Kevin Nguyen

CSCI 2270

24 July 2020

Final Project Report

After running 100 iterations of 100 inserts and searches, I can conclude heaps are more time efficient than linked lists. As can be seen from the performance plot of the insert functions for the doubly linked list and the quadratic hash table, the quadratic hash table outperforms the linked list (Fig. 9). The insert for the doubly linked list appears to be operating at O(n) time complexity, while the quadratic hash table operates at an average of O(1) time complexity. Similarly, for the insert comparison, the quadratic hash again outperforms the doubly linked list with O(1) and O(n) time complexities, respectively (Fig. 10). The reason for these differing performances is likely due to the fact that insert and search operations on a doubly linked list require looping through each element in the list until the desired position is found. Alternatively, the hash function provides an initial position after a single computation. Barring any collisions, this operation immediately provides the desired position. However, collisions do lead to insert times that are greater than O(1) (Figs. 3–8). Chaining, linear probing, and quadratic probing all perform very similarly to each other in terms of performance. Quadratic probing seems to handle collisions better than the other hashing methods, maxing out at about 12,000 nanoseconds (dataset A). This is because quadratic probing only encounters the problem of secondary clustering, whereas linear probing results in primary clustering. In summation, hash tables should be preferred over linked lists when dealing with large datasets.

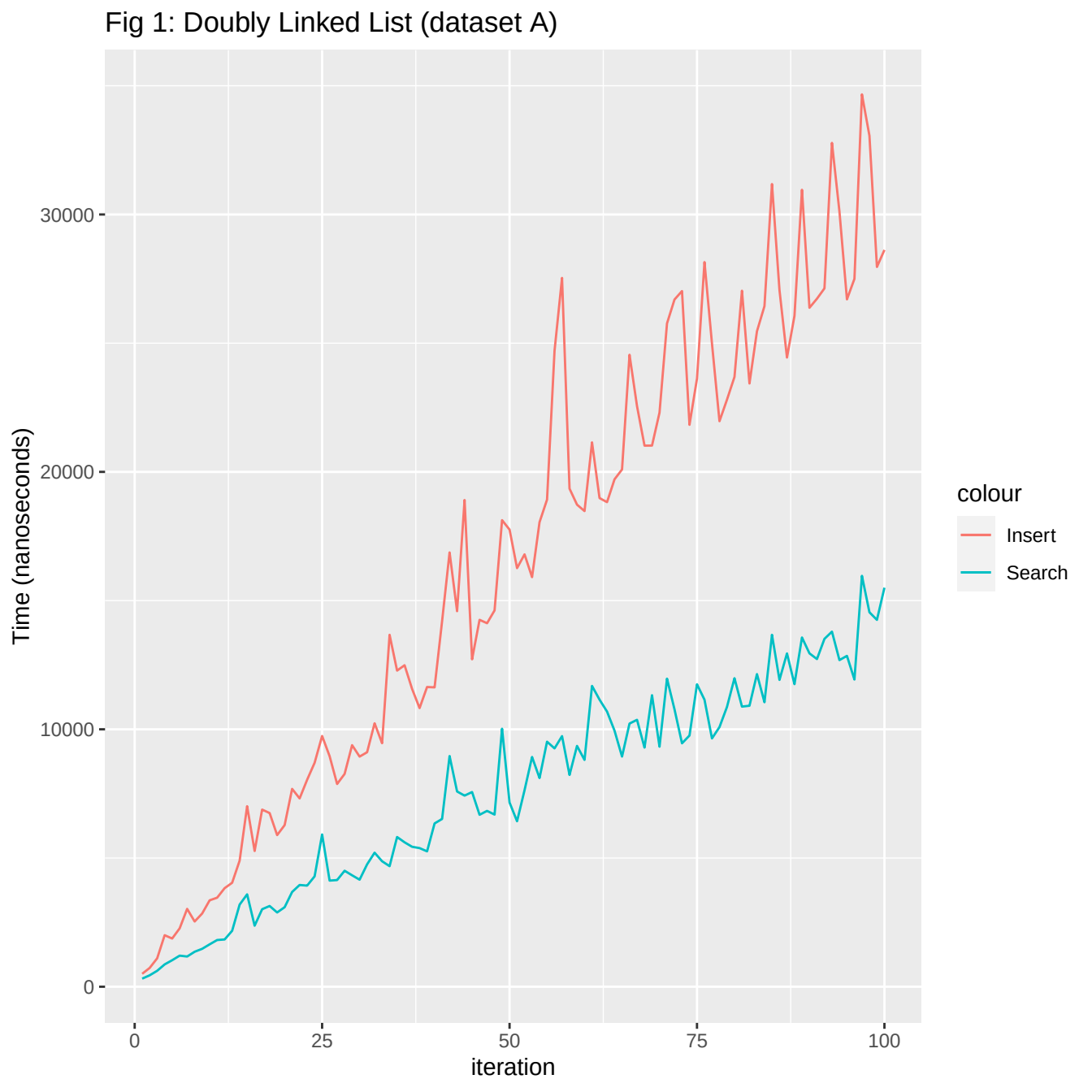


Fig 1: Doubly Linked List (dataset A)
Time (nanoseconds)
30000
20000
10000
0
0
25
50
75
100
iteration
colour
Insert
Search

