Problem 1

a) A

	1	2	3	4	5	6	7	
A	9	1	6	7	6	2	1	

	0	1	2	3	4	5	6	7	8	9
C	0	0	0	0	0	0	0	0	0	0

— c initialized to zero

At the end of the for loop $C[i]$ contains no. of elements equal to i.

	0	1	2	3	4	5	6	7	8	9
C	0	2	1	0	0	0	2	1	0	1

↓

A

	0	1	2	3	4	5	6	7	8	9
C	0	2	3	3	3	3	5	6	6	7

↓

	1	2	3	4	5	6	7			
B	1	1	2	6	6	7	9			

	0	1	2	3	4	5	6	7	8	9
C	0	0	0	0	0	0	0	0	0	0

b)

```
        A                    B
   ┌──────┐          ┌──────┐
   │ 0.9  │      0   │  /   │
   ├──────┤          ├──────┤
   │ 0.1  │      1   │  ────→ ┌─────┐    ┌─────┐
   ├──────┤          ├──────┤  │ 0.1 │──→ │ 0.1/│
   │ 0.6  │      2   │  ────→ └─────┘    └─────┘
   ├──────┤          ├──────┤  ┌─────┐
   │ 0.7  │      3   │  /   │  │ 0.2/│
   ├──────┤          ├──────┤  └─────┘
   │ 0.6  │      4   │  /   │
   ├──────┤          ├──────┤
   │ 0.2  │      5   │  /   │
   ├──────┤          ├──────┤
   │ 0.1  │      6   │  ────→ ┌─────┐    ┌─────┐
   └──────┘          ├──────┤  │ 0.6 │──→ │ 0.6/│
                  7   │  ────→ └─────┘    └─────┘
                     ├──────┤  ┌─────┐
                  8   │  /   │  │ 0.7/│
                     ├──────┤  └─────┘
                  9   │  ────→ ┌─────┐
                     └──────┘  │ 0.9/│
                               └─────┘
```
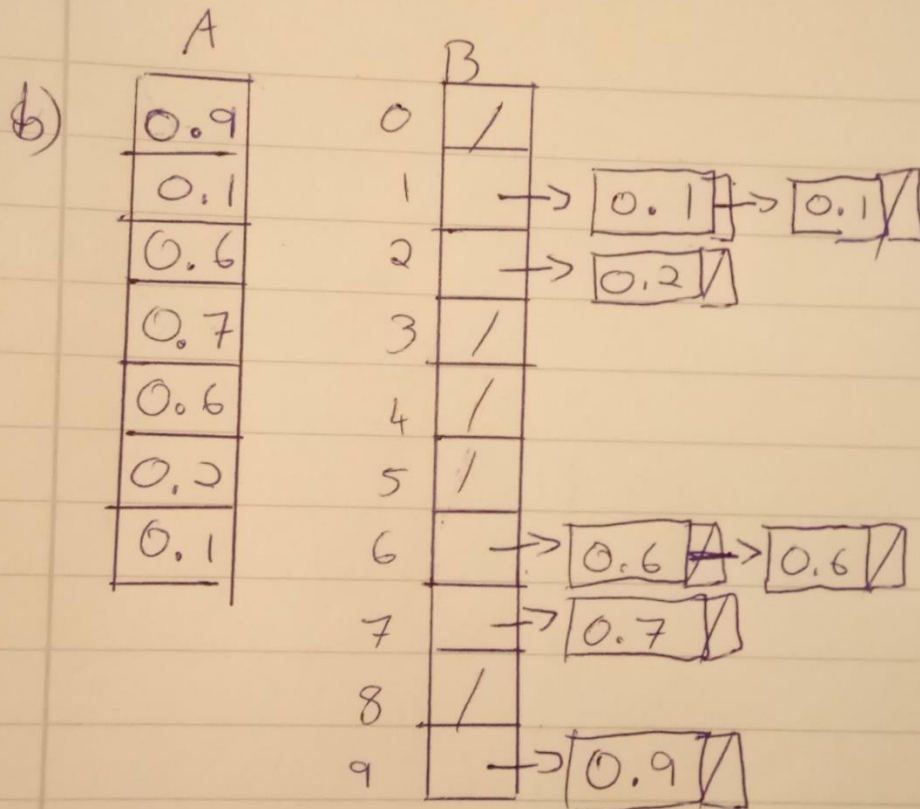
Output = {0.1, 0.1, 0.2, 0.6, 0.6, 0.7, 0.9}

c) Algorithm is the same as counting sort except for line 7, 8 we have:

for i = a to b

$$c[b] = c[b] - c[a-1] \quad // c[-1] = 0$$

e) If the algorithm puts all the elements in the same bucket, then the linked list needs to be traversed every time an element is added.

$$\therefore \quad T(n) = \sum_{i=0}^{n} A[i] = (n^2 + n) \cdot \frac{1}{2} = \underline{O(n^2)}$$

f) Pass the points x and y through a function to determine which bucket they will go into:

```
chooseBucket( x , y , n ) // coordinates x & y
    distance = (x * x) + (y * y)
    return floor (distance * n) // n is the number of points
```

Problem 2

a) Please find attached C++ file.
b) The storage of the implementation is $\Theta(n)$ for the original array plus $\Theta(10)$ for array C plus $\Theta(n)$ for the output array. Therefore, the asymptotic storage space is $S(n) = 2n + 10 \rightarrow \Theta(n)$.

$T(n) = \Theta(n \log n)$.

P.S. I have not attached the medical excuse because I didn't download the scan within the time period of 3 days.