07.10 Virtual Lecture Notes

For a selection sort putting integers in ascending order, you would look through the array and find the largest element. Then you swap it with the last element in the array. If the element is already in the last position, you leave it alone. You then repeat the process but ignore the last element; you put the largest into the last position in this smaller array area. If the element is already in the last position, you leave it alone. This is repeated again, but this time you ignore the last two elements.

You repeat this process until you are down to the last two elements. If the one on the left is the largest, then you swap; otherwise, these are in order and you are done.

Let's look at a selection sort applied to the integer array `myInts` that has been properly declared and initialized:

```
int i;
int k;
int posMax;
int temp;

for( i = myInts.length - 1; i >= 0; i-- )
{
   // find largest element in the i elements
   posMax = 0;
   for( k = 0; k <= i; k++ )
   {
      if( myInts[ k ] > myInts[ posMax ] )
         posMax = k;
   }
   // swap the largest with the position i
   // now the item is in its proper location
   temp = myInts[ i ];
   myInts[ i ] = myInts[posMax ];
   myInts[ posMax ] = temp;
}
```

Notice that the outer `for` loop will ensure that we keep looking at smaller and smaller sections of the array. First, we find the largest item in the search array and store its index in `posMax`. Then we swap the last position in the search area with the one at position `posMax`. That element is now in its proper location. The next time through, `i` will be one less; therefore, we will not look at that last position again (as it has the value it should). The list will be sorted ascending, when we are done.

If we wanted to sort in descending order, we would just change the > to a < in the `if` statement of the `for` loop that searches for the largest element.

Now we apply the selection sort algorithm to our list of houses to sort in ascending order by cost:

```
public static void selectionSort(HouseListing[] source)
{
   int i;
   int k;
   int posMax;
   HouseListing temp;

   for( i = source.length - 1; i >= 0; i-- )
   {
      // find largest element in the i elements
      posMax = 0;
      for( k = 0; k <= i; k++ )
      {
         if( source[ k ].getCost() > source[ posMax
].getCost() )
            posMax = k;
      }
      // swap the largest with the position i
      // now the item is in its proper location
      temp = source[ i ];
      source[ i ] = source[posMax ];
      source[ posMax ] = temp;
   }
}
```

Notice that the swap looks similar to that used in bubble sort. It should, as the swap procedure is always the same when switching two elements in an array. Use of a temporary variable is key so values are not

lost or overwritten.

If we wanted to sort the array in descending order, we just switch the > with a < in the `if` statement. Try it!

```
if( source[ k ].getCost() < source[ posMax ].getCost() )
```

Run the `TestListing3` program and observe the output. Notice, the selection sort method is sorting by cost in ascending order. Were you successful changing the sort to descending order? How about setting up another method to sort by city? Give it a try by copying the current method and make the needed adjustments. The sort method provides debugging print statements. Uncomment these so you can see a bit more detail related to each pass through the sort.

Print