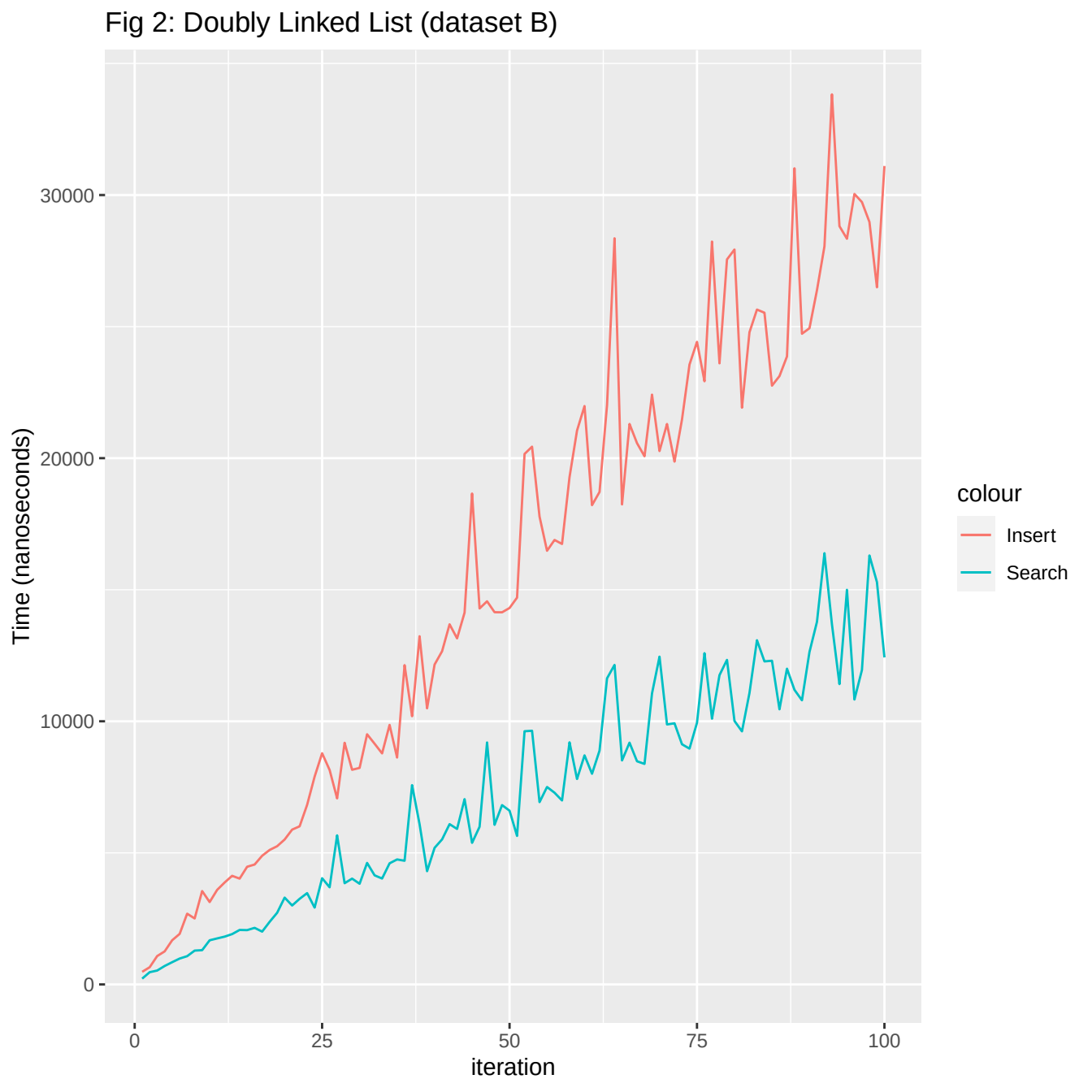
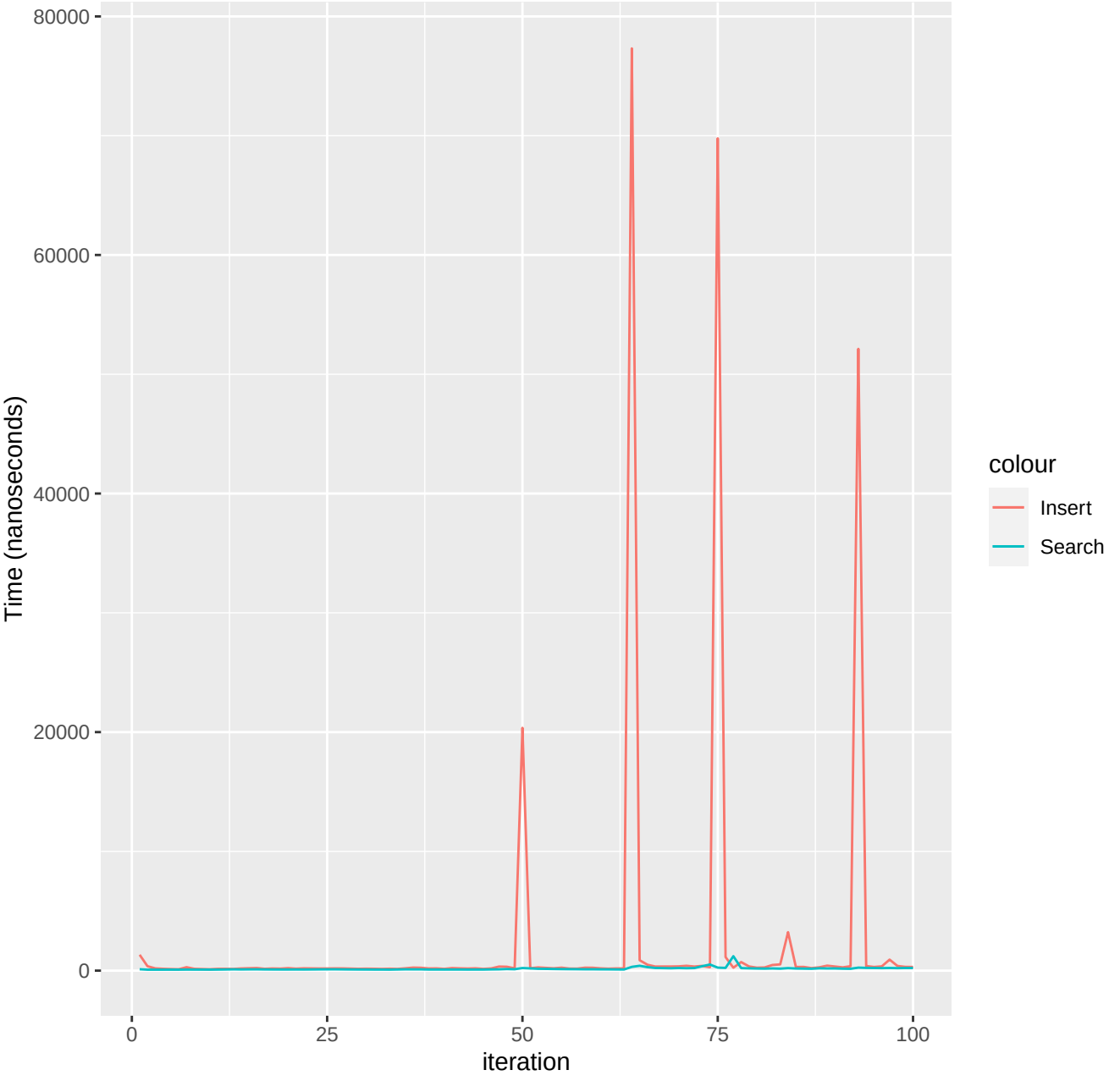


