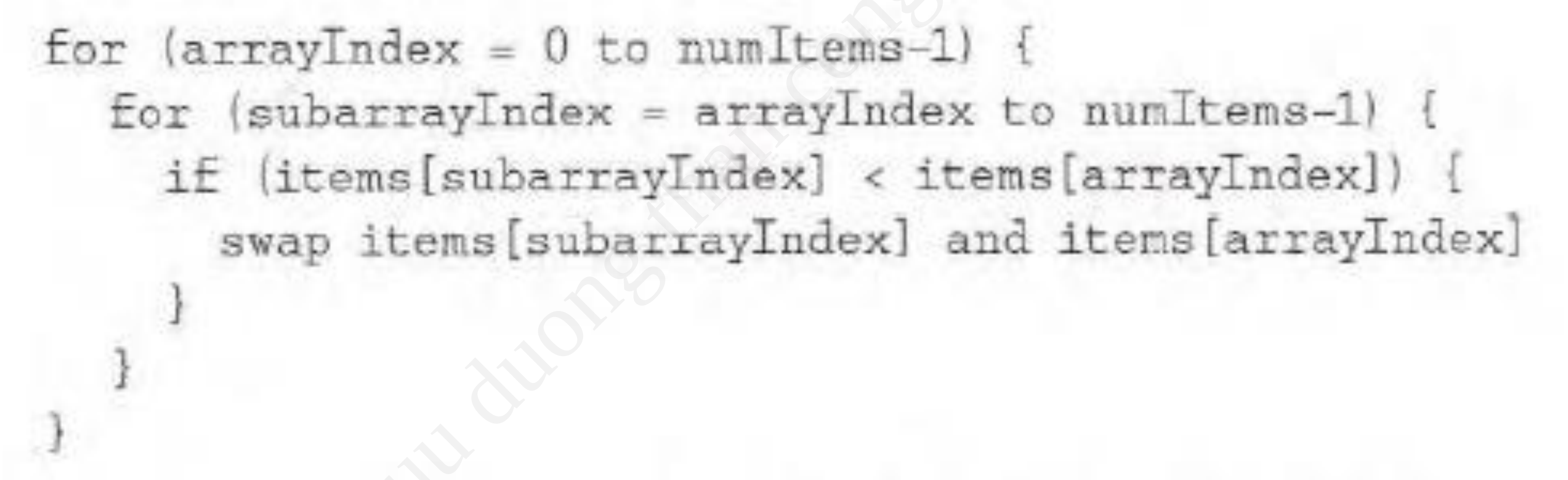
Recursion and Advanced Algorithms

Ho Dac Hung

Selection Sort

- Sorting is the process of putting items in a designated order, either from low to high or high to low.
- The selection sort algorithm starts by finding the lowest item in a list and swapping it with the first. Next, the lowest item among items 2 through the last is found and swapped with item 2. This process in continued until the last item is reached, at which point all the items are stored.

Selection Sort



```
for (arrayIndex = 0 to numItems-1) {
  for (subarrayIndex = arrayIndex to numItems-1) {
    if (items[subarrayIndex] < items[arrayIndex]) {
      swap items[subarrayIndex] and items[arrayIndex]
    }
  }
}
```

Sorting Objects

- Relational operator cannot be used to compare objects. Objects use methods of their class to determine if one object is greater than, less then, or equal to another. The equals() method in a class is used to determine equality. For determining order, the compareTo() method is used.
- Objects that are to be sorted must have a class that implements the Comparable interface.

