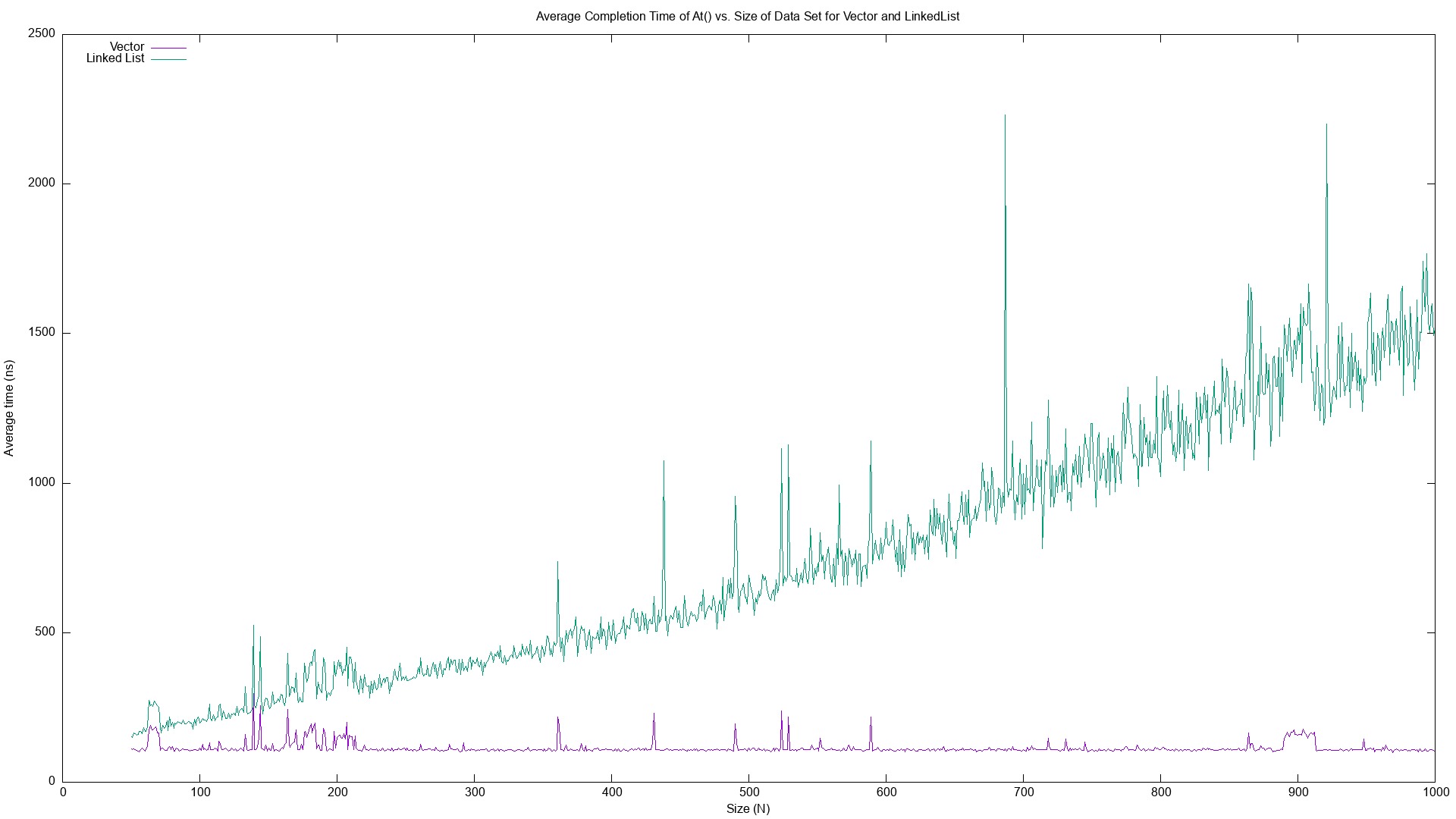
**Vector vs Linked List Analysis**

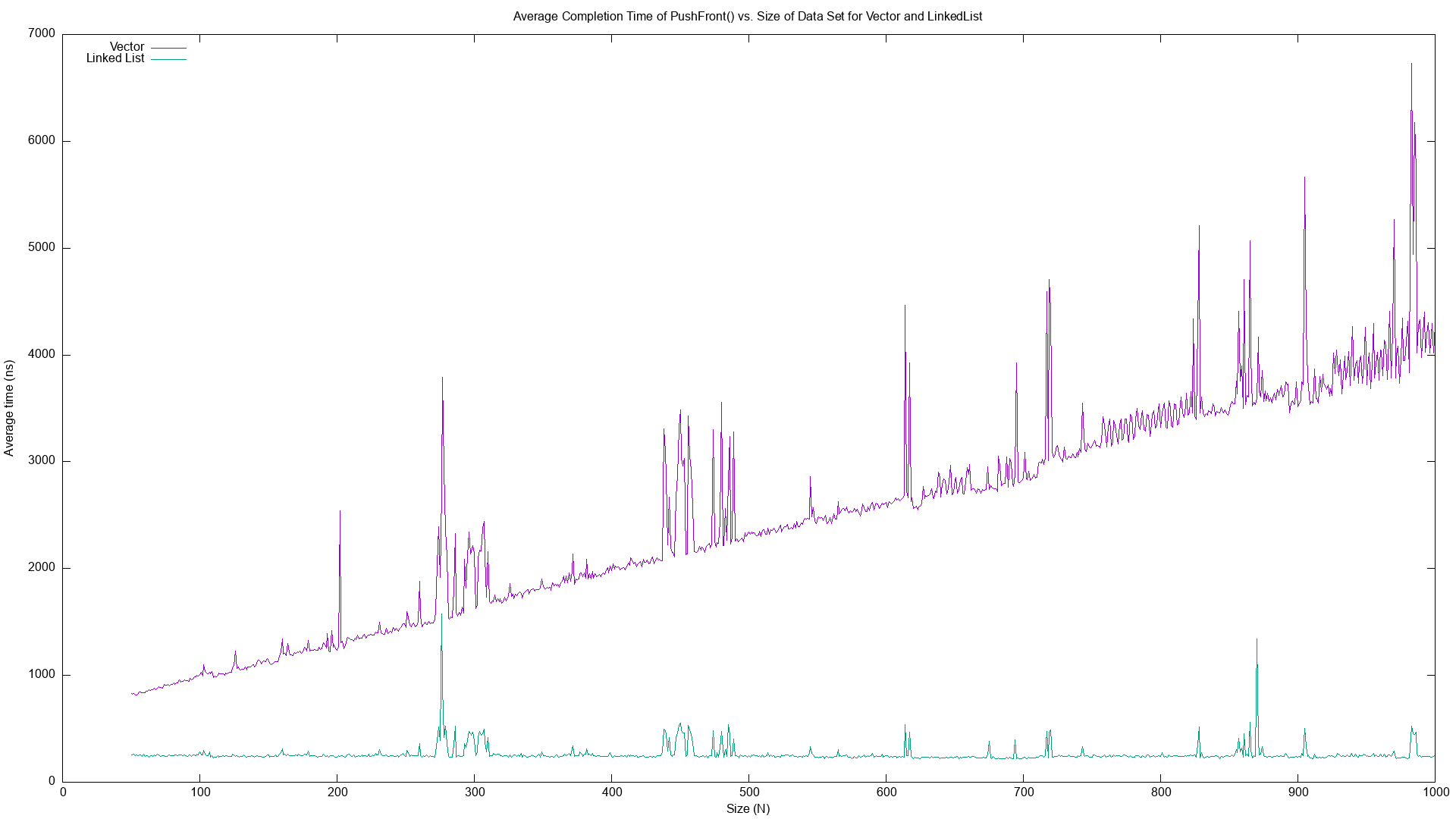
1. At()



The time complexity of the vector is O(1), so the average completion time is almost constant.

The time complexity of the linked list is O(n), so the average completion time grows linearly.

