LAB REPORT : COMPSCI 2XB3

LAB SECTION – L02

Teaching Assistant - *Seyed Parsa Tayefeh Morsal*

Submitted By –

GROUP NUMBER : 2

Name : Adhya Goel

Student Number : 400280182

McMaster email : goela10@mcmaster.ca

Name : Mridul Arora (Contact Member)

Student Number : 400253526

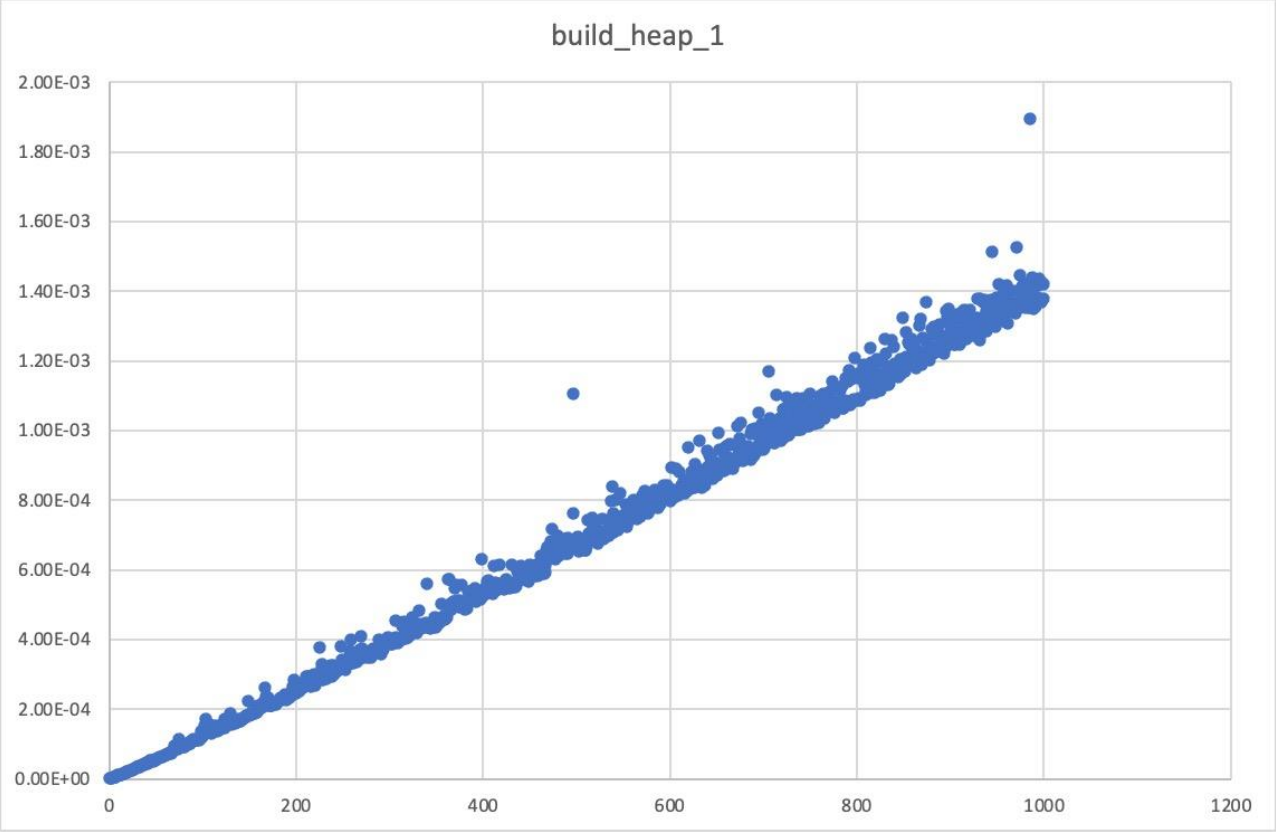
McMaster email : aroram15@mcmaster.ca

Building Heaps

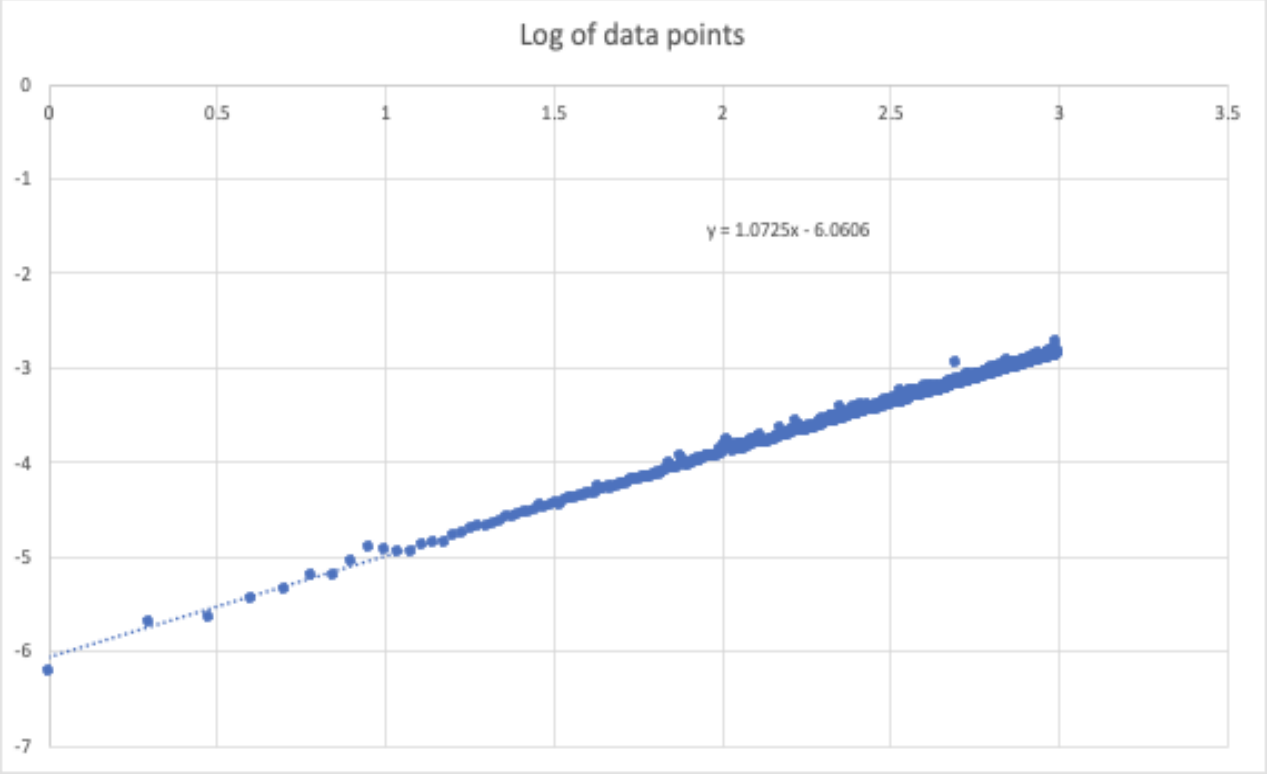
An estimation is made for the complexity of all the three build heap implementations to be nlog(n). This is because, all the three functions consist of for loops