Insertion Sort

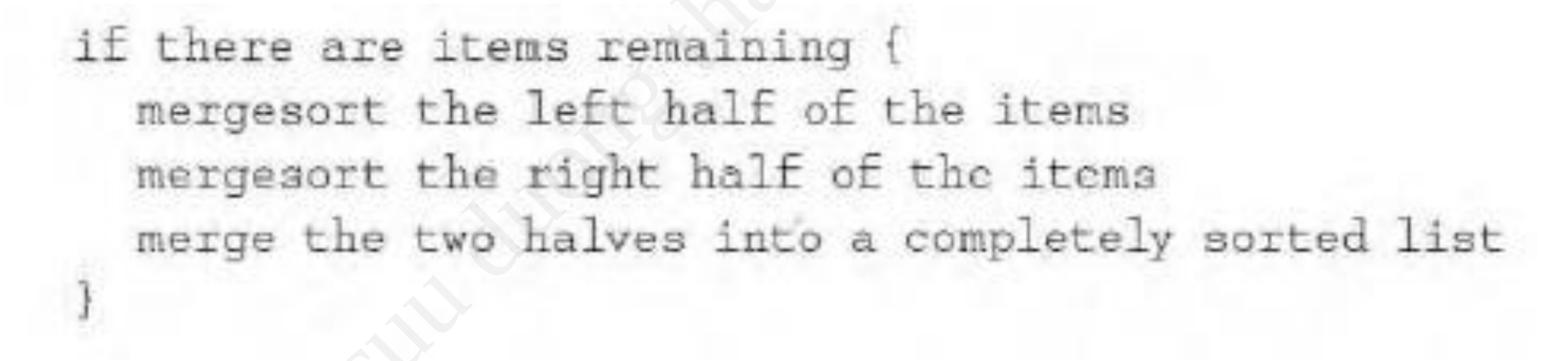
- An insertion sort starts by sorting the first two items in a list. This sort is performed by shifting the first item into the second spot if the second item belongs in the first spot. Next, the third item is properly inserted within the first three items by again shifting items into their approriate position to make room for the moved item. This process is repeated for the remaining elements.

Recursion

- A method can call itself. This process is called recursion and the calls are referred to as recursive calls.
- Some solution would cause infinite recursion. To prevent these, a recursive solution must have a base case that required no recursion.

Mergesort

- The mergesort algorithms takes a “divide and conquer” approach to sorting.

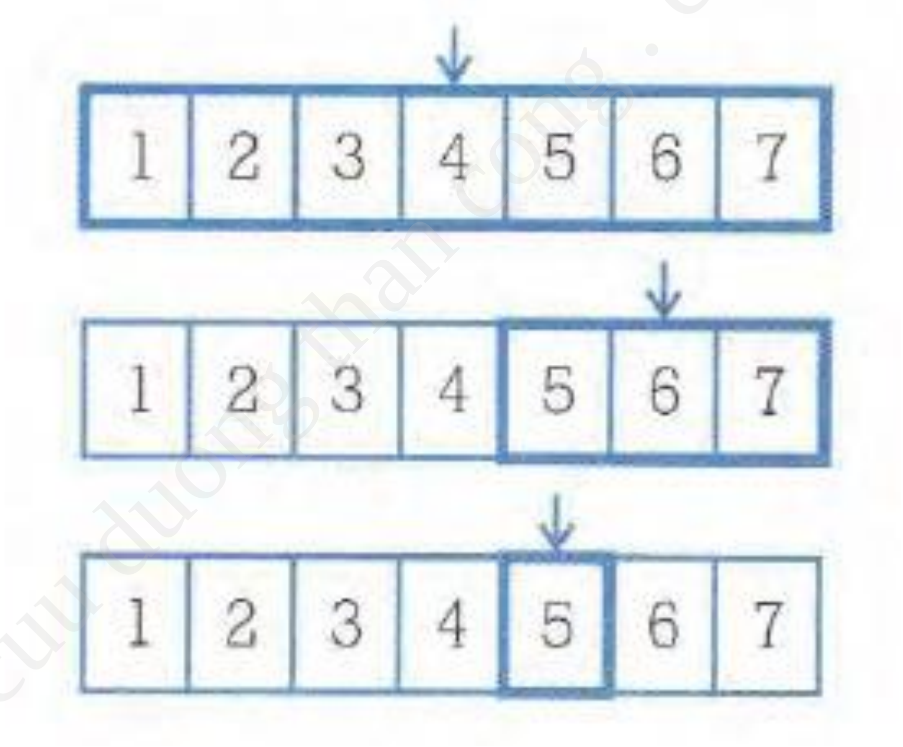


```
if there are items remaining {
   mergesort the left half of the items
   mergesort the right half of the items
   merge the two halves into a completely sorted list
}
```

Binary Search

- Like the mergesort algorothm, the binary search algorithm also taks a “divide and conquer” approach. It works by examining the middle item of an array sorted from low to high, and determining if this is the item sought, or if the item sought is above or below this middle term.

Binary Search



Depth-First Searching

- The depth-first searching algorithm works by searching from a given starting position, processing that position, and then recursively searching from all adjancent positions.