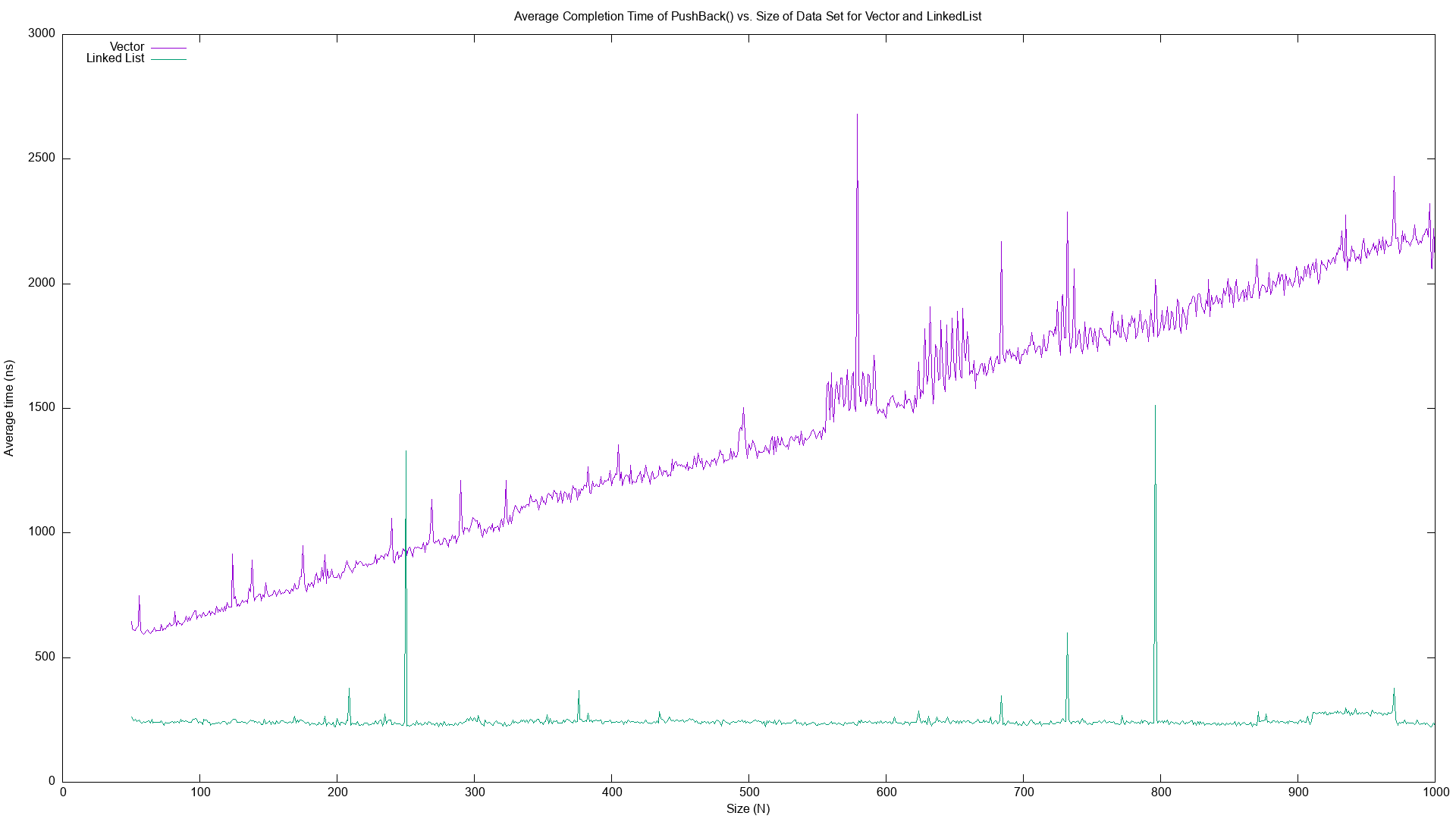
1. PushFront()



The time complexity of the vector is O(n), so the average completion time grows linearly.

The time complexity of the linked list is O(1), so the average completion time is almost constant.

