1. Modify version 2 of counting sort algorithm from the course website by changing the last for loop to go from 1 to $n$ instead of $n$ to 1. Apply this modified counting sort algorithm to the following array of numbers, $A = [6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2]$ to sort it. Is this version of the algorithm, stable? Show the various arrays, and indicate the reason for stability or otherwise.

2. Is bubble sort stable? Why or why not? Use the version we did in class.

3. Explain whether or not heapsort can be used as an auxiliary sorting routine instead of counting sort in radix sort algorithm. Please note that heapsort works in-place. Give a clear reasoning for your answer.

4. Show how to sort $n$ integers in the range 0 to $n^5 - 1$ in the most optimal runtime.

5. We are given $n$ points in the unit circle, $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for $i = 1, 2, \ldots, n$. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. Design an algorithm with an average-case running time of $\theta(n)$ to sort the $n$ points by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin.

6. You are given $n$ integers in the range 0 to $k$, describe an algorithm that preprocesses its input in linear time, and then answers a query about how many of the $n$ integers fall in the range $[a \ldots b]$ in constant time.