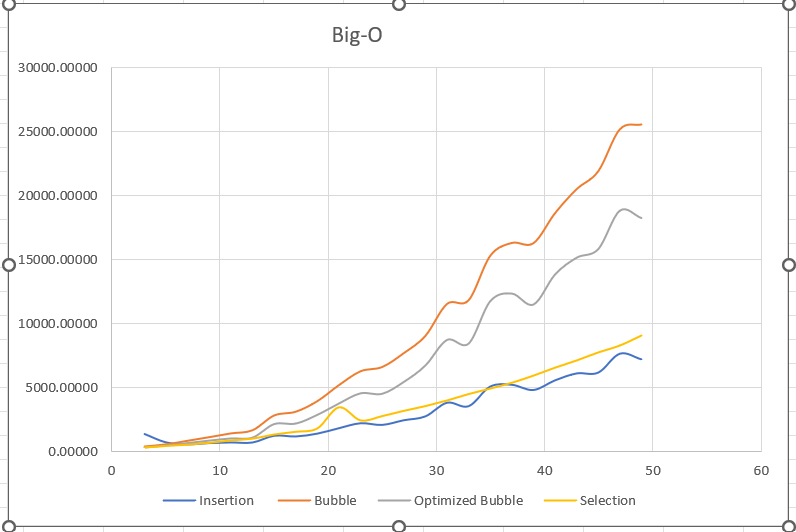
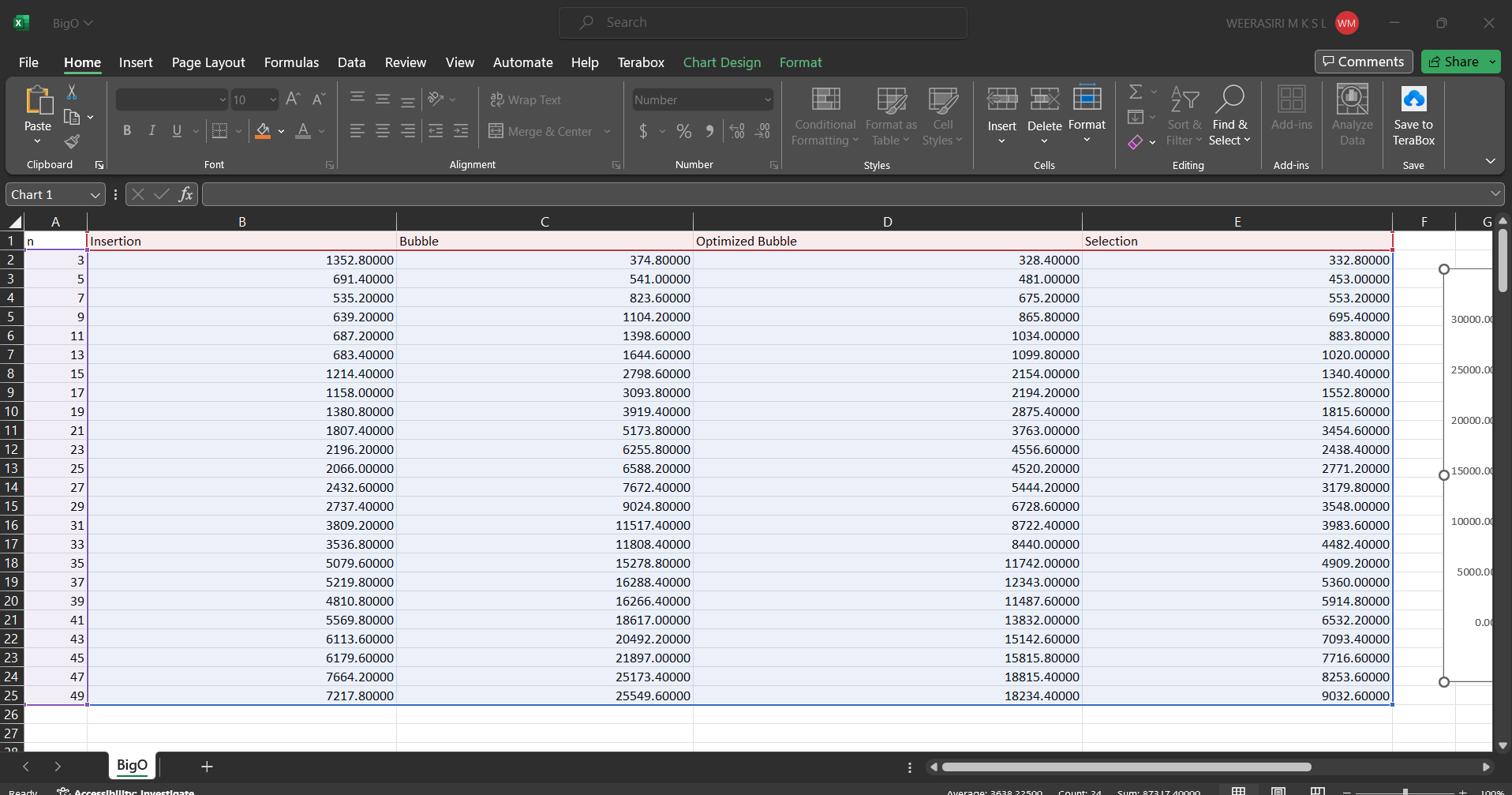
**Inclass Lab(week 2)**