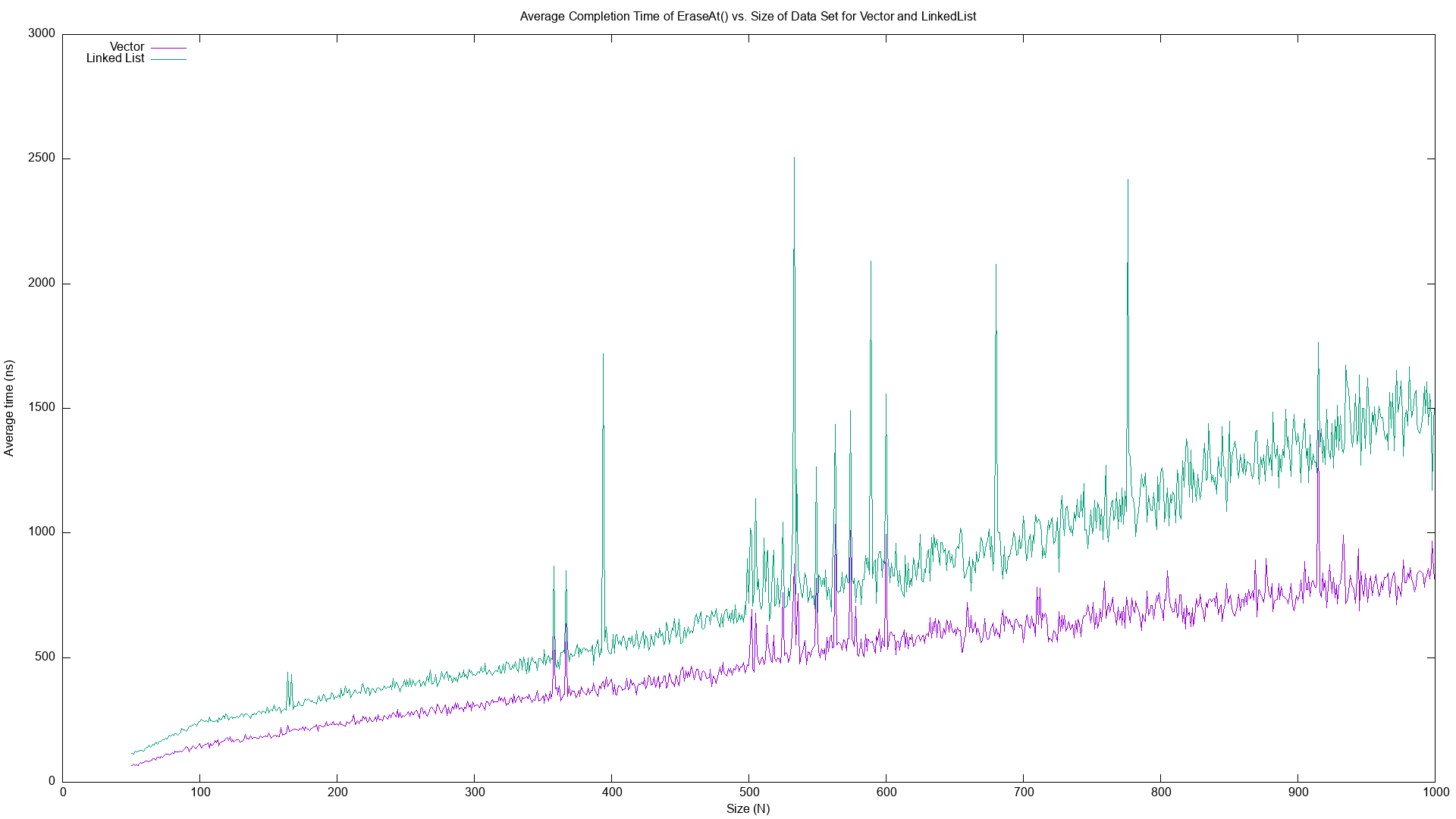
1. PushBack()



The time complexity of the vector is O(n), so the average completion time grows linearly.

The time complexity of the linked list is O(1), so the average completion time is almost constant.

