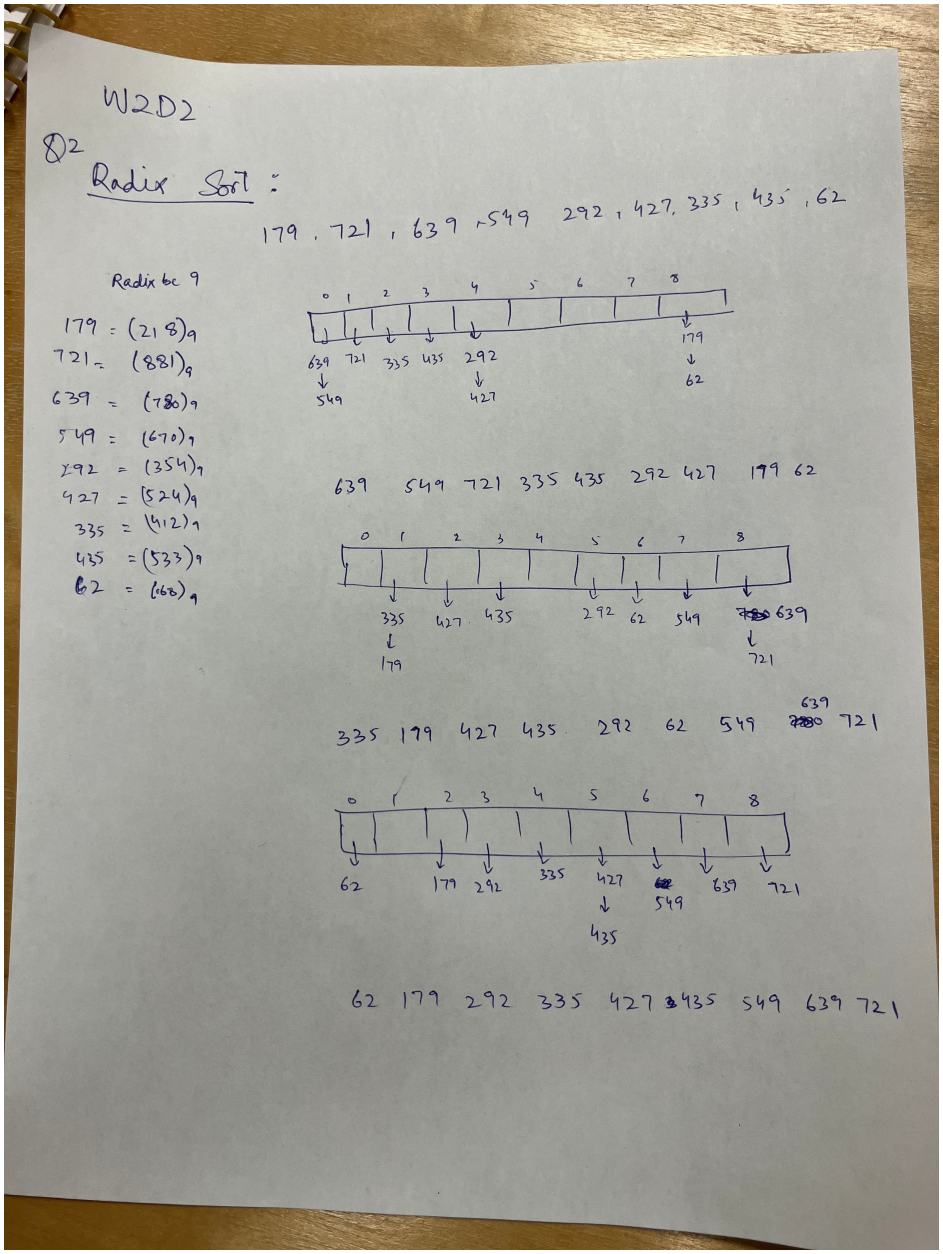
Wednesday, 6 December 2023

Lab7

Question 1. Practice Radix Sort

Repeat what I have done in Slides 24 and 25. Let Radix be 9. Array to be sorted is {179, 721,
639, 549, 292, 427, 335, 435, 62}.



Question 2. Experimenting with lower bound

Devise an algorithm to sort 4 elements using exactly 5 comparisons in the worst case. Does this violate the theoretical lower bound? Justify your answer.

Theoretical lower bound of the decision tree is log(4!) = log(24), therefore the log(16) < lower bound < log(32) => 4 < lower bound < 5. There are 6 comparisons in the algorithm described DecisionTreeAlgorithmFourElements so lower bound has not been violated.

Question 3. Exploring new ideas: Forward and backward sorted array (FBS array)

Definition
An array is said to be FBS array if it satisfies the following three conditions.
(1) Elements in the even locations are sorted in the ascending order.
(2) Elements in the odd locations are sorted in the descending order.
(3) Every element in the even locations are <= every element in the odd locations.
Example {7, 20, 10, 19, 10, 17, 14, 15, 15}
What is the asymptotic running time of your algorithm? What is the fastest possible asymptotic
running time for such an algorithm? Prove your answer.

To analyze the asymptotic running time of an algorithm for checking whether an array is a Forward and Backward Sorted (FBS) array, we need to consider the steps involved in the algorithm.

One way to check if an array is an FBS array is to iterate through the array and compare each element at an even index with the next even index and each element at an odd index with the next odd index, making sure that the conditions (1), (2), and (3) are satisfied.

Let's denote the length of the array as n. In the worst case, we need to compare every pair of adjacent elements to check the FBS conditions. Therefore, the time complexity of this algorithm is $O(n)$.

The fastest possible asymptotic running time for such an algorithm is also $O(n)$ because we need to inspect every pair of adjacent elements in the array at least once to verify the FBS conditions. In other words, we cannot do better than linear time complexity for this specific problem.

Proving $O(n)$ as the fastest possible asymptotic running time can be done by showing that any algorithm solving this problem must examine at least a constant fraction of the input elements. Since every element in the even locations must be compared with its adjacent even location and every element in the odd locations must be compared with its adjacent odd location, the algorithm has to look at every element at least once. Therefore, the lower bound for this problem is $O(n)$, and $O(n)$ is both a lower and upper bound, making it the fastest possible asymptotic running time.