Date: 02/25/16

Jeyabalaji Chandrasekaran

[jchandr2@binghamton.edu](mailto:jchandr2@binghamton.edu)

Design and Analysis of Algorithm – CS575 – 01

Project 1

Part 2

1. 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:

The time complexity is

1. 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:

The time complexity is

1. 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:

T(no\_of\_elements) = 2T(no\_of\_elements/2) + no\_of\_elements

= 2 (no\_of\_elements/2) (1 + log2 (no\_elements/2)) + no\_of\_elements

= no\_of\_elements(1 + log2 (no\_of\_elements) − 1) + no\_of\_elements

= no\_of\_elements(1 + log2 (no\_of\_elements)).

The time complexity is