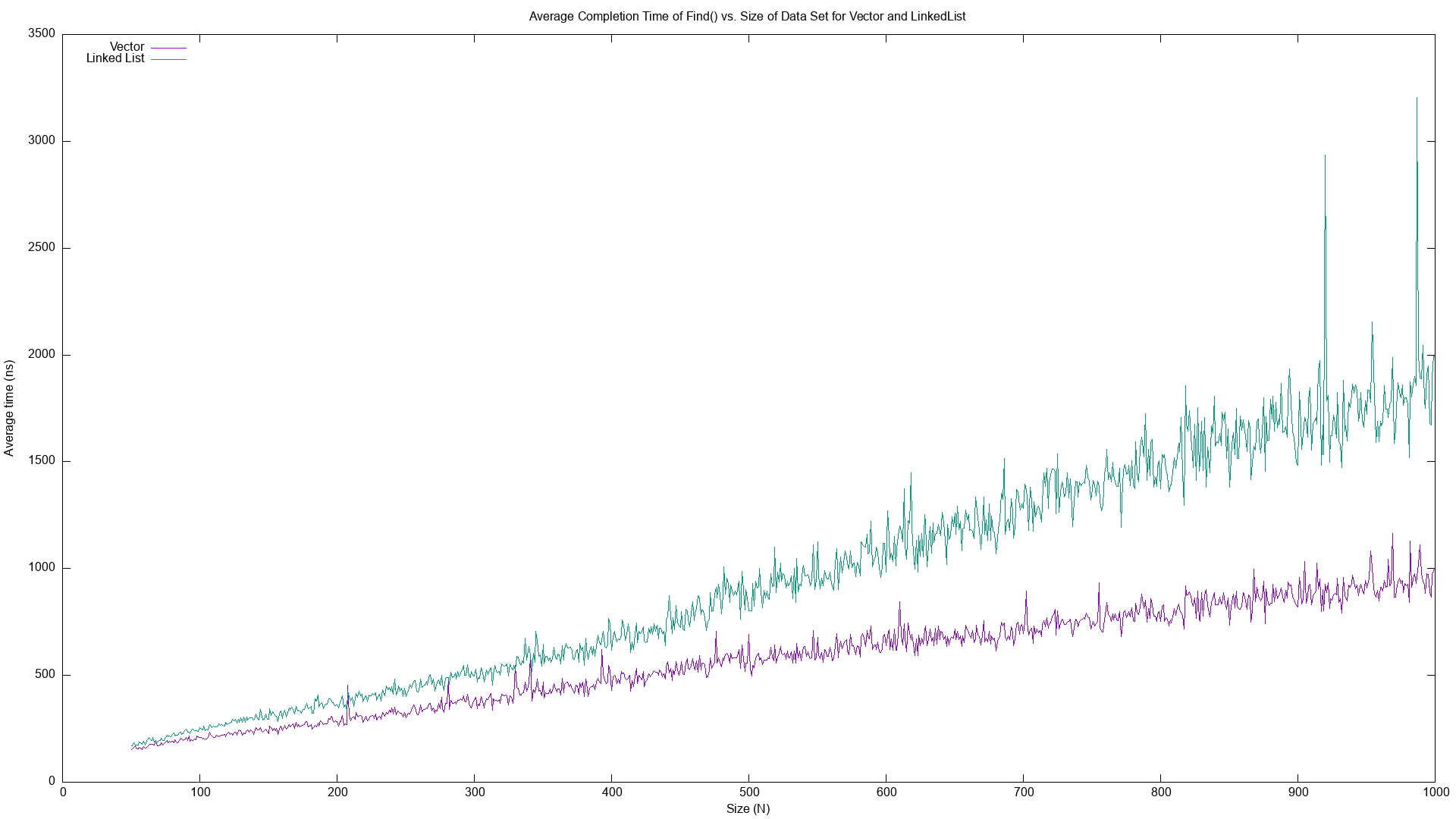
1. EraseAt()



The time complexity of the vector is O(n), so the average completion time grows linearly.

The time complexity of the linked list is O(n), so the average completion time grows linearly.

