4. Searching Algorithms :: Recursive Binary Search Example 2 (key not found)

Let's trace an example to see what happens when the key is not found. Once gain, let $pList$ = { 3, 41, 9, 11, 13, 17, 19, 23, 29, 31, 37 } and let $pKey = 41$.

0. int index = recursiveBinarySearch(list, 41, 0, list.size() - 1)

1. recursiveBinarySearch (pList, pKey = 41, pLow = 0, pHigh = 10)
   Check base case: $0 > 10$ is false
   Compute middle: middle = 5
   Compare pKey to $\text{pList}_{\text{middle}}$
   $41 > 17$ so call recursiveBinarySearch(pList, 41, 6, 10) **and return what it returns**

   2. recursiveBinarySearch (pList, pKey = 41, pLow = 6, pHigh = 10)
      Check base case: $6 > 10$ is false
      Compute middle: middle = 8
      Compare pKey to $\text{pList}_{\text{middle}}$
      $41 > 29$ so call recursiveBinarySearch(pList, 41, 9, 10) **and return what it returns**

      3. recursiveBinarySearch (pList, pKey = 41, pLow = 9, pHigh = 10)
         Check base case: $9 > 10$ is false
         Compute middle: middle = 9
         Compare pKey to $\text{pList}_{\text{middle}}$
         $41 > 31$ so call recursiveBinarySearch(pList, 41, 10, 10) **and return what it returns**

4. Searching Algorithms :: Recursive Binary Search Example 2 (continued)

4. recursiveBinarySearch (pList, pKey = 41, pLow = 10, pHigh = 10)
   Check base case: 10 > 10 is false
   Compute middle: middle = 10
   Compare pKey to $pList_{middle}$
   41 > 37 so call recursiveBinarySearch(pList, 41, 11, 10) **and return what it returns**

   5. recursiveBinarySearch (pList, pKey = 41, pLow = 11, pHigh = 10)
      Check base case: 11 > 10 is true **so return -1 (*not found*) back to Step 4**

4. recursiveBinarySearch (pList, pKey = 41, pLow = 10, pHigh = 10)
   Check base case: 10 > 10 is false
   Compute middle: middle = 10
   Compare pKey to $pList_{middle}$
   41 > 37 call recursiveBinarySearch(pList, 41, 11, 10) **returned -1 so return -1 back to Step 3.**

3. recursiveBinarySearch (pList, pKey = 41, pLow = 9, pHigh = 10)
   Check base case: 9 > 10 is false
   Compute middle: middle = 9
   Compare pKey to $pList_{middle}$
   41 > 31 so call recursiveBinarySearch(pList, 41, 10, 10) **returned -1 so return -1 back to Step 2.**

2. recursiveBinarySearch (pList, pKey = 41, pLow = 6, pHigh = 10)
   Check base case: 6 > 10 is false
   Compute middle: middle = 8
   Compare pKey to $pList_{middle}$
   41 > 29 so call recursiveBinarySearch(pList, 41, 9, 10) **returned -1 so return -1 back to Step 1.**

4. Searching Algorithms :: Recursive Binary Search Example 2 (continued)

1. recursiveBinarySearch (pList, pKey = 41, pLow = 0, pHigh = 10)
   Check base case: $0 > 10$ is false
   Compute middle: middle = 5
   Compare pKey to $\text{pList}_{\text{middle}}$
   $41 > 17$ so call recursiveBinarySearch(pList, 41, 6, 10) **returned -1 so return -1 back to Step 0.**

0. int index = recursiveBinarySearch(list, 41, 0, list.size() - 1) **which returned -1 so assign -1 to *index*.**