**220689N MKSL Weerasiri**

**Graph:**

**Graph By MS Excel**

**Data:**

**Data By the below Code**

**Run in https://www.tutorialspoint.com/compile\_cpp\_online.php**

**Code:**

#include <iostream>

#include <vector>

#include <chrono>

using namespace std;

void print(int n,vector<int> arr){

for(int i=0;i<n;i++){

std::cout<<arr[i]<<" ";

}

std::cout<<"\n";

}

vector<vector<int>> makeRandomArrays(int start\_size,int end\_size,int step, int value\_limit){

vector<vector<int>> arrays;

vector<int> sample;

for(int i=start\_size;i<end\_size+1;i=i+step){

sample.clear();

for(int j=0;j<i;j++){

sample.push\_back(rand()%(value\_limit+1));

}

arrays.push\_back(sample);

}

return arrays;

}

void swap(int &a,int &b){

int temp=a;

a=b;

b=temp;

}

void runtheProgramInsertion(int n,vector<int> array){

bool is\_inserted;

for(int i=0;i<n;i++){

is\_inserted=false;

int temp=array[i];//keep the current value out of the vector

for(int j=i-1;j>=0;j--){

if(array[j]>temp){//shift the higher values to the right

array[j+1]=array[j];

}else{//insert the key value

array[j+1]=temp;

is\_inserted=true;

break;

}

}

if(!is\_inserted){//insert the value to the front of the vector

array[0]=temp;

}

}

}

void runtheProgramBubble(int n,vector<int> array){

for(int i=0;i<n;i++){

for(int j=0;j<n-1;j++){

if(array[j]>array[j+1]){//check wheather the next value is greater than current value

swap(array[j+1],array[j]);// swap values

}

}

}

}

void runtheProgramOptimizedBubble(int n,vector<int> array){

bool is\_swapped;

for(int i=0;i<n;i++){

is\_swapped=false;

for(int j=0;j<n-i-1;j++){

if(array[j]>array[j+1]){//check wheather the next value is greater than current value

swap(array[j+1],array[j]);// swap values

is\_swapped=true;

}

}

if(!is\_swapped){//no swaps, that means the numbers are sorted

break;

}

}

}

void runtheProgramSelection(int n,vector<int> array){

int index\_minimum;

for(int i=0;i<n;i++){

index\_minimum=i;//keep minimum value's index

for(int j=i+1;j<n;j++){

if(array[j]<array[index\_minimum]){//check for minimum value

index\_minimum=j;

}

}

if(index\_minimum!=i){//minimum index changed, that means need to swap

swap(array[index\_minimum],array[i]);

}

}

}

int main() {

//Get the values

vector<vector<int>> arrays=makeRandomArrays(3,50,2,100);

double sum\_duration;

vector<double> avg\_duration;

string topic;

for(int sorting=0;sorting<4;sorting++){

avg\_duration.clear();

for(int t=0;t<arrays.size();t++){

sum\_duration=0.0f;

for(int i=0;i<5;i++){//5 times

auto start = chrono::high\_resolution\_clock::now();

switch(sorting){

case 0:

runtheProgramInsertion(arrays[t].size(),arrays[t]);

topic="\n\n\nInsertion\n#########\n";

break;

case 1:

runtheProgramBubble(arrays[t].size(),arrays[t]);

topic="\n\n\nBubble\n#########\n";

break;

case 2:

runtheProgramOptimizedBubble(arrays[t].size(),arrays[t]);

topic="\n\n\nOptimized Bubble\n#########\n";

break;

case 3:

runtheProgramSelection(arrays[t].size(),arrays[t]);

topic="\n\n\nSelection\n#########\n";

break;

default:

break;

}

auto end = chrono::high\_resolution\_clock::now();

// Calculating total time taken by the program.

double time\_taken =

chrono::duration\_cast<chrono::nanoseconds>(end - start).count();

sum\_duration=sum\_duration+time\_taken;

}

avg\_duration.push\_back(sum\_duration/5.0f);

}

cout<<topic;

for(int i=0;i<avg\_duration.size();i++){

printf("%.20f\n",avg\_duration[i]);

}

}

return 0;

}