Insertion Sort and Selection Sort

Selection sort and insertion sort are two simple sorting algorithms; they are more often efficient than bubble sort, though they aren't the top of the class algorithmically.

Selection sort

Selection sort is the most conceptually simple of all the sorting algorithms. It works by selecting the smallest (or largest, if you want to sort from big to small) element of the array and placing it at the head of the array. Then the process is repeated for the remainder of the array; the next largest element is selected and put into the next slot, and so on down the line.

Because a selection sort looks at progressively smaller parts of the array each time (as it knows to ignore the front of the array because it is already in order), a selection sort is slightly faster than bubble sort, and can be better than a modified bubble sort.

Here is the code for a simple selection sort:

```
        for(int x=0; x<n; x++)

        {

                int index_of_min = x;

                for(int y=x; y<n; y++)

                {

                        if(array[index_of_min]>array[y])

                        {

                                index_of_min = y;

                        }

                }

                int temp = array[x];
```

```
            array[x] = array[index_of_min];

            array[index_of_min] = temp;

        }
```

The first loop goes from 0 to n, and the second loop goes from x to n, so it goes from 0 to n, then from 1 to n, then from 2 to n and so on. The multiplication works out so that the efficiency is n*(n/2), though the order is still O(n^2).

Insertion Sort

Insertion sort does exactly what you would expect: it inserts each element of the array into its proper position, leaving progressively larger stretches of the array sorted. What this means in practice is that the sort iterates down an array, and the part of the array already covered is in order; then, the current element of the array is inserted into the proper position at the head of the array, and the rest of the elements are moved down, using the space just vacated by the element inserted as the final space.

Here is an example: for sorting the array the array 52314 First, 2 is inserted before 5, resulting in 25314 Then, 3 is inserted between 2 and 5, resulting in 23514 Next, one is inserted at the start, 12354 Finally, 4 is inserted between 3 and 5, 12345

While the code to do this is fairly straightforward, I'll leave it as an exercise to the reader.

Previous: Bubble Sort
Next: Heap Sort