

Insert Sort

Insert Sorting an array is $O(n^2)$. We traverse the array and at a typical stage we have the array slice $A[1..k-1]$ sorted and $A[k..n]$ unsorted. We make progress and maintain the array slice $A[1..k]$ sorted by inserting $A[k]$ into its proper position. We then get $A[1..k]$ sorted and $A[k+1..n]$ remains unsorted. We use Binary Search to determine the proper location for the insertion of $A[k]$.

Insert Sort is an in-place sort and inserting an item requires that many items are shifted 'right'.

```
sort(a:ARRAY[G]; low,high:INTEGER) is
  require  -- a[low..high] is non-empty
    Range: low <= high and a.lower <= low and high <= a.upper
    Non_Trivial: a /= void and then a.count > 0
  local
    k:INTEGER
    bs: BINARY_SEARCH_G[G]
  do
    from
      !!bs
      k := low+1
    until
      k > high
    loop
      bs.Search(a,low,k-1,a.item(k))
      insert(a.item(k), bs.index+1, a, low, k-1)
      k:=k+1
    end
  ensure
    Sorted: Is_Ordered(a,low,high)
end—sort
```

The attribute, index, returns the proper location even if $A[k]$ is already in $A[1..k-1]$.

The procedure, insert, shifts the array slice $A[i..high]$ to the 'right' by one location and then puts the item x at index i .

```

insert(x:G; i:INTEGER; a:ARRAY[G]; low,high:INTEGER) is
  -- insert x at position i in array a.
  require
    Range: low <= i and i <= high+1
  local
    k :INTEGER
  do
    from
      k:=high+1
    until
      k = i
    loop
      a.put(a.item(k-1),k)
      k := k-1
    end
    a.put(x,i)
  end—insert

```

```

Is_Ordered(a:ARRAY[G]; L,H:INTEGER):BOOLEAN is
  -- check whether array is ordered
  require  -- a[L..H] is non-empty
    Range: L <= H and a.lower <= L and H <= a.upper
    Non_Trivial: a /= void and then a.count > 0
  local
    i, j : INTEGER
  do
    from
      i := L
      j := H
    until
      i = j
    loop
      if a.item(i) <= a.item(i+1) then
        i := i+1
      else
        j := i
      end
    end
    result := i = H
  ensure
    -- (All k | L <= k < H : a@k <= a@(k+1))
  end—Is_Ordered

```