Algorithms Homework 4 Code Report

Ryan Rau [redacted]

February 21 2020

1 Problem

Show how to sort n integers in range 0 to $n^3 - 1$ in O(n) time

2 Solution

2.1 Algorithm

Given that I could find a d in radix sort to be a constant I knew that I could using counting sort to achieve a run time of O(n). With that I started with the following pseudo code and began modifying it.

```
RADIX-SORT(A, d)
    for i to d
        use a stable sort to sort array A on digit i


COUNTING-SORT(A, B, k)
    let C[0..k] be new array
    for i = 0 to k
        C[i] = 0
    for j = 1 to A.length
        C[A[j]] = C[A[j]] + 1
    for i = 1 to k
        C[i] = C[i] + C[i - 1]
    for j = A.length downto 1
        B[C[A[j]]] = A[j]
        C[A[j]] = C[A[j]] - 1
```

Since all values of the array will be between 0 and $n^3 - 1$ I though about ways to determine how many digits the number would have given any n. After messing with bases, I concluded that I could create a three digit number if I put the number into base n.

$$log_n(n^3) = 3$$

With that I modified the counting sort algorithm based off one I found on geeksforgeeks.com (see Resources) and sorted based off the modulo of the value divided by n^i where i is the digit/place that is being sorted (0, 1, 2)

2.2 Time Complexity

Since I was able to reduce d to be a constant value of 3 and that the run time of counting sort is $O(n)$, I was able to conclude the that the time complexity of my algorithm is $O(n)$

$$O(n) + O(n) + O(n) = O(3n) = O(n)$$

2.3 Testing Results

To test my algorithm, I feed it multiple values for n and ensured that sorted it properly. For example:

For n = 5 the results of my program/algorithms was:

```
n = 5
Unsorted Array:
[31, 65, 0, 95, 90]
Sorting...
Sorting pass 0
[65, 0, 95, 90, 31]
Sorting pass 1
[0, 31, 65, 90, 95]
Sorting pass 2
[0, 31, 65, 90, 95]
Sorted Array:
[0, 31, 65, 90, 95]
```

3 Resources

While working on this project, I found the following site to be helpful:

https://www.geeksforgeeks.org/radix-sort/