**CS 201 Data Structures**

**Assignment 1**

**Time Complexity and Big-Oh**

**Submission Date: Thursday August 31, 2017 in class**

**Problem1**

Assume that basic operations and input output take single time units to complete. Calculate the time complexity function T(n) and big-oh of T(n) for the following program fragments both in best case and worst case. Also give the values of c and N with the big-oh function:

|  |
| --- |
| **GCD Code fragment**  int GCD(int n,int m){  int remainder;  while(m!=0){  remainder = n%m;  n = m;  m = remainder;  }    return n;  } |
| **Code fragment that returns true if arr contains two numbers whose sum is equal to num**  bool arrSum(int num, int arr[], int size)  {  int i, j;    for(i =0; i<size; i++){  for(j=i+1;j<size;j++)  if(arr[i]+arr[j] == num)  return true;  }    return false;  } |
| **Code fragment that returns true if arr contains two numbers whose sum is equal to num assuming arr is sorted. BinSearch(int a[], int size, int key) returns a number between 0 and size-1 if search is successful and false otherwise.**  bool arrSum(int num, int arr[], int size)  {  int i, j;    for(i =0; i<size; i++){    if(BinSearch(arr, size, num-arr[i])!=-1)  return true;  }    return false;  } |
| **Code fragment for bubble sort**  void bubbleSort(int arr[],int n)  {  bool done = false;  for(int i = 1; (i < n) && !done; i++)  {  done = true;  for (int j=0; j <n-i; j++)  {  if (arr[j+1] < arr[j])  {  swap(arr[j+1],arr[j])  done = false; //a swap is made and so sorting continues  }  }  }  } |