CS301 Assignment 1

Oğuzhan Özdemir 20979

1 Problem 1 Recurrences

(a)$\Theta(n^3)$
$(b)\Theta(n^{\log_2 7})$
$(c)\Theta(\sqrt{n}lgn)$
$(d)\Theta(n^2)$

2 Problem 2 Longest Common Subsequence

a)i)To find out the longest common subsequence we need to find out every subsets of set. This takes $O(2^n)$ time and comparison between two sets takes length of second set so its still dominated by $O(2^n)$. We can prove it by making recursion tree.
lcs(ABC, ACB)
/
lcs(AB, ACB) lcs(ABC, AC)
/ /
lcs(A, AC), lcs(AB, A)
lcs(A, A) lcs(A, A)
So it multiply running time when it cant find match by double worst case scenario $O(2^n)$ times .

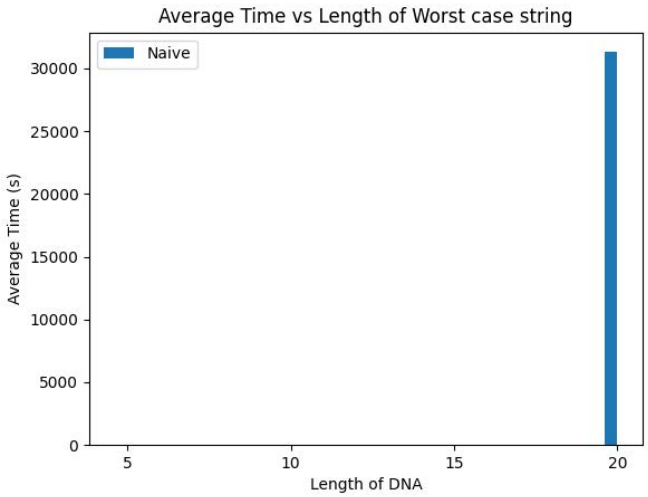
ii)The recursive algorithm with memoization takes the sets by collective. We can think this like matrix form. First element from first set checks with first element of second set then it checks with first and second element from second set. Let assume first set X have x1, x2, x3 second set Y have y1, y2, y3 and their length of sets lx and ly so it checks like x1 to y1 then x1 y1,y2 then x1 y1 y2 y3. After checking with x1 now we check x1,x2 to y1 it goes like this. We collects elements and encounters on the way collectively.Because of that best asymptotic worst-case running time of the recursive algorithm with memoization is O(lxly).

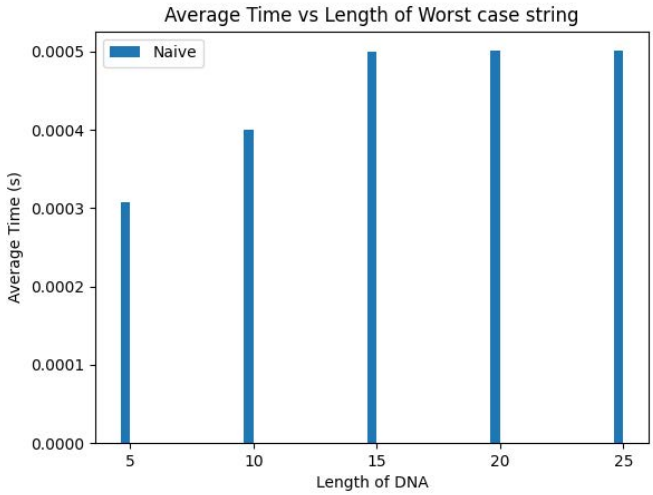
b)i) (Naive approach of 25 length didnt finished after 3 days running.)
PC specs CPU: Ryzen 9 5950X 16 core, RAM: 32gb 3600mhz, OS:windows 10 Pro-n 64bit

5	10	15	20	25
0.0005138999999999561	0.19127860000000002	33.993510099999995	31306.3898981	-*
0.0003075000000000161	0.00040049999999997	0.0004997253417968	0.000500679016113	0.00050091743469