which has n complexity and it is given that insert and sink functions has O(log n) complexity. Thus, nlog(n) is taken to be the complexities of all the three functions.

- build_heap_1()

  Initially, the following graph shows it gives linear trendline:



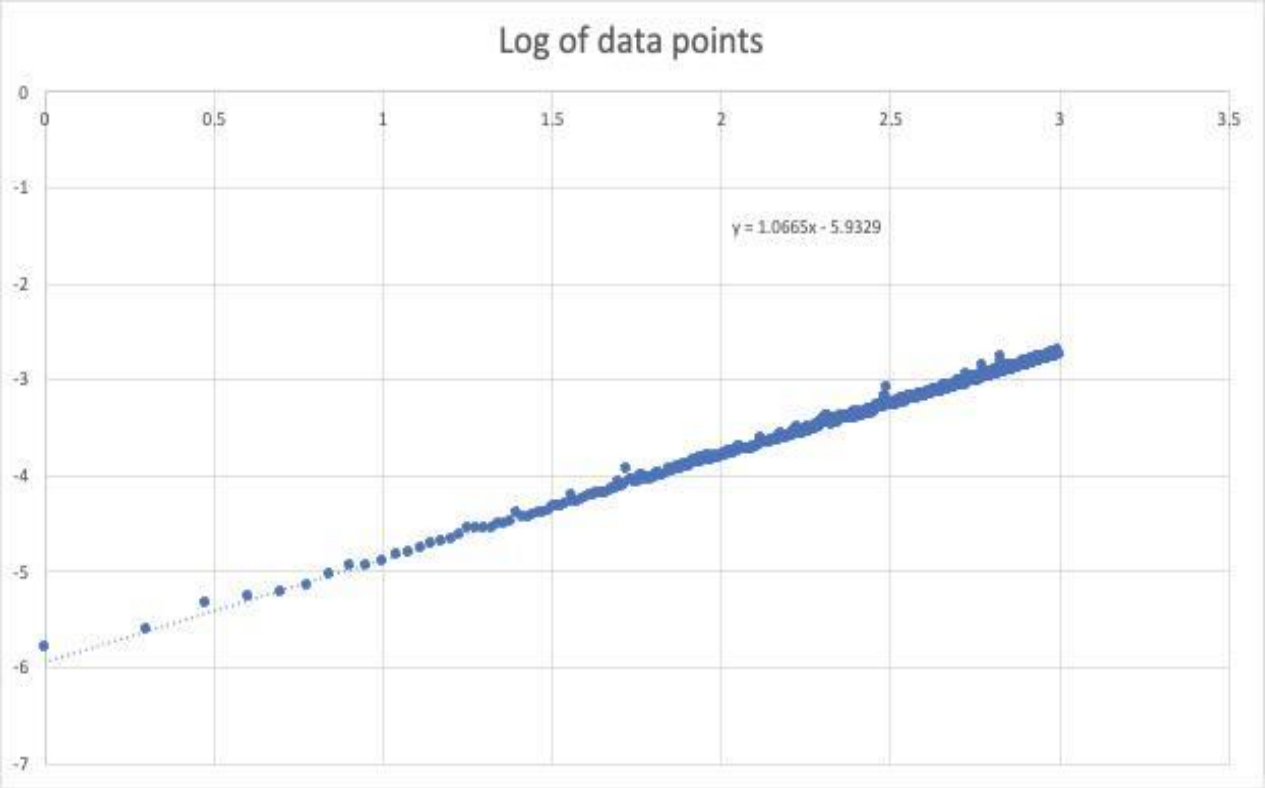
So, to obtain the actual runtime graph for build_heap_1() a graph was plotted for log of n and log of runtime where n is the length of the list and the slope came out to be 1.0725. So, the assumption came out to be wrong as after experimenting the graph came out to be linear.



