckets into arr[]
    k = 0
    for i in range(n):
        for j in range(len(buckets[i])):
            arr[k] = buckets[i][j]
            k += 1
a1 = array('d', [0.38, 0.90, 0.47, 0.69, 0.52, 0.88, 0.71, 0.18, 0.20])
bucket_sort(a1)

for x in a1:
    print(x, end=' ')

```

Explanation of the code:

a1=array('d',[0.38,0.90,0.47,0.69,0.52,0.88,0.71,0.18])

bucket_sort(a1)

def bucket_sort(a1):

n=len(a1)

n=9

if n==0:

return

buckets=[[ ] for j in range(n)] #n=9

```
[[], [], [], [], [], [], [], [], []]
```

for i in range(9): #i=0

index=int(arr[i]*n)

index=int(arr[0]*9)

index=int(0.38*9)

index=int(3.42)

index=3

buckets[index].append(arr[0])

buckets[3].append(0.38)

0 1 2 3 4 5 6 7 8

[[],[],[],[0.38],[],[],[],[],[]]

for i in range(9): #i=1

index=int(arr[i]*n)

index=int(arr[1]*9)

index=int(0.9*9)

index=int(8.1)

index=8

buckets[index].append(arr[1])

buckets[8].append(0.9)

0 1 2 3 4 5 6 7 8

[[],[],[],[0.38],[],[],[],[],[0.9]]

for i in range(9): #i=2

index=int(arr[i]*n)

index=int(arr[2]*9)

index=int(0.47*9)

index=int(4.23)

index=4

buckets[index].append(arr[2])

buckets[4].append(0.47)

0 1 2 3 4 5 6 7 8

[[],[],[],[0.38],[0.47],[],[],[],[0.9]]

for i in range(9): #i=3

index=int(arr[i]*n)

index=int(arr[3]*9)

index=int(0.69*9)

index=int(6.21)

index=6

buckets[index].append(arr[3])

buckets[6].append(0.69)

0 1 2 3 4 5 6 7 8

[[],[],[],[0.38],[0.47],[],[0.69],[],[0.9]]

for i in range(9): #i=4

index=int(arr[i]*n)

index=int(arr[4]*9)

index=int(0.52*9)

index=int(4.68)

index=4

0 1 2 3 4[0],4[1] 5 6 7 8

[[],[],[],[0.38],[0.47,0.52],[],[0.69],[],[0.9]]

buckets[index].append(arr[4])

buckets[4].append(0.52)

for i in range(9): #i=5

index=int(arr[i]*n)

index=int(arr[5]*9)

index=int(0.88*9)

index=int(7.92)

index=7

0 1 2 3 4[0],4[1] 5 6 7 8

[[],[],[],[0.38],[0.47,0.52],[],[0.69],[0.88],[0.9]]

buckets[index].append(arr[5])

buckets[7].append(0.88)

for i in range(9): #i=6

index=int(arr[i]*n)

index=int(arr[6]*9)

index=int(0.71*9)

index=int(6.39)

index=6

0 1 2 3 4[0],4[1] 5 6[1] 6[2] 7 8

[[],[],[],[0.38],[0.47,0.52],[],[0.69,0.71],[0.88],[0.9]]

buckets[index].append(arr[6])

```
            buckets[6].append(0.71)

for i in range(9):                          #i=7
            index=int(arr[i]*n)
            index=int(arr[7]*9)
            index=int(0.18*9)
            index=int(1.62)
            index=1
            buckets[index].append(arr[7])
            buckets[1].append(0.18)

for i in range(9):                          #i=8
            index=int(arr[i]*n)
            index=int(arr[8]*9)
            index=int(0.2*9)
            index=int(1.8)
            index=1
            buckets[index].append(arr[8])
            buckets[1].append(0.2)
```

0 1 2 3 4[0], 4[1] 5 6[1] 6[2] 7 8

[[],[0.18],[],[0.38],[0.47,0.52],[],[0.69,0.71],[0.88],[0.9]]

0 1[0],1[1] 2 3 4[0], 4[1] 5 6[1] 6[2] 7 8

[[],[0.18,0.2],[],[0.38],[0.47,0.52],[],[0.69,0.71],[0.88],[0.9]]

continue from line 24 insertion_sort that we have already done in our previous classes.

Class Assignment

Q.1: Perform the remaining steps in example#1(LAB#25).

Q.2: Create an array of size 10. Instead of using elements from 0 to 1, use elements in the array greater than 1. Then apply the code from example #1 (LAB #25). If you encounter any errors, discuss the error and identify the reason behind it.