Table 1: Worst case runtime by second

b)ii)





b)iii) Result of the experiment of worst case running time naive approach is horrible for scaling. The memoization approach easily scaled performed instantly however naive keep getting slower quickly. At 5 string length there isn't any recognizable difference but for example at 15 length there is 60,702.6785 times difference. At 25 length, i couldn't find solution after 3 days running time. Experimental results confirmed theoretical results. Exponential time worse then linear time.

c)i

Table 2: average running times in seconds (µ), and the standard deviation (σ)

5µ	5σ	10µ	10σ
6.669362386067709e-05	0.00017294289970201532	0.0018016020456949869	0.0015016996106074225
3.3346811930338543e-05	0.00012690537574170177	0.00011680126190185546	0.00021533985971495956

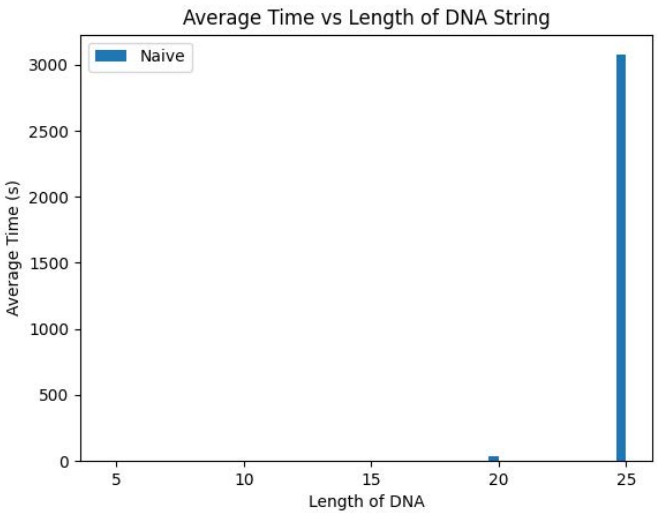
Table 3: average running times in seconds (µ), and the standard deviation (σ)

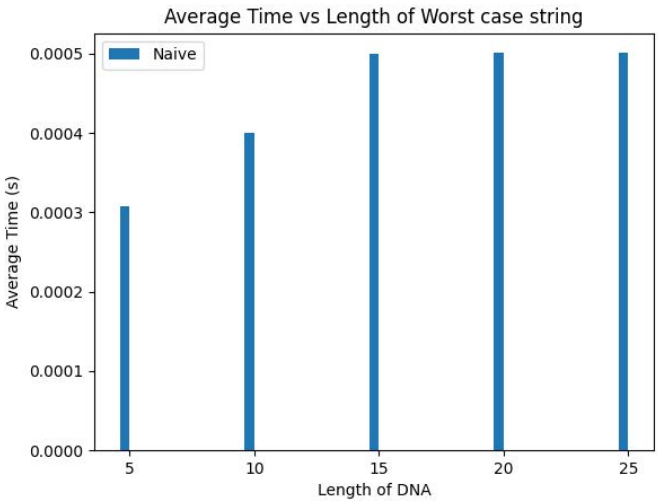
15µ	15σ	20µ	20σ
0.16711046695709228	0.11860847179209576	35.94946860472361	52.52794327836527
0.0001334508260091146	0.00022508648873823347	0.0002002080281575521	0.0002493957363429655

Table 4: average running times in seconds (µ), and the standard deviation (σ)

25µ	25σ
3075.0650581280393	5005.161071807295
0.0002502123514811198	0.00025448993007253857

c)ii)





c)iii) For this experiment results naive approach still not scalable but better then worst case analysis. At least in this experiment there is result for 25 length DNA sequence. In this experiment there isn't any 0 lcs DNA sequence. So naive approach only goes $O(2^n)$ when there isn't any common characters. Still the difference between 2 approaches less then worst case but still naive is not usable in longer cases. For example on some occasion 25 length DNA sequence took 26819 second.