Name: Georges Hatem

CS590 Homework 9: Divide-and-Conquer Creativity Exercises

Due Date: April 3, 2022

Problem 11.6.9:

The correctness of the Algorithm (Shown by Mathematical Induction)

Induction: The Stooge Sort Algorithm works for one-element and two-element arrays

Consider that it works for all arrays smaller than A[i.....j] and let us show that it also works for A[i.....j].

After execution of Stooge Sort (A, i, j-m) for sorting the first part, A[i..(j-m)] is sorted, which means that every element of A[(i+m)..(j-m)] is no smaller than every element of A[i..(i+m-1)]; It can be written as A[(i+m)...(j-m)] >= A[i...(i+m-1)].

Thus, A[(i+m)...j] has at least length (A[(i+m)...(j-m)])=j-i-2m+1 elements each of which is no smaller than each element of A[i...(i+m-1)].

After execution of StoogeSort(A, i+m, j) for sorting the last part, A[(i+m)…j] is sorted, which implies that

A[(j-m+1)…j] is sorted …. (1)
A[(j-m+1)…j] >= A[(i+m)…(j-m)] …. (2)

Since A[(i+m)…j] has at least (j-i-2m+1) elements no smaller than each element of

A[i…(i+m-1)] and length (A[(j-m+1)…j]) <= j-i-2m+1, it complete that

A[(j-m+1)…j] >= A[i…(i+m-1)] … (3)

Combine the equations (2) and (3), it can write as:

A[(j-m+1)…j] >= A[i…(j-m)]

After execution of StoogeSort(A,i,j-m) for sorting the 1st part again, combine the equations 1 and 4 and the whole Array A[i,j] is sorted.

<u>Characterize the running time, T(n) for Stooge-Sort using as recurrence equation:</u>

- Time taken to execute for lines 1 to 5 is constant time
- In Lines 6 to 8, the function calls itself recursively. Each time, size of the array is 2/3 of the original array size.

Therefore, the recurrence equation of Stooge-Sort is:

$$\boldsymbol{T(n) = 3T\left(\frac{2n}{3}\right) + O(1)}$$

Determining an Asymptotic Bound for T(n) by means of the Master's Method:

Let's characterize the recurrence equation below using the master theorem:

$$T(n) = \left(3 * T\left(\frac{2n}{3}\right)\right) + O(1)$$

From the recursion equation general form given in Section 11.2 for Master's Theorem, we have the following:

$$T(n) = \left(a * T\left(\frac{n}{b}\right)\right) + f(n) \quad for\; n \geq d$$

From the above, we can deduce the following:

$$f(n) = O(1), a = 3, and\; b = \frac{3}{2}$$

Let's find what the special function gets us:

$$n^{log_b(a)} = n^{log_{1.5}(3)} \cong n^{2.7095}$$

From the analysis, we can see that:

$$n^{2.7095} > O(1)$$

Which means that:

$$n^{log_b(a)} > f(n)$$

This characterizes case 1 of the master's theorem defined in Section 11.2. This means that T(n) is $\boldsymbol{\Theta(n^{log_b(a)})}$ based on case 1 of the master's theorem defined in Section 11.2.

So,

$$\boldsymbol{T(n)\ is\ \Theta(n^{log_{1.5}(3)})}$$

Which means that

$$\boldsymbol{T(n)\ is\ \Theta(n^{2.7095})}$$