The Binary Search Algorithm

KV, Winter 2020

The binary search algorithm works best on data sets that are stored in vectors, i.e., a **linear data structure with random access** to its elements. The values in the vector must be stored in **order, either ascending (default) or descending**.

Binary search is will not work on unsorted vectors …

The Binary Search Algorithm:

Determine whether target value X is included in a sorted Vector of inputs.
Start out with a vector V and its lowest and highest index.

low = 0
high = size of vector – 1

As long as low is no greater than high, repeat

1. Find the element in the middle of the vector; the middle element is at index

   mid = (low + high)/2 (why?)
   midval = V[mid]

2. Compare midval and X:

   If midval == X, target X is found; return TRUE
   If midval < X, keep searching for X in "right half"; see step 3.a.
   If midval > X, keep searching for X in "left half"; see step 3.b.

3. Depending on case:
   a. Update: low = mid + 1
   b. Update: high = mid -1

   … next iteration …

4. This step will be reached when low > high. Target X is not among the inputs in V. Return FALSE.

Performance of the Binary Search Algorithm:

- The target value is found with **O(log N**) search effort. (Best case, average case, and worst case); what are the cases?

    - Best case: target is equal to the middle element
    - Average case: target equal to the middle element at some level of halves
    - Worst case: the target value is not among the input values.