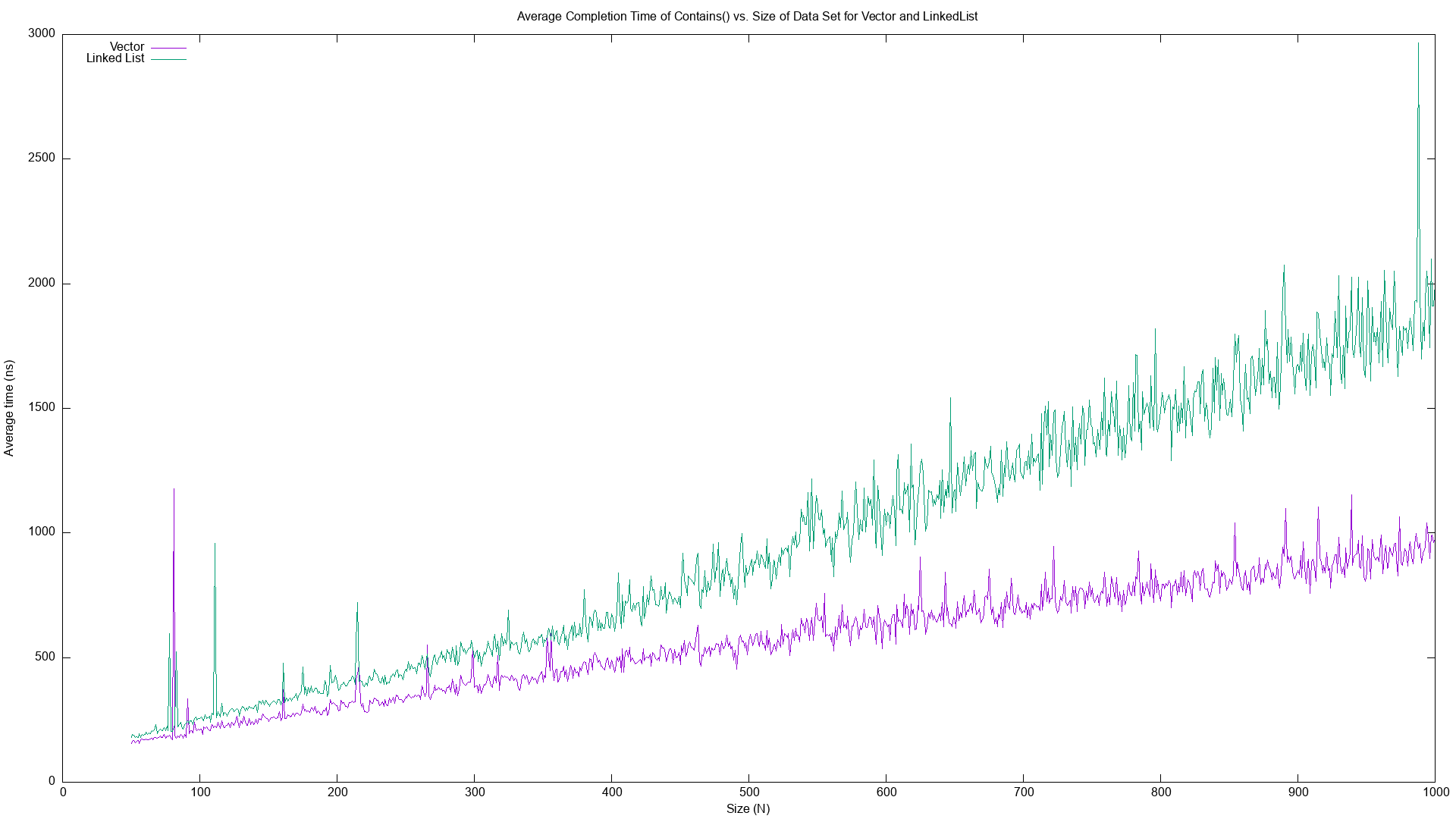
1. Find()



The time complexity of the vector is O(n), so the average completion time grows linearly.

The time complexity of the linked list is O(n), so the average completion time grows linearly.