Fig 2: Doubly Linked List (dataset B)
Time (nanoseconds)
30000
20000
10000
0
0
25
50
75
100
iteration
colour
Insert
Search

Fig 3: Hash Table, Chained (dataset A)



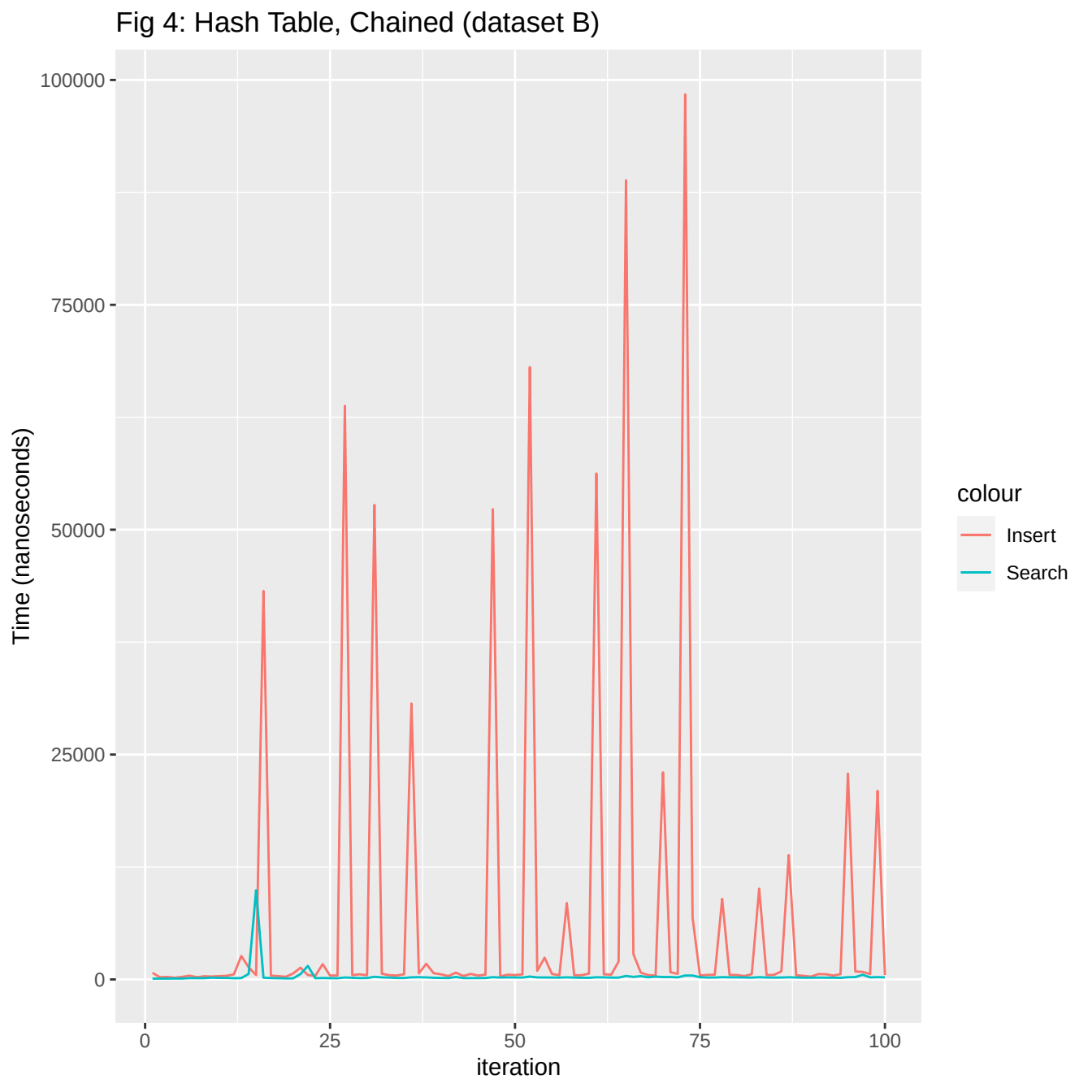


Fig 4: Hash Table, Chained (dataset B)
