

CSE 3010 – Data Structures & Algorithms

Lecture #26

What will be covered today

- Comparison of simple sorting techniques
- Introduction to divide and conquer algorithms
- Examples of divide and conquer sorting algorithms

Comparison of simple sorting techniques

Sorting Technique	How Does It Work
Bubble sort	<ul style="list-style-type: none">• Each iteration goes through the entire array and swaps the elements when found not in order• At the end of each iteration the largest element reaches its rightful position in the array
Selection sort	<ul style="list-style-type: none">• Select the smallest element in the array in each iteration• Place the smallest element in its rightful position
Insertion sort	<ul style="list-style-type: none">• Implicitly a sorted sub list is maintained• Element to be inserted in the sorted sub-list finds its appropriate place and inserts itself• Array is searched sequentially• Unsorted items are moved• Unsorted items are inserted into the sorted sub-list, within the same array

Bubble sort – How does it work?

321	214	66	66	66	32	14
214	66	214	102	32	14	32
66	321	102	32	14	66	66
600	102	32	14	78	78	78
102	32	14	78	102	102	102
32	14	78	214	214	214	214
14	78	312	312	312	312	312
78	600	600	600	600	600	600

Bubble sort algorithm

```
void bubbleSort(int numb[]) {  
    int i, j, temp;  
    for (i = 0; i < SIZE-1; i++)  
        for (j = 0; j < SIZE-i-1; j++) {  
            if (numb[j] > numb[j+1]) {  
                temp = numb[j];  
                numb[j] = numb[j + 1];  
                numb[j + 1] = temp;  
            }  
        }  
}
```

Selection sort – How does it work?

321	14	14	14	14	14	14	14
214	214	32	32	32	32	32	32
66	66	66	66	66	66	66	66
600	600	600	600	78	78	78	78
102	102	102	102	102	102	102	102
32	32	214	214	214	214	214	214
14	321	321	321	321	321	321	321
78	78	78	78	600	600	600	600

Selection sort algorithm

```
void selectionSort(int intArray[]) {  
    int indexMin,i,j, temp;  
    for(i = 0; i < SIZE-1; i++) {  
        indexMin = i; // Set current element as minimum  
        for(j = i+1; j < SIZE; j++)  
            // Check the element to be minimum  
            if(intArray[j] < intArray[indexMin])  
                indexMin = j;  
        if(indexMin != i) { // swap the numbers  
            temp = intArray[indexMin];  
            intArray[indexMin] = intArray[i];  
            intArray[i] = temp;  
        }  
    }  
}
```

Insertion sort – How does it work?

321	214	214	66	66	66	66	66	66	66	66	66	32
214	321	66	214	214	214	214	102	102	102	102	32	66
66	66	321	321	321	321	102	214	214	214	32	102	102
600	600	600	600	600	102	321	321	321	32	214	214	214
102	102	102	102	102	600	600	600	32	321	321	321	321
32	32	32	32	32	32	32	32	600	600	600	600	600

Insertion sort algorithm

```
void insertionSort(int numb[]) {  
  
    int value;  
    int position;  
    int i;  
    for(i = 1; i < SIZE; i++) {  
        value = numb[i];  
        position = i;  
        while (position > 0 && numb[position-1] > value) {  
            numb[position] = numb[position-1];  
            position--;  
        }  
        if(position != i)  
            numb[position] = value;  
    }  
}
```