Homework 3

Radix Sort

1. If Radix Sort is applied to the array of numbers [4567, 3210, 2345, 4321, 5678], show how these numbers will get rearranged after each of the four passes of the algorithm.

   a. Step 1: 0's position

0	1	2	3	4	5	6	7	8	9
3210	4321				2345		4567	5678	

   3210, 4321, 2345, 4567, 5678

   b. Step 2: 10s position

0	1	2	3	4	5	6	7	8	9
	3210	4321		2345		4567	5678		

   3210, 4321, 2345, 4567, 5678

   c. Step 3: 100s position

0	1	2	3	4	5	6	7	8	9
		3210	2345/4321		4567	5678			

   3210, 4321, 2345, 4567, 5678

   d. Step 4: 1000s position

0	1	2	3	4	5	6	7	8	9
		2345	3210	4567/4321	5678				

   2345, 3210, 4321, 4567, 5678

Divide and Conquer & Algorithm Design

2. Consider the following algorithm:

```
Algorithm Mystery(A: Array [i..j] of integer)
i & j are array starting and ending indexes
begin
if i=j then
   return A[i]
else
   k=i+floor((j-i)/2)
   temp1= Mystery(A[i..k])
   temp2= Mystery(A[(k+1)..j]
   if temp1<temp2 then
      return temp1 else return temp2
end
```

What does the recursive algorithm above compute?

Finds out the largest element present in the array and returns it after it terminates.

Adjacency Matrix

3. Represent the follow graph using its adjacency matrix. The number in the circle indicates the corresponding index of the node.
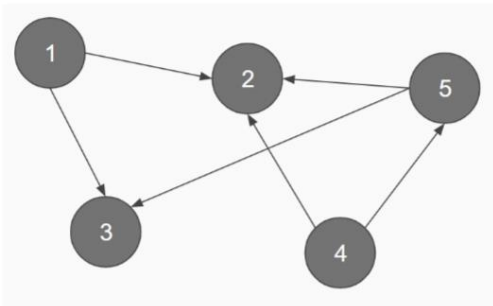
Adjacency matrix

0	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	1	0	0	1
5	0	1	1	0	0

Prim's Algorithm / MST

4. Rule out the MST of the following graph using Prim's algorithm. The number in the circle indicates the corresponding index of the node, while the numbers on the edges are their weights. **Use node 0 as the starting node. You must show your work.**

Step 1:

S = {0}
v/s = {1,2,3,4,5,6}
A = { }
lightest edge = {01}

Step 2:

S = {0, 1}
v/s = {2,3,4,5,6}
A = {01}
lightest = {13}

Step 3:

S = {0, 1, 3}
v/s = {2,4,5,6}
A = {01, 13}
lightest = {34}

Step 4:

S = {0,1,3,4}
v/s = {2,5,6}
A = {01, 13, 34}
lightest = {24}

Step 5:

S = {0,1,3,2,4}
vs = {5,6}
A = {01, 13, 34, 42}
lightest = {21, 26}

Step 6:

S = {0,1,2,3,4,6}
v/s = {5}
A = {01, 13, 34, 42, 26}
No lightest