I certify that every answer in this assignment is the result of my own work; that I have neither copied off the Internet nor from any one else's work; and I have not shared my answers or attempts at answers with anyone else.

1: Suppose a B-tree of minimum degree 3 (t = 3) has only one node — the root (also a leaf) — containing keys 30, 50, 70, 90, and 110.
Based on the algorithm covered in class, show the state of the tree after insertions of keys 20, 160, 80, 60, 35, 100, 75, 85, 95, and 99 in that order.
Draw the tree before and after every insertion that involves splitting of some node.
Indicate clearly the key just inserted.
Draw the final state

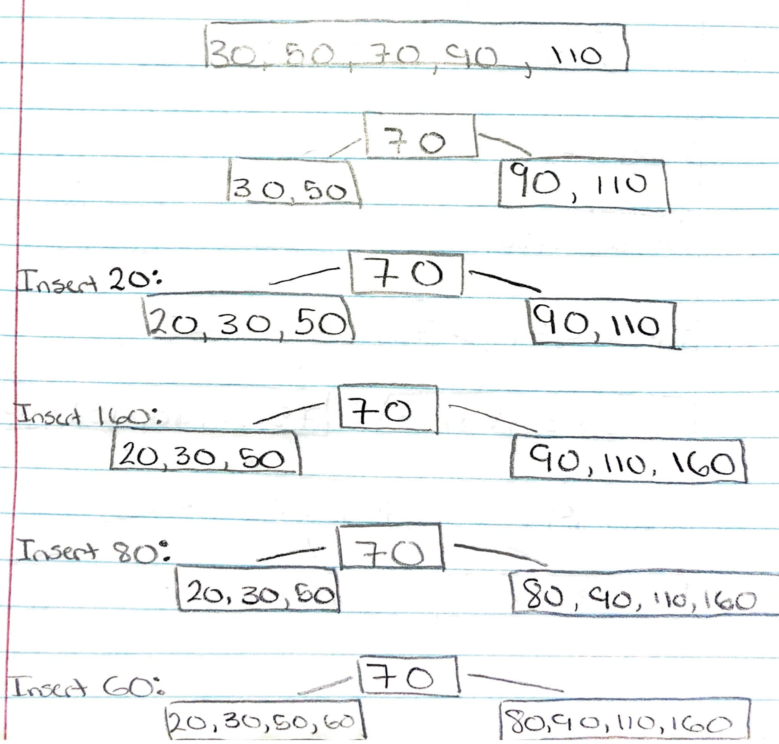
HW 8

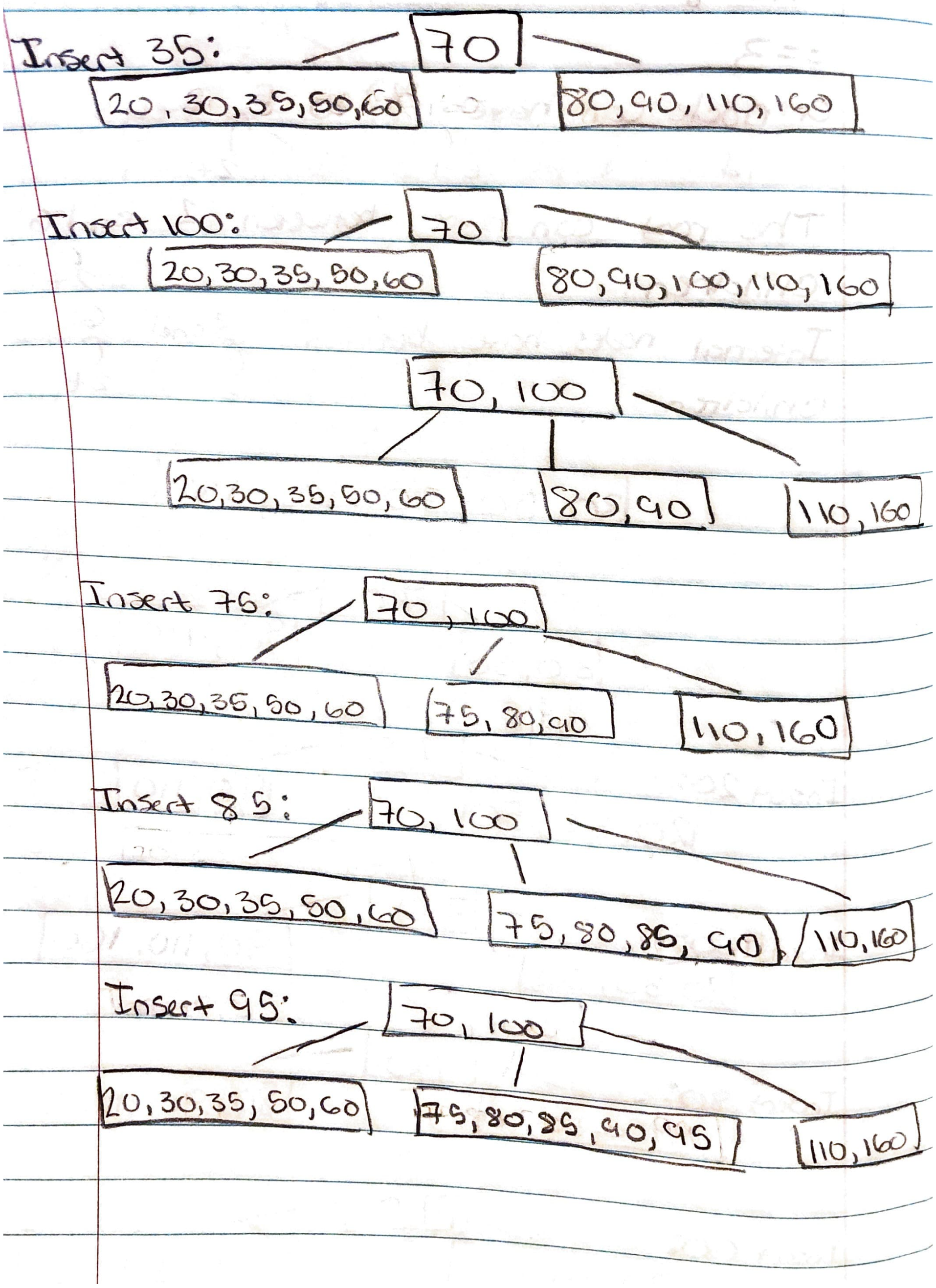
t=3

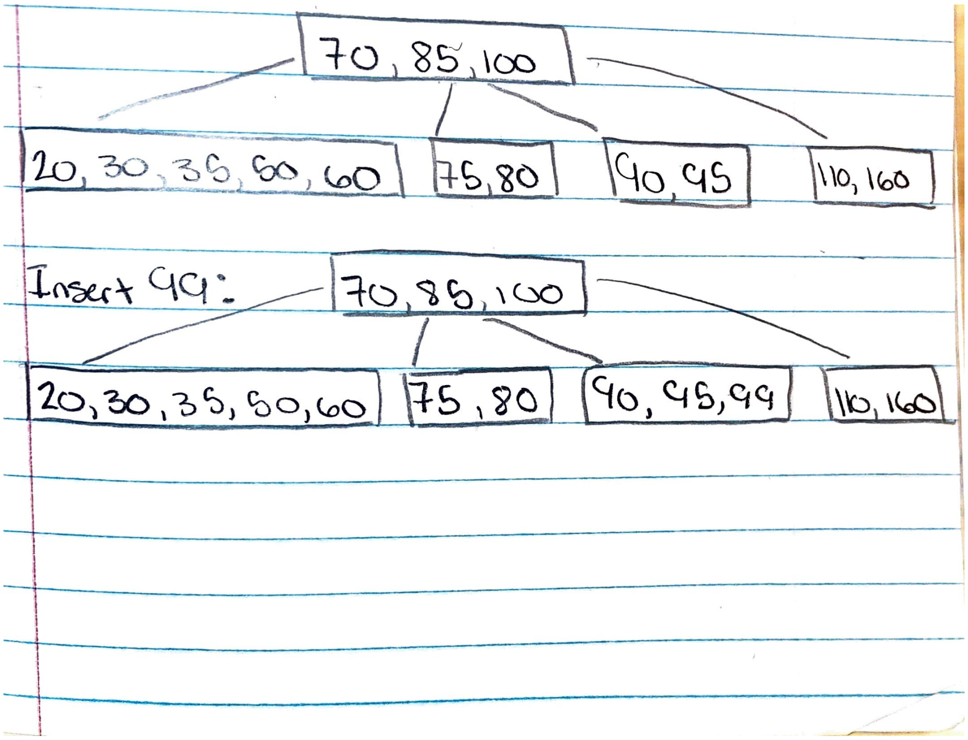
a node can have 2 to 5 keys
$\uparrow$ t-1 $\uparrow$ 2t-1

