<u>Question 1 Report</u>

b)

i) $f(n) = \Theta(g(n))$ where $0 <= c1*g(n) <= f(n) <= c2*g(n)$

$g(n) = n$ => This is because we iterate n times in this algorithm

asymptotic running time complexity = Θ(n)

ii) Recurrence Equation = $T(n/2) + \Theta(1)$

Cost in i^{th} height = $n/2^i$

Height $=\log_2 n$

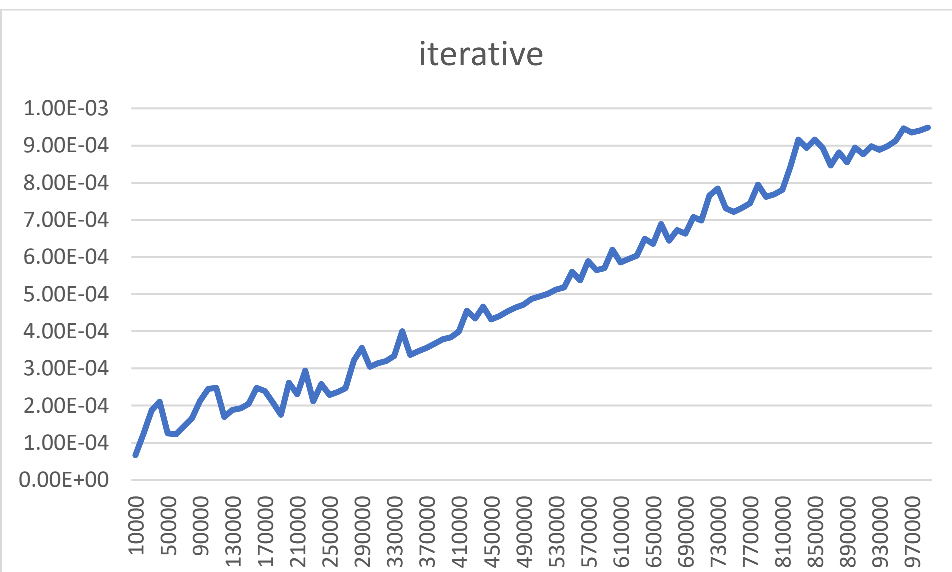
Cost of leaves = $2^\wedge(\log_2 n) = n^\wedge(\log_2 1) = 1$

Cost of tree = $\sum_{i=0}^{h-1} a^i * f(n/b^\wedge i) = \sum_{i=0}^{h-1} n/2^\wedge i$

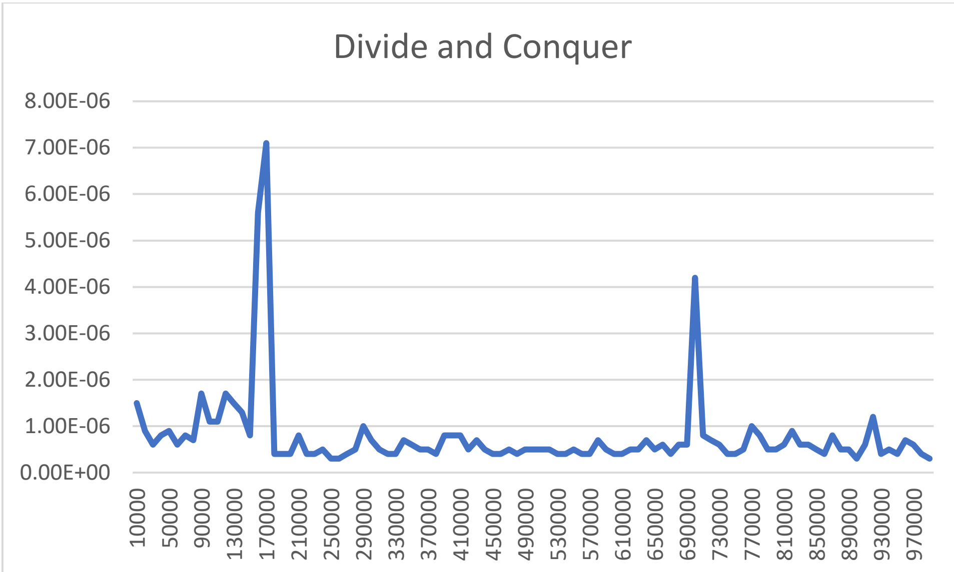
$T(n) = \log(n) + 1$

asymptotic running time complexity = Θ(log(n))

c)



This graph shows that the real time complexity is approximately n for the iterative algorithm.



This graph shows that the real time complexity is approximately log(n) for the divide and conquer algorithm, however as you can see there is some spikes in the graph due to me running other applications while plotting the graph.

d) the graphs show that that the real time complexity is approximately the same as the ones we found in part b. As we increase the value of n the divide and conquer algorithm takes less execution time than the iterative algorithm therefore the divide and conquer had better scalability than the iterative algorithm for big values of n.