

Date: 02/25/16

Jeyabalaji Chandrasekaran
jchandr2@binghamton.edu

Design and Analysis of Algorithm – CS575 – 01
Project 1
Part 3

a) Insertion Sort

```
int insertion_sort(int elements[],int no_of_elements)
{
    if(no_of_elements<2)
        return 0;
    for(int i=1;i < no_of_elements;i++) ← first comparison
    {
        int temp_value = elements[i];
        int j = i - 1;
        while(j >= 0 && elements[j] > temp_value) ← second comparison
        {
            elements[j+1] = elements[j];
            j--;
        }
        elements[j+1] = temp_value;
    }

    return 1;
}
```

Solution:

$$\sum_{i=1}^{no_of_elements} (i - 1) = \frac{no_of_elements(no_of_elements - 1)}{2}$$

The time complexity is $\Theta(n^2)$

b) Counting Sort

```
int counting_sort( int* elements, int no_of_elements)
{
    int min, max, z = 0;

    if(no_of_elements < 2)
    {
        cout << "Only one element in the array.";
        return 0;
    }
}
```

```

find_min_and_max( elements, no_of_elements, min, max );
int nlen = ( max - min ) + 1; int* temp = new int[nlen];
memset( temp, 0, nlen * sizeof( int ) );

for( int i = 0; i < no_of_elements; i++ ) ← first comparison
    temp[elements[i] - min]++;

for( int i = min; i <= max; i++ ) ← second comparison
{
    while( temp[i - min] )
    {
        elements[z++] = i;
        temp[i - min]--;
    }
}

delete [] temp;
return 1;
}

```

Solution:

$$\sum_{i=1}^{no_of_elements} 1 + \sum_{i=min}^{max} 2 = no_of_elements + 2(no_of_elements)$$

The time complexity is $\Theta(n)$

c) Merge Sort

```

int merge (int *a, int n, int m)
{
    int i, j, k;
    int *x = (int*)malloc(n * sizeof(int));
    for (i = 0, j = m, k = 0; k < n; k++)
    {
        x[k] = j == n    ? a[i++]
                   : i == m    ? a[j++]
                   : a[j] < a[i] ? a[j++]
                   : a[i++];
    }
    for (i = 0; i < n; i++)
    {
        a[i] = x[i];
    }
}

```

```

    }
    free(x);
    return 1;
}

int merge_sort (int *elements, int no_of_elements)
{
    if (no_of_elements < 2)
        return 0;
    int m = no_of_elements / 2;
    merge_sort(elements, m);
    merge_sort(elements + m, no_of_elements - m);
    merge(elements, no_of_elements, m);
    return 1;
}

```

Solution:

$$\begin{aligned}
 T(\text{no_of_elements}) &= 2T(\text{no_of_elements}/2) + \text{no_of_elements} \\
 &= 2 (\text{no_of_elements}/2) (1 + \log_2 (\text{no_of_elements}/2)) + \text{no_of_elements} \\
 &= \text{no_of_elements}(1 + \log_2 (\text{no_of_elements}) - 1) + \text{no_of_elements} \\
 &= \text{no_of_elements}(1 + \log_2 (\text{no_of_elements})).
 \end{aligned}$$

The time complexity is $\Theta(n \log n)$