**Binary Search for Insert Position**

Binary Search for Insert Position is an algorithmic technique used to find the position where a given element should be inserted in a sorted array to maintain the array's sorted order. If the element is present in the array, it returns the index of the element; otherwise, it returns the index where the element should be inserted.

1. **Problem solving:**

**Initialize Pointers:**

Initialize left and right pointers for binary search.

**Binary Search Loop:**

While left pointer is less than or equal to the right pointer, do the following:

Calculate mid-point.

If the element at mid-point is equal to the target, return mid.

If the element at mid-point is less than the target, update left pointer to mid + 1.

If the element at mid-point is greater than the target, update right pointer to mid - 1.

**Return Insert Position:**

If the target is not found, return the left pointer as the insert position.

1. **Flow chart**

**start**

**Initalize pointer**

**left = 0, right = nums.length - 1**

**Splitting**

**Calculate mid point**

**Length calculation**

**output**

**End**

**While left <= right**

**Return target**

**If nums[mid]**

**== target**

**YES**

**No**

**If nums[mid]**

**< target**

**YES**

**Update left = mid + 1**

**No**

**Update left = mid + 1**

**If nums[mid]**

**> target**

**No**

**YES**

**Return Insert Position**

**End**

1. **Output**