- build_heap_2()

  Initially, the following graph shows it gives linear trendline:

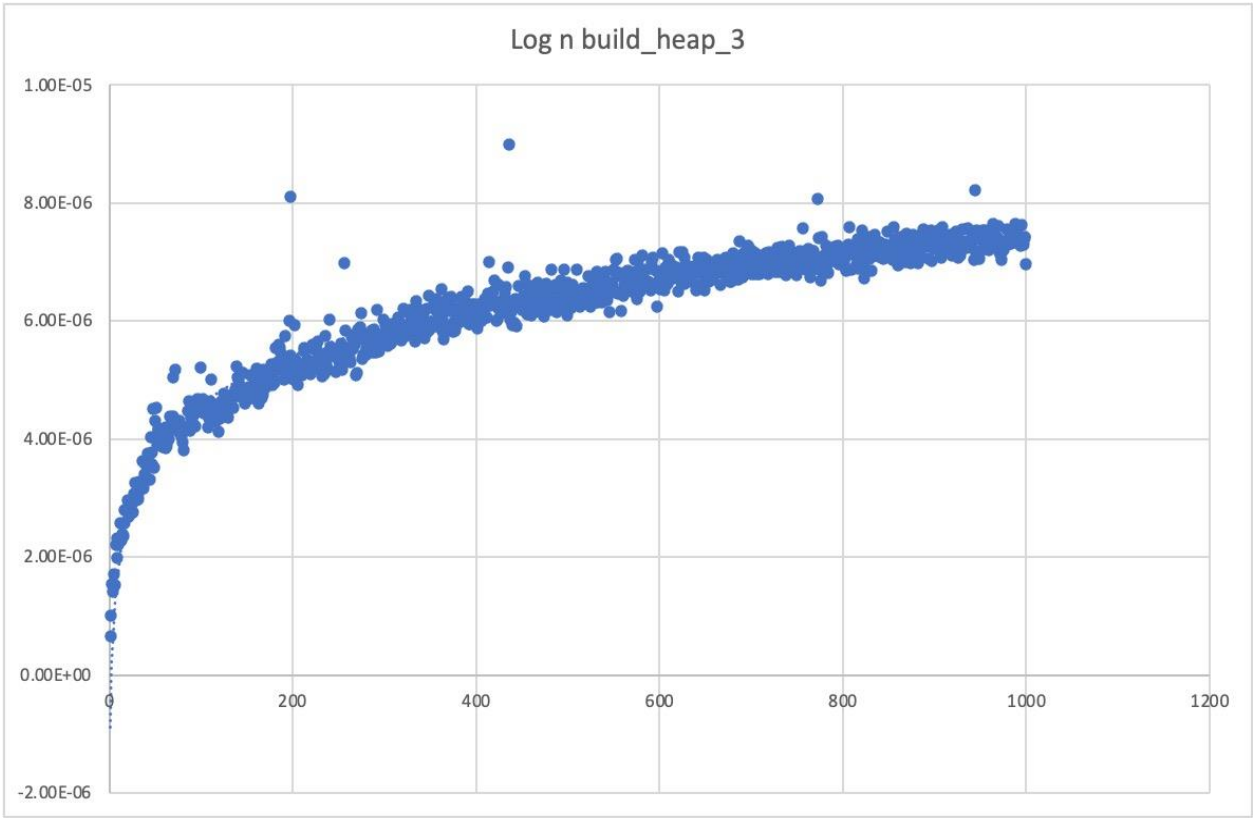
So, to obtain the actual runtime graph for build_heap_2() a graph was plotted for log of n and log of runtime where n is the length of the list and the slope came out to be 1.0665. So, the assumption came out to be wrong as after experimenting the graph came out to be linear.



- build_heap_3()

  Initially, the following graph shows it gives linear trendline:

So, to obtain the actual runtime graph for build_heap_3() divide the runtime by n to get the graph of log(n) where n is the length of the list. And, indeed after experimenting the graph came out to be log(n).



For build_heap_3 when we run for loop for n where n is the length of the list, it performs poorly. But, when the for loop is run for n/2 it performs more quickly and precisely.

K-heap sort

K-heap sort allows decrease priority operations to be performed more quickly than binary or traditional heaps, also it has slower delete minimum operations. Function sink() of k-heaps has a complexity of $O(k \log_k n)$ as it calls itself recursively for k children.