Python - Recursion

Recursion allows a function to call itself. Fixed steps of code get executed again and again for new values. We also have to set criteria for deciding when the recursive call ends. In the below example we see a recursive approach to the binary search. We take a sorted list and give its index range as input to the recursive function.

Binary Search using Recursion

We implement the algorithm of binary search using python as shown below. We use an ordered list of items and design a recursive function to take in the list along with starting and ending index as input. Then, the binary search function calls itself till find the searched item or concludes about its absence in the list.

Example

```
def bsearch(list, idx0, idxn, val):
   if (idxn < idx0):
      return None
   else:
      midval = idx0 + ((idxn - idx0) // 2)
# Compare the search item with middle most value
   if list[midval] > val:
      return bsearch(list, idx0, midval-1,val)
   else if list[midval] < val:
      return bsearch(list, midval+1, idxn, val)
   else:
      return midval
list = [8,11,24,56,88,131]
print(bsearch(list, 0, 5, 24))
print(bsearch(list, 0, 5, 51))
```

Output

When the above code is executed, it produces the following result −

```
2
None
```

Print Page