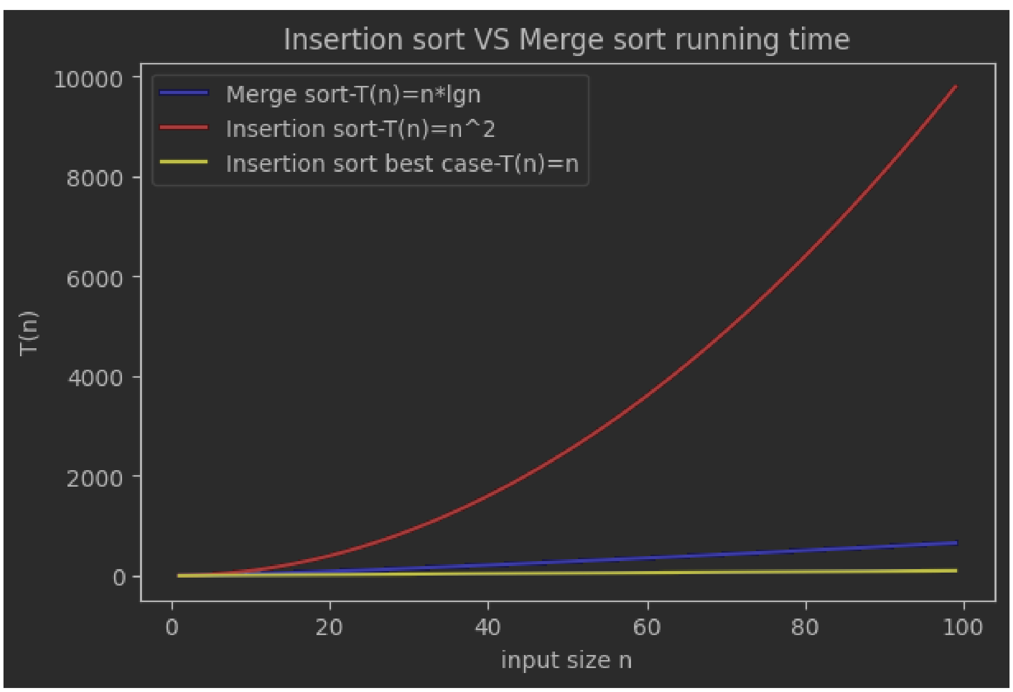
**4DS3: Data Structures and Algorithms Spring/Summer 2023**
**Challenge Project 1**

**1. Given that Insertion sort is worst case θ ( $n^2$ ) and Merge sort is worst case θ(nlg n):**
**a. Under what conditions would Insertion sort outperform Merge sort? You can give a concrete example of this and show/plot the growth of the function as a graph.**

The condition that insertion sort outperforms Merge sort is when insertion sort is best case or the input size is small.

| | Insertion Sort | Merge Sort |
|---|---|---|
| Best Case | $\theta(n)$ | θ(nlg n) |
| Average Case | $\theta(n^2)$ | θ(nlg n) |
| Worst Case | $\theta(n^2)$ | θ(nlg n) |




As the graph shows, when insertion sort is running at the best case, it is quicker than Merge sort.

**b. Describe scenario(s) in which Insertion sort may be preferable to Merge sort as it is implemented in the pseudocode we discussed in class?**

Scenario Insertion sort is preferable to Merge sort:
When the array is nearly sorted and the input size is small, insertion sort is using less run time.

| pseudocode | Cost | # of time of each line execution | Calculation |
|---|---|---|---|
| For j=2 to n | C1 | n | T(n)=C1*n+C2*(n-1)+ C4*(n-1)+C5*(n-1)+C8 *(n-1)=(C1+C2+C4+C 5+C8)*n+(-C2-C4-C5- C8)<br>=> T(n)=a*n+b |
| key=A[j] | C2 | n-1 | |
| //Insert A[j] into the sorted sequence A[1…j] | C3 | n-1 | |
| i=j-1 | C4 | n-1 | |
| While i>0 and A[i]>key | C5 | n-1 | |
| A[i+1]=A[i] | C6 | 0 | |
| i=i-1 | C7 | 0 | |
| A[i+1]=key | C8 | n-1 | |

