Comparison of Time Complexity: Linear versus Binary Search

Linear Search:

* Best Case: O(1)
* Average Case: O(n)
* Worst Case: O(n)

Binary Search:

* Best Case: O(1)
* Average Case: O(log n)
* Worst Case: O(log n)

Explanation:

Linear search scans each element in the array from start to end until it finds the target or reaches the end. Worst case: checks all elements. O(n) time complexity.  
  
Binary search requires sorted data it repeatedly divides a sorted array in half to find the target. Each step cuts the search space by half. O(log n) time complexity.

Suitable Algorithm for the Given Platform

In the Java program:

* The user enters a list of items.
* The list is sorted before using binary search.
* Both linear and binary search methods are used to search for a product ID.

If the dataset is small or only one search is needed, linear search is more suitable. It doesn’t require sorting and works well for small datasets.

If multiple searches are expected on the same dataset, binary search is more efficient. It requires the list to be sorted first (O(n log n)), but once sorted, each search only takes O(log n) time.

Conclusion:  
Use linear search when the dataset is small or only one search is needed.  
Use binary search when the dataset is large or when many searches are required after sorting the data once.