3. Searching Algorithms :: Recursive Binary Search Example 1 (key is found)

Let's trace an example. Let $pList$ = { 3, 7, 9, 11, 13, 17, 19, 23, 29, 31, 37 } and $pKey = 7$.

0. int index = recursiveBinarySearch(list, 7, 0, list.size() - 1)

1. recursiveBinarySearch (pList, pKey = 7, pLow = 0, pHigh = 10)
   Check base case: $0 > 10$ is false
   Compute middle: middle = 5
   Compare pKey to $\text{pList}_{\text{middle}}$
   $7 < 17$ so call recursiveBinarySearch(pList, 7, 0, 4) **and return what it returns**

   2. recursiveBinarySearch (pList, pKey = 7, pLow = 0, pHigh = 4)
      Check base case: $0 > 4$ is false
      Compute middle: middle = 2
      Compare pKey to $\text{pList}_{\text{middle}}$
      $7 < 9$ so call recursiveBinarySearch(pList, 7, 0, 1) **and return what it returns**

      3. recursiveBinarySearch (pList, pKey = 7, pLow = 0, pHigh = 1)
         Check base case: $0 > 1$ is false
         Compute middle: middle = 0
         Compare pKey to $\text{pList}_{\text{middle}}$
         $7 > 3$ so call recursiveBinarySearch(pList, 7, 1, 1) **and return what it returns**

3. Searching Algorithms :: Recursive Binary Search Example 1 (continued)

4. recursiveBinarySearch (pList, pKey = 7, pLow = 1, pHigh = 1)
   Check base case: $1 > 1$ is false
   Compute middle: middle = 1
   Compare pKey to $\text{pList}_{\text{middle}}$
   $7 == 7$ so **return middle = 1 back to Step 3 (*at this point we stop recursing*)**

3. recursiveBinarySearch (pList, pKey = 7, pLow = 0, pHigh = 1)
   Check base case: $0 > 1$ is false
   Compute middle: middle = 0
   Compare pKey to $\text{pList}_{\text{middle}}$
   $7 > 3$ so call recursiveBinarySearch(pList, 7, 1, 1) **which returned 1 so return 1 back to Step 2.**

2. recursiveBinarySearch (pList, pKey = 7, pLow = 0, pHigh = 4)
   Check base case: $0 > 4$ is false
   Compute middle: middle = 2
   Compare pKey to $\text{pList}_{\text{middle}}$
   $7 < 9$ so call recursiveBinarySearch(pList, 7, 0, 1) **which returned 1 so return 1 back to Step 1.**

1. recursiveBinarySearch (pList, pKey = 7, pLow = 0, pHigh = 10)
   Check base case: $0 > 10$ is false
   Compute middle: middle = 5
   Compare pKey to $\text{pList}_{\text{middle}}$
   $7 < 17$ so call recursiveBinarySearch(pList, 7, 0, 4) **which returned 1 so return 1 back to Step 0.**

0. int index = recursiveBinarySearch(list, 7, 0, list.size() - 1) **which returned 1 so assign 1 to *index*.**