Toop invariant: The element searchnum is not present in list [oleft-1]
on Balone the first Heration of the loop, left=0 and
Initialization: Defore the first iteration of the loop, left=0 and right=n-1. Searchnum & list [0] and & list [n,n-1], trivially, because the ranges do not make sense.
hight = 1/1 2000 to the
because the ranges as the loop, the
10,00000 (410)
Maintenance: Suppose, as to describe the last [Oleft-] and were compute Jose invariant holds. i. searchnum & list [Oleft-] and were compute Joseph invariant holds. i. searchnum & list [right], n-1]. Now, let Jeft = right holds. In Jine 5, we compute & list [right], n-1]. Now, let Jeft = right holds. In Jine 5, we compute & list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum, then it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a list [right] / 2. If Jist [middle] & searchnum it must be a
& list [right + 1, n-1]. Now, let I widdle I searchnum, then it must be
Josep invariant hours. Now, let Jeft=pight holds. In some of must be & list [right], n-1]. Now, let Jeft=pight holds. In some of must be & list [right]/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be middle = (Jeft+right)/2. If List [middle] & searchnum, then it must be present in list[].
The case was in the initialize state
1st midalet 1 of temps in asconding the
Before the start of the rest iteration, and its easy to see the start of the rest iteration, and its easy to see the start of the rest iteration, and its easy to see the start of the rest iteration, and its easy to see the search number of sear
before the run of list [0left-] and searchnung
Before the start of the read of Combine our maintenance of search num & list [0left-1] (Combine our maintenance) and search num & list [0left-1] and search num & of prev iteration search num & list [old-left, old-middle] which we proved above) and search num & list [old-left, old-middle] which we proved above) and search num & list [old-left, old-middle] which we proved above) and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above above and search num & list [old-left, old-middle] which we proved above above and search num & list [old-left, old-middle] which we proved above and search num & list [old-left, old-middle] which we proved above above are list [old-left, old-left, old-middle] which we proved above are list [old-left, old-left, old-lef
of previous old-middle which we prove invariant holds in this
List Orania tot n-1 (prev assumption).
will hold if list [middle]
case. Similarly prove that loop invariant out
List [oright + 1, n-1] (prev aramption) case. We can similarly prove that loop invariant will hold if list[middle]) Searchnum Lion will terminate in two cases:
Searchnum Termination: The function will terminate in two cases: Termination: The function will terminate in two cases: 1) If list [middle] = searchnum. So, then searchnum has been found, and 1) If list [middle] = searchnum. So, then searchnum has been found, and 1) If list [middle] = searchnum. So, then searchnum has been found, and
Termination of the samphium. So, then searchnum has bear so
we peturn the In loft right is reached. Just before this oteration,
2) When the conditions widdle (left right)/2.
1) If list missed. The post middle we peturn the post middle reached. Just before this iteration, 2) When the condn. loft/right is reached. Just before this iteration, 2) When the condn. loft/right is reached. Just before this iteration, Left/right would hold. middle (loft-right)/2. Left/left-right .: then left/right (contradiction) To left left-right .: then left/right (contradiction)
Jeft = right would hold middle [west of the contradiction] If, left > left + right = right = right = right = right = left = left + right = right = left + right = left + right = right = right = left + right = rig