100000
75000
50000
25000
0
Time (nanoseconds)
0
25
50
75
100
iteration
colour
Insert
Search

Fig 5: Hash Table, Linear (dataset A)

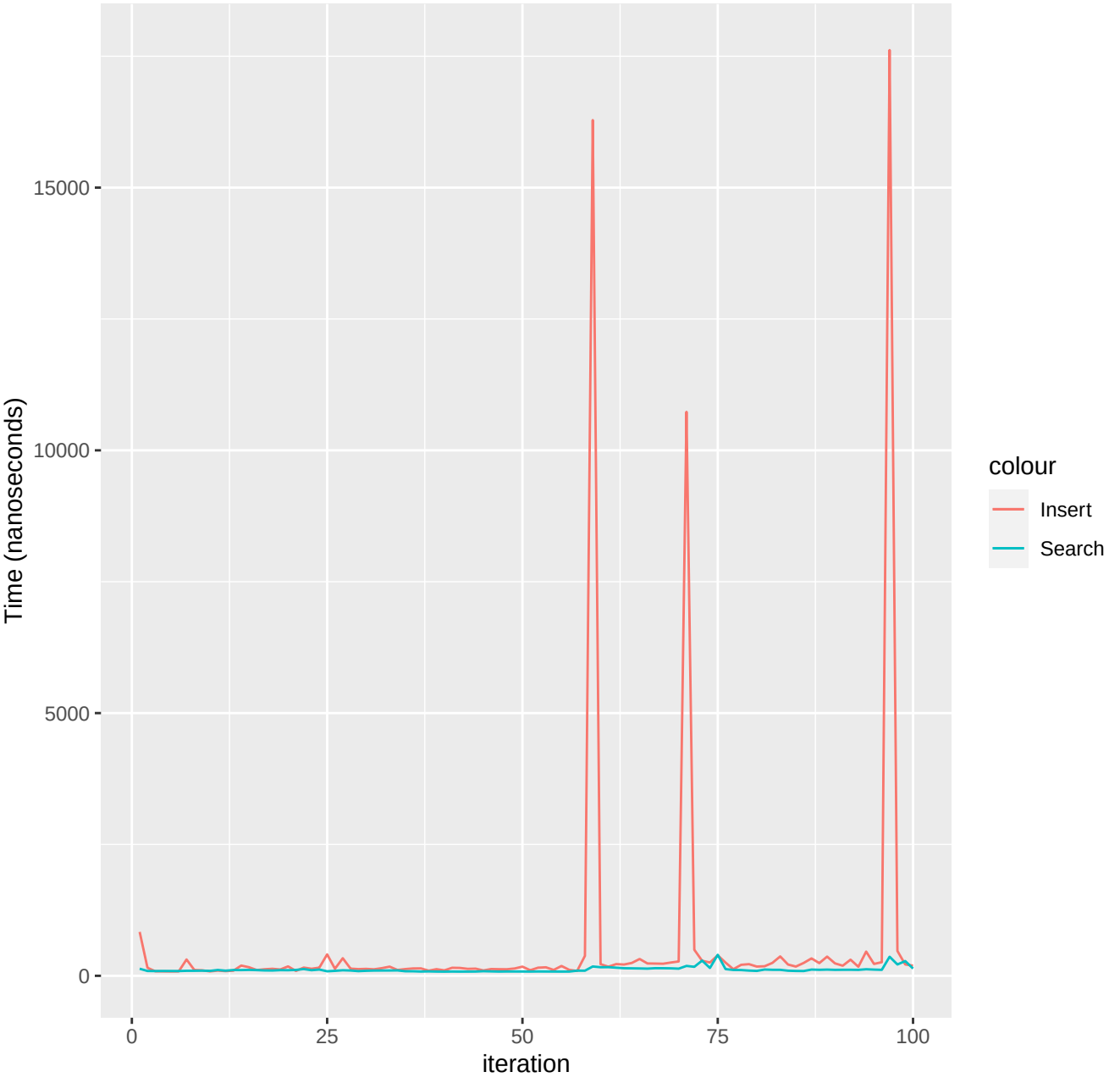


Fig 6: Hash Table, Linear (dataset B)