The root can have between 2 and 6 children
$\uparrow$ 2t

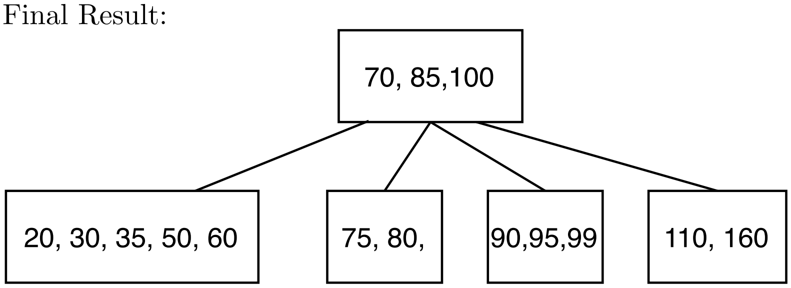
Internal nodes have between 3 and 6 children
$\uparrow$ t $\uparrow$ 2t




Insert 35:
70
20, 30, 35, 50, 60
80, 90, 110, 160
Insert 100:
70
20, 30, 35, 50, 60
80, 90, 100, 110, 160
70, 100
20, 30, 35, 50, 60
80, 90
110, 160
Insert 75:
70, 100
20, 30, 35, 50, 60
75, 80, 90
110, 160
Insert 85:
70, 100
20, 30, 35, 50, 60
75, 80, 85, 90
110, 160
Insert 95:
70, 100
20, 30, 35, 50, 60
75, 80, 85, 90, 95
110, 160

