1. Give an analysis of the running times for each of the following program fragments:

|  |  |
| --- | --- |
| **Fragment #** | **Code** |
| A | for (int i = 0; i < n; ++i)  cout<<i; |
| B | for (int i = 1; i <n ; i \*= 2)  cout<<i; |
| C | for (int i = 0; i < n; i++)  f(n); // where f(n) has O(log n) |
| D | void decimal2binary (int n)  { if(n>0)  {decimal2binary(n/2); cout<<n%2; }  } |

1. Create a C/C++ function that takes an array of integers and returns whether it has repeated items or not. Determine the big O of your solution. **Repeat if the array is sorted.**
2. Write a program that recursively finds the maximum value of an array of integers.
3. Sort an array using bubble sort where you compare each two consecutive numbers and swap them if they are not in an ascending order. You may run your operations in a number of paths equal to the array size or if all array elements are in ascending order stop!
4. Create a dynamic array of float numbers, the size of the array is determined by the user through cin, each element in the array holds a value of 1/(index)! i.e a[i]=1.0/i!, run your program and compute the sum of the array elements (which value the sum tends to ?)
5. Create a class for email address book. Your class should have the following

* An array of strings containing email followed by the person name, an integer number to tell how many emails are stored so far. (private)
* ***gets*** and ***sets*** methods for email and name for each entry. ***Print\_all*** method to list all entry values (public) a constructor method to initialize the size used so far to zero.

1. Estimate the big O for the following T(n)