**BINARY SEARCH ALGORITHM:**

Binary search is **an efficient algorithm for finding an item from a sorted list of items**. It works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to just one.

[**Worst complexity**](https://www.google.com/search?q=binary+search+algorithm+worst+complexity&stick=H4sIAAAAAAAAAOPgE-LQz9U3MEzOrdIyzk620s8uiM8p1y_OLyrJzEuPT8xJzy_KLMnItSrPLyouiU_Ozy3ISa3ILKmML85ILEpNWcSqkZSZl1hUqVCcmliUnKEA16EA1qGA0AEApfZGGmwAAAA&sa=X&ved=2ahUKEwj-6_HStbX0AhXRSmwGHa-uDK8Q6BMoAHoECE0QAg)**:**O(log n)

[**Average complexity**](https://www.google.com/search?q=binary+search+algorithm+average+complexity&stick=H4sIAAAAAAAAAOPgE-LQz9U3MEzOrdIyzU620s8uiM8p1y_OLyrJzEuPT8xJzy_KLMnItUosSy1KTE-NT87PLchJrcgsqYwvzkgsSk1ZxKqVlJmXWFSpUJyaWJScoQDXowDVo4DQAwDiVtOVcAAAAA&sa=X&ved=2ahUKEwj-6_HStbX0AhXRSmwGHa-uDK8Q6BMoAHoECEwQAg)**:**O(log n)

[**Best complexity**](https://www.google.com/search?q=binary+search+algorithm+best+complexity&stick=H4sIAAAAAAAAAOPgE-LQz9U3MEzOrdIyyk620s8uiM8p1y_OLyrJzEuPT8xJzy_KLMnItUpKLS6JT87PLchJrcgsqYwvzkgsSk1ZxKqelJmXWFSpUJyaWJScoQDXoADSoIDQAACYYGBOagAAAA&sa=X&ved=2ahUKEwj-6_HStbX0AhXRSmwGHa-uDK8Q6BMoAHoECEsQAg)**:**O(1)

[**Space complexity**](https://www.google.com/search?q=binary+search+algorithm+space+complexity&stick=H4sIAAAAAAAAAD3KMQ6AIAwAwMnEycEXMKoLuvoZUmsDRCqkkCg-x5e6efO1Xd9q1vOC_Ezjgas-kgmXzlGKP62BYKP44nhl4ijVZAdC-9sMmz9BqsoEgk79T-UESAojp0C3L_UDXlIy_mIAAAA&sa=X&ved=2ahUKEwj-6_HStbX0AhXRSmwGHa-uDK8Q6BMoAHoECEMQAg)**:**O(1)

**ALGORITHM:**

Binary\_Search(a, lower\_bound, upper\_bound, val) // 'a' is the given array, 'lower\_ bound' is the index of the first array element, 'upper\_bound' is the index of the last  array element, 'val' is the value to search

Step 1: set beg = lower\_bound, end = upper\_bound, pos = - 1

Step 2: repeat steps 3 and 4 while beg **<**=end

Step 3: set mid = (beg + end)/2

Step 4: if a[mid] = val

set pos = mid

print pos

go to step 6

else if a[mid] **>** val

set end = mid - 1

else

set beg = mid + 1

[end of if]

[end of loop]

Step 5: if pos = -1

print "value is not present in the array"

[end of if]

Step 6: exit