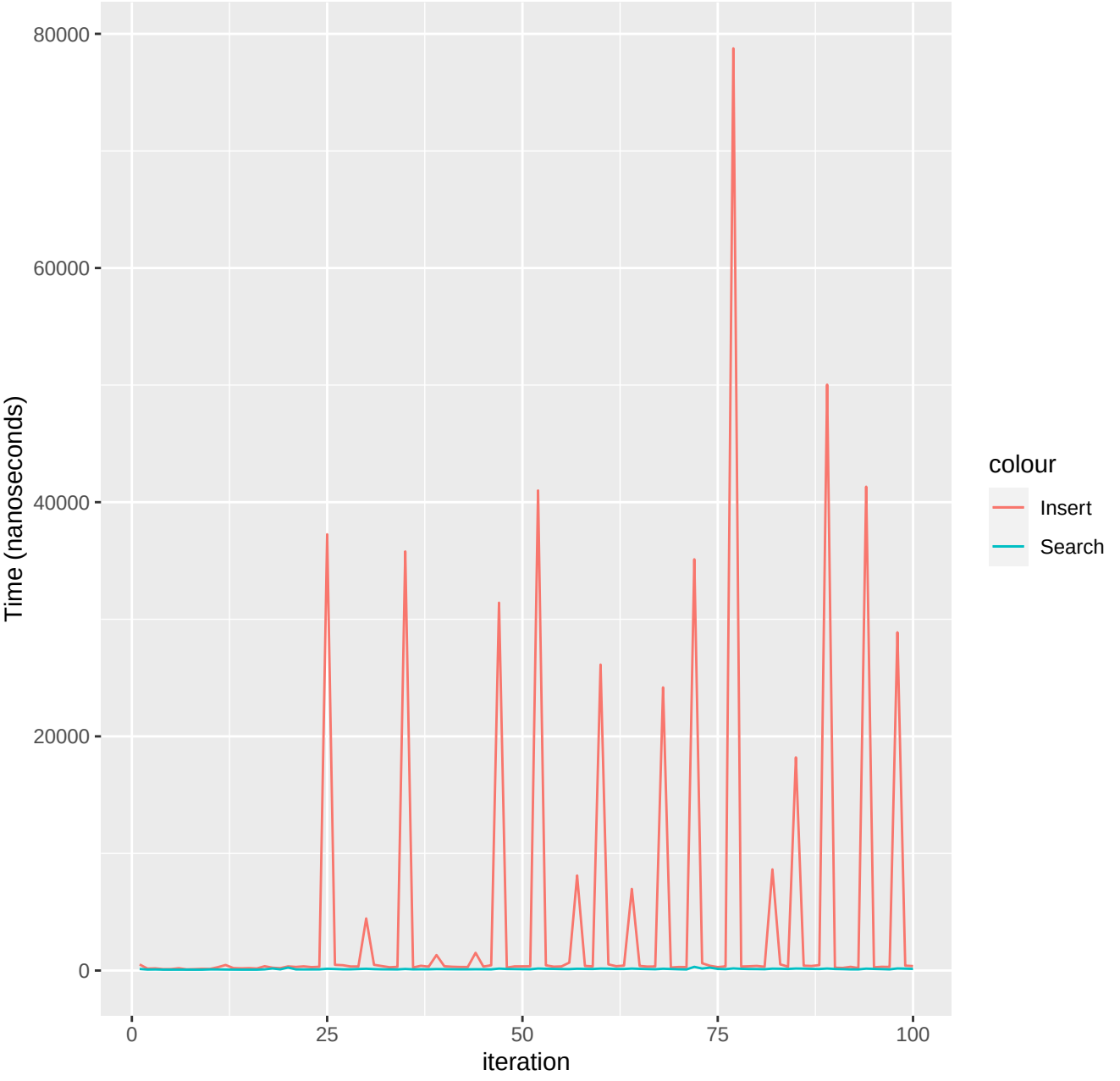


Fig 7: Hash Table, Quadratic (dataset A)