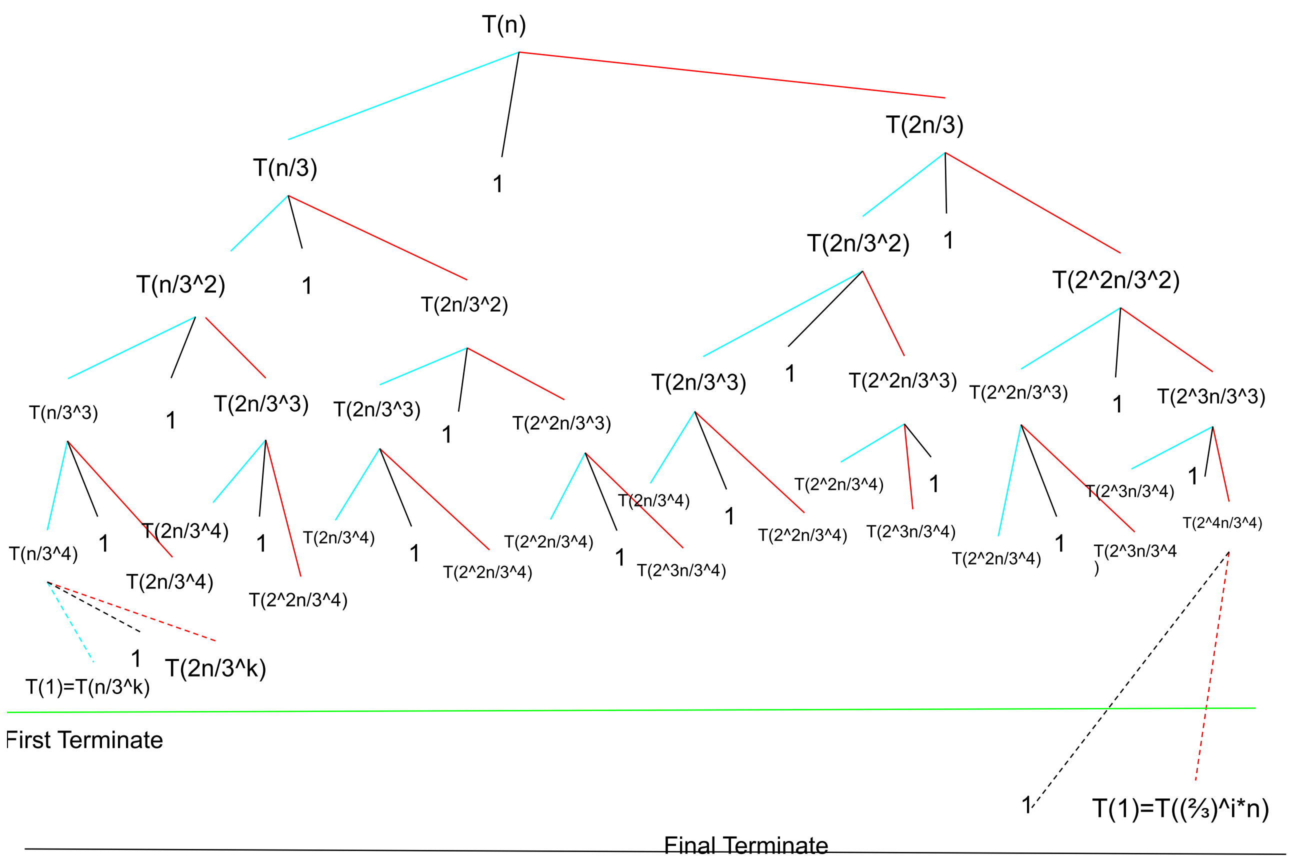
T(n) of Merge sort=O(nlg n)> T(n) of  insertion sort's best case.

**2. Other than speed, what other factors may determine "efficiency" of an algorithm?**

There are some other factors that may determine the "efficiency" of an algorithm:
1. Input size
2. Memory usage (auxiliary space). Some of the algorithms are not in-place algorithms, they might take up some extra space to run the algorithm.
3. The sorted arrays can impact the efficiency of an algorithm.
4. Power consumption (Rosenburg, n.d.)