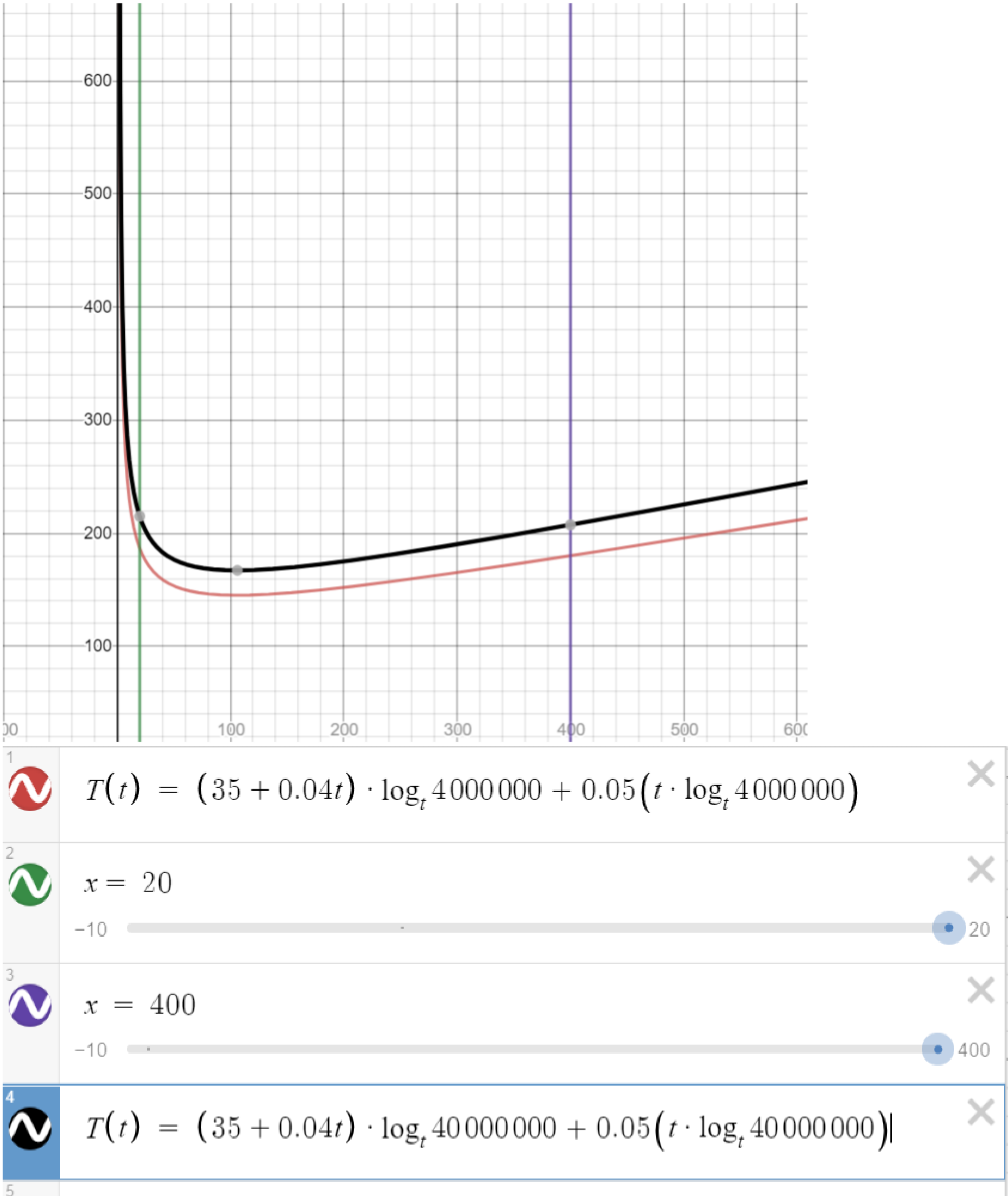


Final Result:



2: Consider the worst-case search of a B-tree of minimum degree t containing n keys using the algorithm BTREESEARCH (see the text). Suppose the constant hidden in the O() describing the CPU time is 5 microseconds and that the time taken by DISKREAD is a + bt, where a = 35 milliseconds, b = 40 microseconds. Further assume that the number of non-root levels in the B-tree is $log_t(\frac{n}{4})$
(a) Plot the worst-case time taken by B-TREE-SEARCH as a function of t when n = 4, 000, 000. Let t range at least from 20 through 400. Hint: If your plotting software only supports natural logarithms, then express $log_t n$ in terms of ln n and ln t.. Similarly, if your plotting software only supports logarithms to the base 10, then express logt n in terms of log n and log t.
(b) Plot a similar graph for n = 40,000,000
(c) What do you infer regarding a suitable range for t?

T(n) = (a + bt)(height($log_t n$) + O($tlog_t n$)
We Plot T(n) vs t. Here's the two graphs we get for (a) and (b). (a) is in red, (b) is in black



(c) It seems to me that a good range for t would be $20 \le t \le 110$, at this point we see the lowest dip in both graphs before the begins increasing over time again.