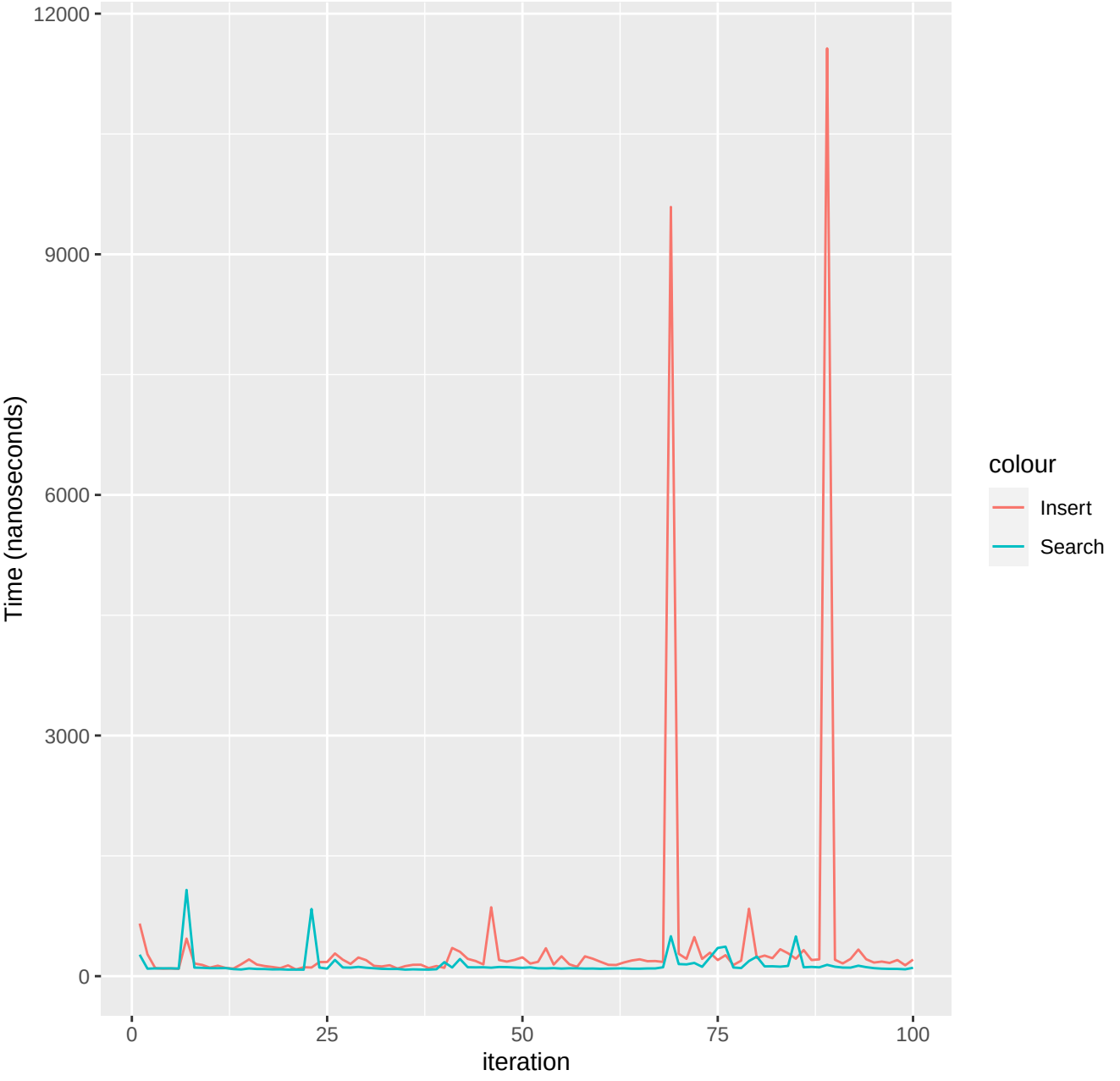


Fig 8: Hash Table, Quadratic (dataset B)