**3. Find the order of growth of the following recurrence relations using the Substitution Method.**
**a.** $T(n) = T(n/3) + T(2n/3) + 1$

Step1: Guess a solution for T(n)




At lower bound:
Pretend end at first termination:

At some point, $\frac{n}{3^k} = 1, n = 3^k, k = log_3 n$

K levels, and 1 calculator per level: T(n)=k*1=$log_3 n$

T(n)=$\Omega(log_3 n)$

At upper bound:
i levels, and 1 calculator per level: T(n)=i*1

At some point$(\frac{2}{3})^i n = 1$⇒$(\frac{2}{3})^i = n$ ⇒ n= $(\frac{3}{2})^i$ ⇒i=$log_{3/2}$n

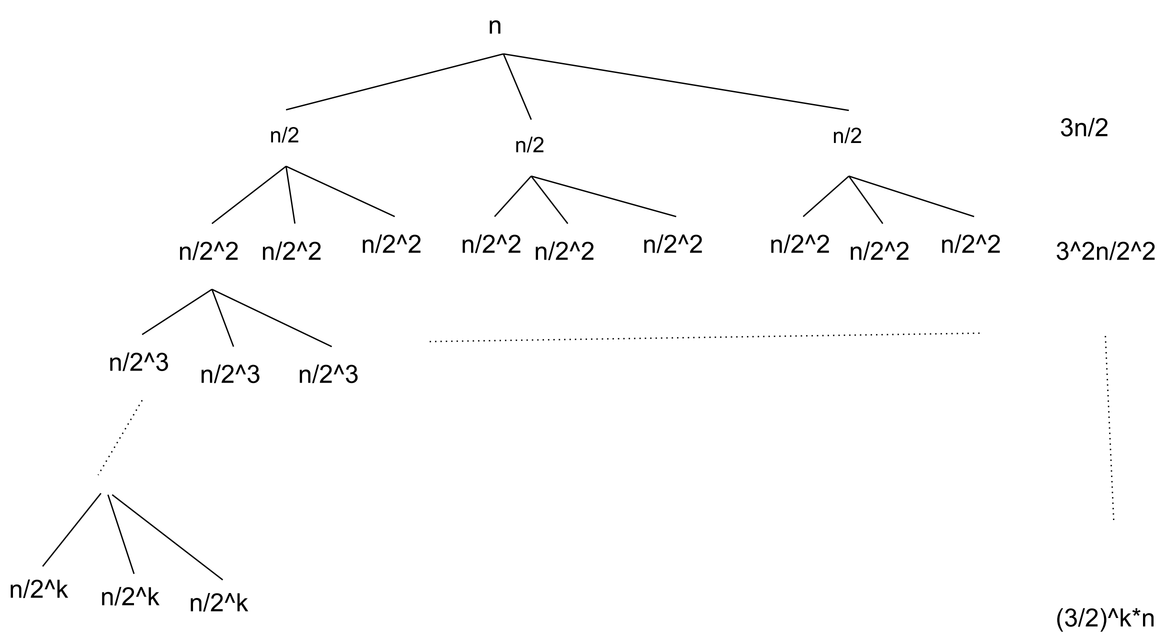
T(n)≤i*1= $log_{\frac{3}{2}} n$

T(n)=O($lg\ n$)

Tn=$\Theta\ (lg\ n)$
Step 2: T(n)=C*lg(n/3)+Clg(2n/3)+1
⇒T(n)= C(lg n-lg3)+C( lg n-lg 3/2)+1
=C lg n- C*lg 3+C *lg n+C*lg2-C*lg3+1=2C*lg n-C*2*lg3+C*lg2+1
≈C*lg n-C+1
Then, T(n)≤C*log n, if -C+1≤0.
Therefore, T(n)=O(log n), when C$\geq\ 1$

**b.** $T(n) = 3T\ (\ n/ 2\ ) + O(n)$

T(n)=3T(n/2)+Cn
T(n/2)=3T($\frac{n}{2^2}$) $+\ C\ *\ \frac{n}{2}$
T(n)=$3^2T(\frac{n}{2^2})\ +\ 3C\ *\ \frac{n}{2}\ +\ Cn$
T($\frac{n}{2^2}$) =3T($\frac{n}{2^3}$) $+\ C\ *\ \frac{n}{2^2}$
T(n)=$3^3T(\frac{n}{2^3})$+$3^2\ *\ C\frac{n}{2^2}$+$3C\ *\ \frac{n}{2}\ +\ Cn$
T(n)=$3^kT(\frac{n}{2^k})\ +\ n\ *\sum_{1}^{k}(1\ +\ \frac{3}{2}\ \ +\ (\frac{3}{2})^2\ +...+\ (\frac{3}{2})\ )$