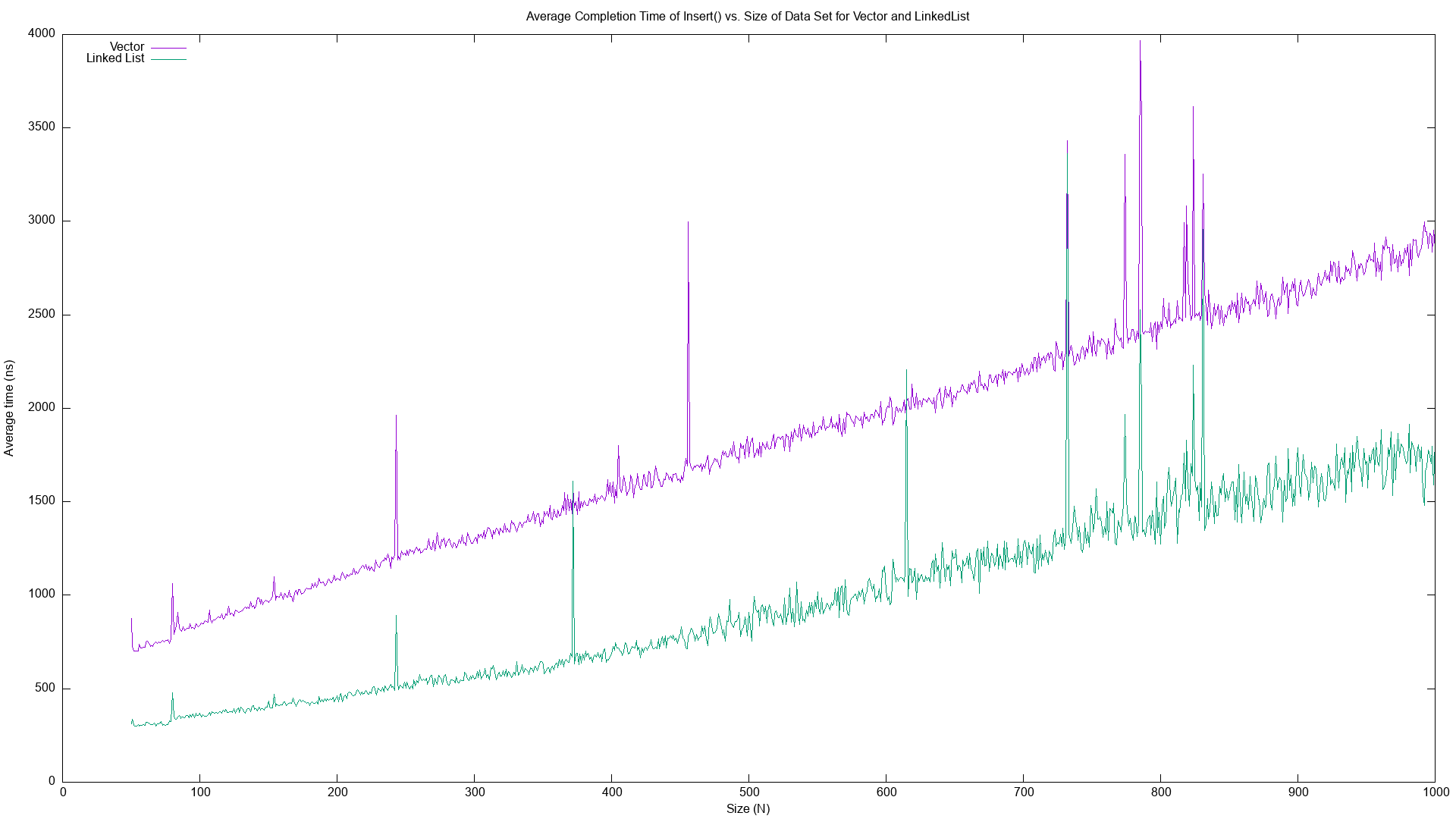
1. Contains()



The time complexity of the vector is O(n), so the average completion time grows linearly.

The time complexity of the linked list is O(n), so the average completion time grows linearly.

1. Insert()



The time complexity of the vector is O(n), so the average completion time grows linearly.

The time complexity of the linked list is O(n), so the average completion time grows linearly.