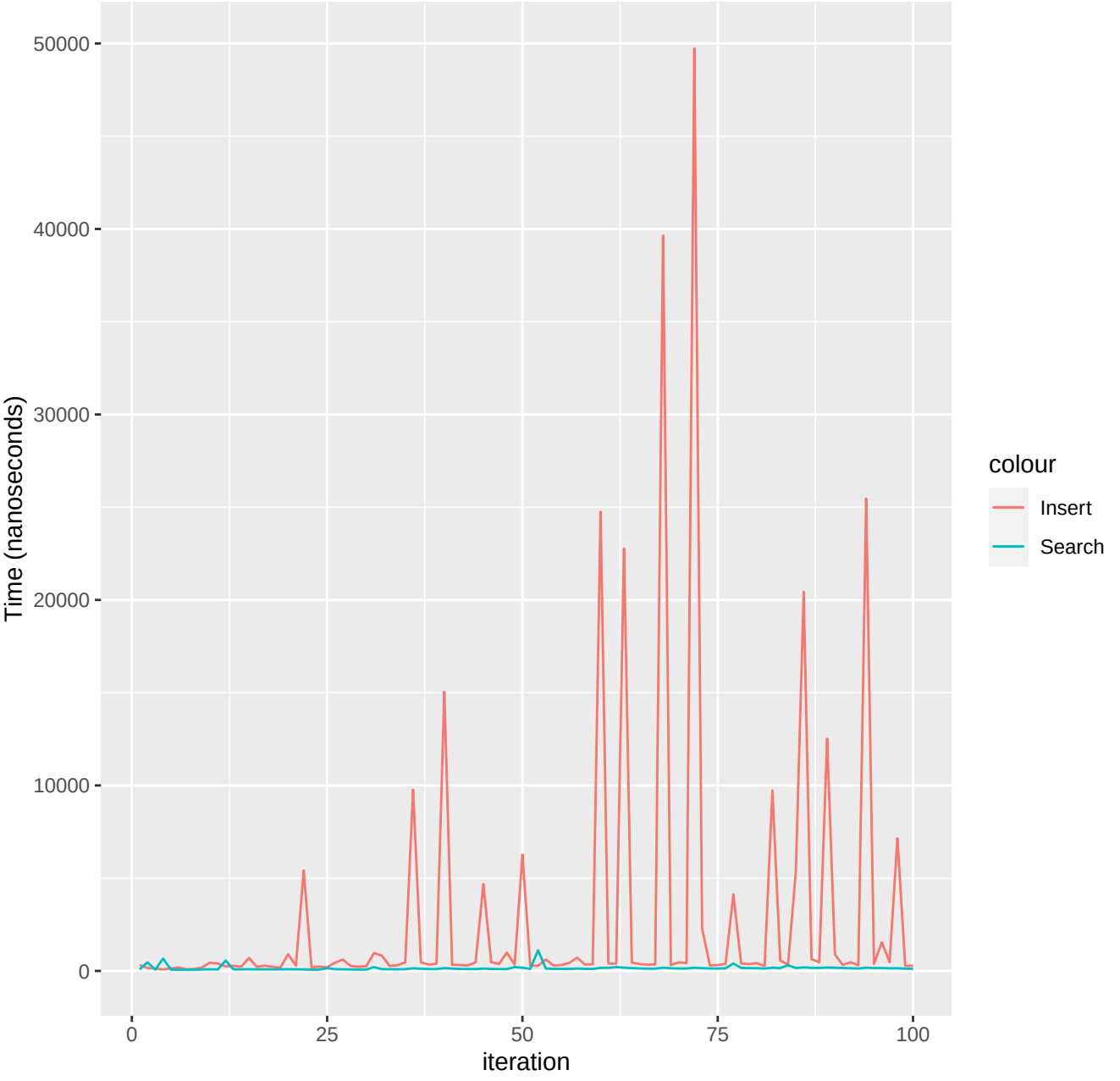
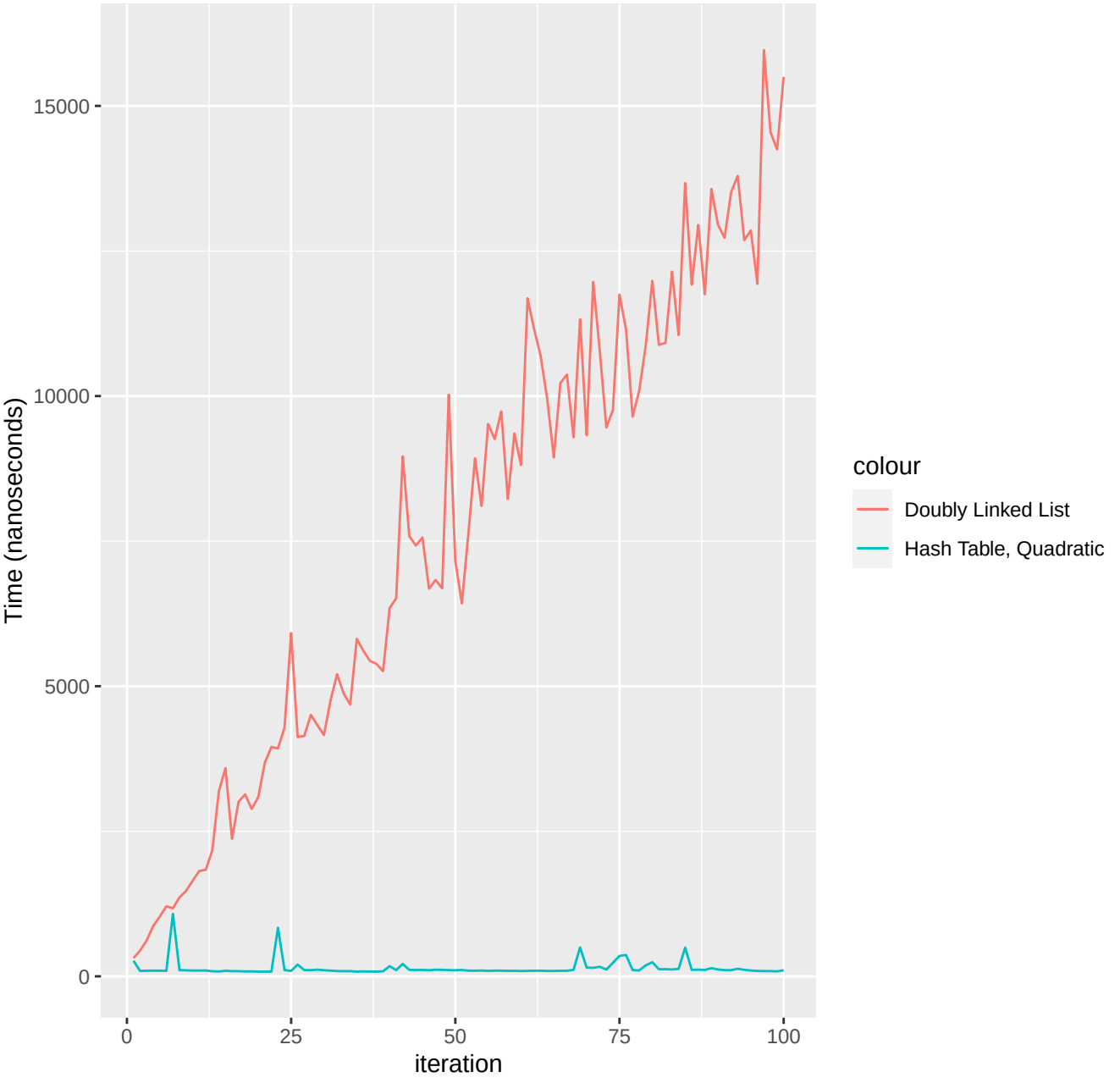