At some point $\frac{n}{2^k}\ =\ 1$, k=$log_2n$

K level, and each level calculation is $(\frac{3}{2})^k$n≈n*$(\frac{3}{2})^{lgn}$=n*$\frac{3^{log_2 n}}{2^{log_2 n}} = 3^{lg\,n} = n^{log_2 3}$

Assume T(n)=Θ(n*$(\frac{3}{2})^{lgn}$)=Θ($n^{log_2 3}$)

prove:

T(n)=3*C($(\frac{n}{2})^{log_2 3}$ )+$C_2$n=$\frac{3}{2^{log_2 3}}$*C*$n^{log_2 3}$+$C_2$n=C*$n^{log_2 3}$+$C_2$n=C$n^{1.58}$+n

T(n)≤O($n^{log_2 3}$) for C≥0, n≥n0

**4. Find the order of growth of the following recurrence relations using the Master Method. a.** $T(n)$ **= 9**$T$ **(** $n$**/ 3 ) +** $n$

T(n)=a T($\frac{n}{b}$)+f(n)

X=$n^{log_b a}$

y=f(n)

a=9,b=3,x=$n^{log_3 9}$=$n^2$,y=n

⇒x>y. Thus, T(n)=$\Theta(x) = \Theta(n^2)$

**b.** $T$**(**$n$**) = 3**$T$ **(** $n$ **/4 ) +** $n$ **lg** $n$

a=3,b=4,x=$n^{log_4 3}$,y=n lg n

x<y, T(n)=Θ(n lg n).

When regularity condition is applied, af(n/b)≤ $cf(n)$ for some C<1

3(nlgn/4)≤ $C\ *\ nlgn$

⇒C≥ $\frac{3}{4}$

when $\frac{3}{4}$ ≤ $C$<1, the regularity condition is satisfied for T(n)=Θ(n lg n).

5. Can the master method be applied to the following recurrence? Why or why not?

$T(n)$ = 4$T$ ( $n$ /2 ) + $n$^2 lg n

a=4, b=2, x=$n^{log_2 4}$=$n^2$

Therefore, y>x

When regularity condition is applied, af(n/b)≤ $cf(n)$ for some C<1:

4(n^2lgn/2)≤ $C * n^2 lgn \Rightarrow 2n^2 lgn \leq C * n^2 lgn$

C≥2

As per the regularity condition C<1, master method does not apply to this case.

**REFLECTION QUESTIONS**

In no more than 2-5 sentences for each question, please answer the following:

**a) How much did you know about "data structures and algorithms" before starting this course?**

Before starting this course, I did not have much knowledge of the data structures and algorithms. I tried to work on some algorithm problems through Leetcode, but do not have full knowledge of why they should perform in this way and the time that they are presenting for.

**b) What assumptions did you have about the subject?**

The assumptions that I have about this subject are:

- Understanding how to calculate the time and space of the algorithms.
- Applying different algorithm techniques to solve data problems, like sorting, data searching, data pipeline.
- Ability to write algorithms by using different coding.

**c) Have you used data structures and algorithms to solve any real-life problems in your career/life?**

I do not have experience using data structures and algorithms to solve any real-life problems. As QA technician, I used inspection data to evaluate the products and summarize our inspection performance. However, I applied the data structures and algorithms to the work.

**d) Is there a problem you would like to solve/see this being useful somewhere in your work/life?**

There is definitely that I can apply data structures and algorithms to solve work problems. During the work, we collected different data from different systems or different applications. To summarize or search for certain information, it is even more difficult if the data comes from different sources or platforms. Data structure and algorithms can help to restructure the data and generate the data that we would like to obtain.

**e) From what you have learned so far, is the subject matter what you expected? Why or why not?**
I had better understood the time complexity concept of the algorithm. Applying effective algorithms to real life is the main goal for me to learn this subject, but this goal is still unachievable.

## References

Rosenburg, A. N. (n.d.). *Algorithmic efficiency*. Wikipedia. Retrieved May 20, 2023, from https://en.wikipedia.org/wiki/Algorithmic_efficiency