**Lab 5:**

**Question 1:** Among n identical-looking coins, one is fake. The fake coin is known to be lighter than the genuine one. With a balance scale, we can compare any two sets of coins. Design and efficient (with minimum number of weighing) divide-into-5 algorithm for detecting the fake coin. Set up a recurrence relation for the number of comparisons or weighing for this algorithm and solve it by backward substitution method for n = 5k

**Question 2**: The following array is being sorted using Hoare’s Quicksort method. We know that Quicksort completes sorting of the left partition by calling itself recursively, and then handles the second partition. Show the trace of the first 5 calls of the Quicksort method, and show the result after each call.

14, 2, 5, 7, 45, 77, 1, 55, 47, 51, 63, 19, 20, 54, 28, 35

1st call: QuickSort(A[0…15]) ->

2nd call:

3rd call:

4th call:

5th call:

**Question 3:** What is the worst case input for quicksort algorithm? Which type of input (decreasing order, ascending order, all elements are equal or random array)? Why? Explain this case with one example.

**Question 4:** What is the best case input for quicksort algorithm? Which type of input (decreasing order, ascending order, all elements are equal or random array)? Why? Explain this case with one example.

**Question 5:** Let's assume we have n sized array. We select the first item in the list as the pivot.

* For a strictly descending order array, which partition method uses less swap (Hoare or Lomuto). Why? Explain your answer with one example.
* For a strictly ascending order array, which partition method uses less swap (Hoare or Lomuto). Why? Explain your answer with one example.