Monday, April 11, 2022

Homework 1

Nils Streedain

1. Problem 1: Mergesort3

   i. Write pseudo-code for Mergesort3 and Merge3

```
void mergesort3 (vector v) {
    If size is 2 and v[0] > v[1], swap them

    If size > 2 {
        mergesort3(left)
        mergesort3(middle)
        mergesort3(right)
        merge3(left, middle, right, v)
    }
}



void merge3 (vector left, vector middle, vector right, vector output) {
    int l, m, r, i = 0
    Loop over each index of output {
        If left is not empty & middle & right is empty, or not empty but less than left
            output[I] = left[l++]
        Else If middle is not empty & left & right is empty, or not empty but less than middle
            output[I] = middle[m++]
        Else move right
            output[I] = right[r++]
        i++
    }
}
```

   ii. Let T(n) denote the running time of Mergesort3 on an array of size n. Write a recurrence relation for T(n).

   - $T(n) = 3T(n/3) + O(n)$

   iii. Solve the recurrence relation to obtain the asymptotic running time.

   - $T(n) = O(n * \log(n))$

2. Problem 2: Stoogesort

   i. Write pseudocode for Stooge sort.

```
void stoogesort (vector v, int min, int max) {
    If the value at the start of the list is larger than the value at the end, swap them.

    If (max - min + 1) > 2 {
        stoogesort(leftTwoThirds)
        stoogesort(rightTwoThirds)
        stoogesort(leftTwoThirds)
    }
}
```

   ii. Let T(n) denote the running time of Stooge sort on an array of size n. Write a recurrence relation for T(n).

   - $T(n) = 3T(2n/3) + O(1)$

   iii. Solve the recurrence to determine the asymptotic running time.

   - $T(n) = O(n^{\log_{(3/2)}3})$

5. Problem 5: Data analysis

   i. Collect running times

   - Merge Sort 3

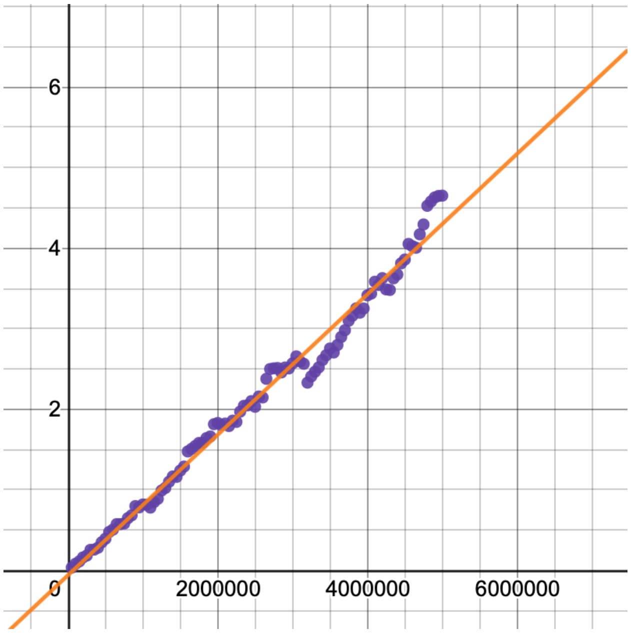
n	**500000**	**1000000**	**1500000**	**2000000**	**2500000**	**3000000**	**3500000**	**4000000**	**4500000**	**5000000**
Time (s)	0.42239	0.87899	1.29332	1.91495	2.13205	2.69311	2.87915	3.56843	4.04232	4.86808

   - Stooge Sort

n	**500**	**1000**	**1500**	**2000**	**2500**	**3000**	**3500**	**4000**	**4500**	**5000**
Time (s)	0.07569	0.22408	0.64958	1.93858	5.80868	5.81331	5.82149	17.5425	17.4568	17.4778

ii. Plot Data and fit a curve

- Merge Sort 3

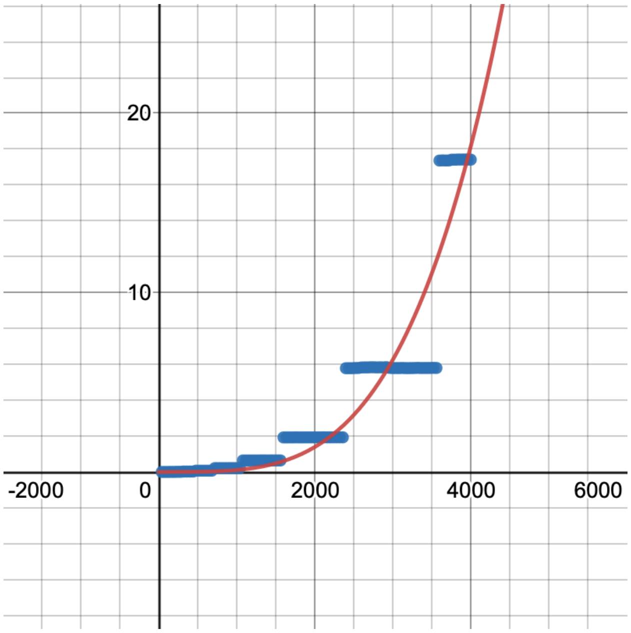


This data set most closely matches a linear O(n). This is expected, considering this matches the worst case for merge sort (O(n)) because it is a list of randomly arranged elements.

Best fit: y ≈ 8.72(10^{-7}) * x - .06

Expected: y ≈ mx + b

- Stooge Sort



This data set most closely matches an exponential O($n^{3.71}$). This follows the expected of $n^{\log_{(3/2)}3}$ which is about $n^{2.71}$.

Best fit: y ≈ 8(10^{-13}) * $x^{3.71}$

Expected: y ≈ $mx^{2.71}$