

Binary Search

The Binary Search Algorithm below finds the index of a value in an array of integers sorted in ascending order as follows:

1. Set `left` and `right` to the minimum and maximum indexes of `elements` respectively.
2. Loop until `target` is found, or `target` is determined not to be in `elements` by doing the following for each iteration:
 - a. Set `middle` to the index of the middle item in `elements[left] ... elements[right]` inclusive.
 - b. If `target` would have to be in `elements[left] ... elements[middle - 1]` inclusive, then set `right` to the maximum index for that range.
 - c. Otherwise, if `target` would have to be in `elements[middle + 1] ... elements[right]` inclusive, then set `left` to the minimum index for that range.
 - d. Otherwise, return `middle` because `target == elements[middle]`.
3. Return `-1` if `target` is not contained in `elements`.

```

/**
 * Find the index of a value in an array of integers sorted in ascending order.
 *
 * @param elements an array containing the items to be searched.
 * Precondition: items in elements are sorted in ascending order.
 * @param target the item to be found in elements.
 * @return an index of target in elements if target found;
 *         -1 otherwise.
 */
public static int binarySearch(int[] elements, int target)
{
    int left = 0;
    int right = elements.length - 1;
    while (left <= right)
    {
        int middle = (left + right) / 2;
        if (target < elements[middle])
        {
            right = middle - 1;
        }
        else if (target > elements[middle])
        {
            left = middle + 1;
        }
        else
        {
            return middle;
        }
    }
    return -1;
}

```