# 17.03 Virtual Lecture Notes

Let us assume we have the same array setup as with the bubble sort and insertion sort. We would like to apply selection sort to the array to sort by cost, like we did for the previous sort algorithms.

Remember that in selection sort, you go 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.

So first let 'us apply this to an integer array **`myInts`**.

```
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 it to our list of houses:

```
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.

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

```
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;
     }
  }
```

That is all there is to a selection sort.

- Download the TestListing4.java file to your module 17 demo programs directory and open it.
- Run the file and make sure you understand it before you continue.