Fig 9: Search comparison (dataset A)



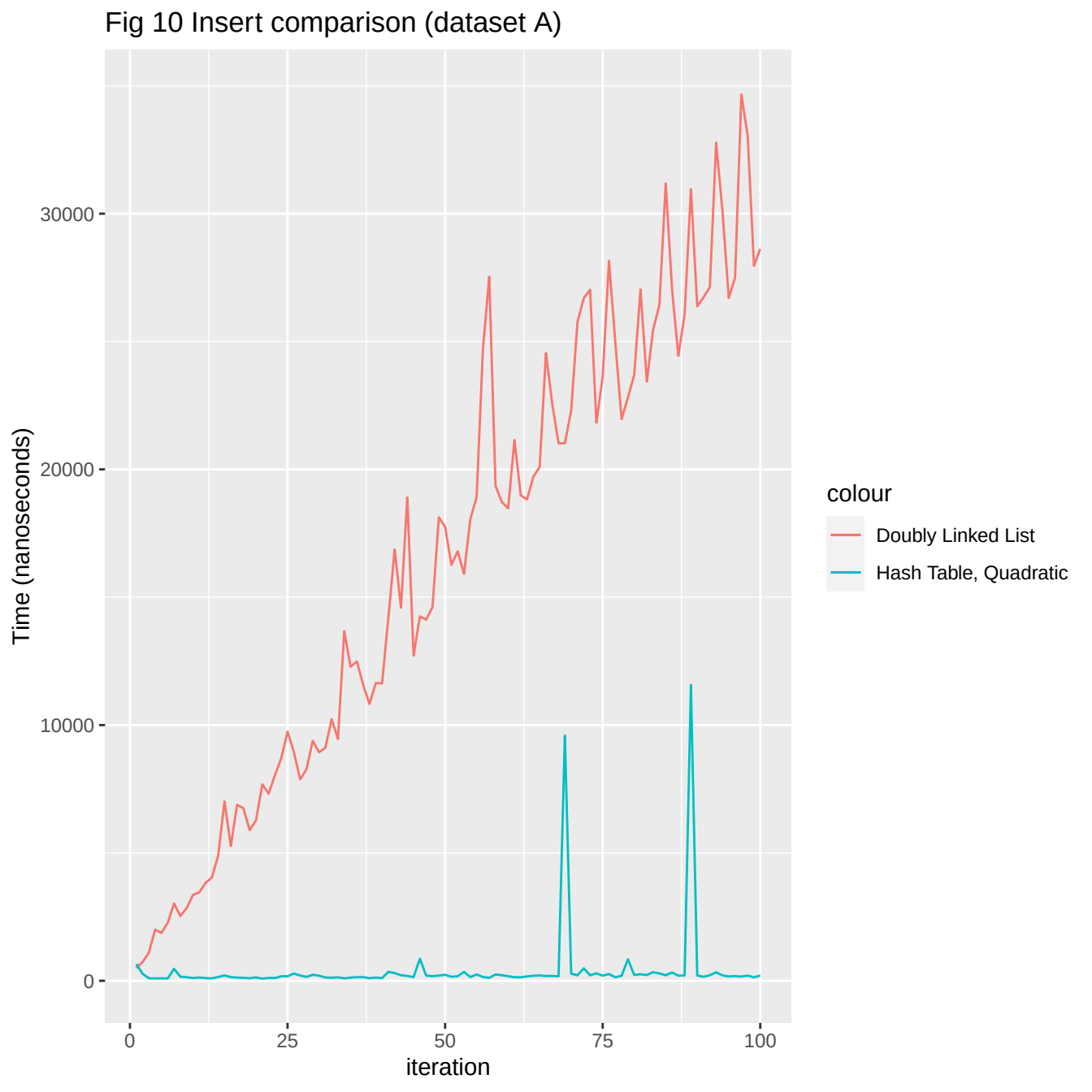


Fig 10 Insert comparison (dataset A)
Time (nanoseconds)
30000
20000
10000
0
0
25
50
75
100
iteration
colour
Doubly Linked List
Hash Table, Quadratic