**Algorithm: Binary Search**

1. **Start**
2. **Given a sorted list/array and a target value.**
3. **Set the left pointer (low) to the first index of the list.**
4. **Set the right pointer (high) to the last index of the list.**
5. **Repeat steps 6-9 while low is less than or equal to high.**
6. **Calculate the middle index as (low + high) / 2, rounding down if necessary.**
7. **If the middle element is equal to the target value, return the middle index.**
8. **If the middle element is greater than the target value, update high to be (middle - 1).**
9. **If the middle element is less than the target value, update low to be (middle + 1).**
10. **If the target value is not found, return -1.**
11. **Stop.**

* **The binary search algorithm is an efficient algorithm used to search for a target value within a sorted list or array. It follows a divide-and-conquer approach by repeatedly dividing the search space in half.**
* **The binary search algorithm reduces the search space in half at each iteration, making it highly efficient. It leverages the fact that the list is sorted to determine which half of the remaining search space contains the target value, eliminating the need to search through the entire list.**
* **It's important to note that the binary search algorithm assumes a sorted list as input. If the list is unsorted, you will need to sort it first before applying the binary search algorithm.**