# **Design & Analysis of Algorithms**

# **Practice Problems**

# **TIME COMPLEXITY**

* **Sum of arithmetic series: 1+2+3+4+…+N = (N\*(N+1))/2**
* **Sum of geometric series: 1+2+4+8+16+…+N = (a\*(r^N – 1 ))/r-1**
* **Sum of square of each term: 1 +2^2 +3^2 + 4^2 + … +N^2 = (N\*(N+1)\*(2N+1))/6**

**Q: Determine the time complexity of the following code snippets.**

|  |  |
| --- | --- |
| **1) Solved sample problem**  **int Sum=0; // O(1) Time**  **for(int i=1; i<=N; i++) //(1+1+1+...+1) - - - N\*1 Times =O(N)**  **for(int j=1; j<=N; j++) //(N+N+...+N) - - - N\*N Times = O(N2)**  **Sum++; //(N+N+...+N) - - - N\*N Times = O(N2)**  **Overall Complexity: O(1) +O(N) + O(N2) + O(N2) = O(N2)** | **2)**  **int Sum=0;**  **for(int i=1; i<N; i\*=2)**  **for(int j=1; j<=i; j++)**  **Sum++;** |
| **3)What is the algorithm’s complexity of the following piece**  **of code.**  **int Sum=0;**  **for(int i=1; i<=N; i++)**  **for(int j=1; j<=N; j++)**  **for(int k=1; k<=N; k++)**  **Sum++;** | **4)**  **int Sum=0;**  **for(int i=1; i<=N; i++)**  **for(int j=1; j<=i; j++)**  **for(int k=1; k<=j; k++)**  **Sum++;** |
| **5)**  **int Sum=0;**  **for(int i=1; i<N; i\*=2)**  **for(int j=1; j<N; j\*=2)**  **Sum++;** | **6)**  **Int Sum=0;**  **for(int i=1; i<=N\*N; i+=2)**  **for(int j=1; j<N\*N; j\*=2)**  **Sum++;** |
| **7)**  **int Sum=0;**  **for(int i=1; i<=N; i++)**  **for(int j=1; j<=i\*i; j++)**  **Sum++;** | **8)**  **int Sum=0;**  **for(int i=1; i<N\*N; i\*=2)**  **for(int j=1; j<N\*N; j\*=2)**  **Sum++;** |
| **9)**  **int Sum=0;**  **for(int i=1; i<=N; i++)**  **for(int j=1; j<=i; j++)**  **for(int k=1; k<=j; k++)**  **Sum++;** | **10)**  **int Sum=0;**  **for(int i=1; i<=N; i\*=2)**  **for(int j=1; j<i; j\*=2)**  **Sum++;** |
| **11)**  **int Sum=0;**  **for(int i=1; i<=N\*N; i\*=2)**  **Sum++** | **12)**  **int Sum=0;**  **for(int i=1; i<=sqrt(N); i\*=2)**  **for(int j=1; j<=1000000; j++)**  **Sum++;** |

**Algorithm Design Problems**

Q#1: Given a sorted array of integers of size “N” and a target value. Your task is to design an algorithm to determine whether there exists any pair of indices having sum equals to the target value.

**Constraint: O(N) solution is required**

**bool searchPair(Arr[ ], N, trVal)**

Sample Problem: {3,5,6,9,12,18,24}, trVal = 17.

Return true because 5+12 = 17

Sample Problem: {3,5,6,9,12,13,18,24}, trVal = 13.

Return false because there is no pair having sum equals to 13.

Q#2: We have an unsorted array of distinct integers of size “N” having data in the range of (1-N). Assume that one of the values in the array is mistakenly replaced by 0. Now we are interested in determining the overwritten value. Design an algorithm to determine the overwritten value i.e., the original value which is replaced by 0.

**Constraint: O(N) solution is required using constant amount of memory**

Q#3: Insertion sort Algorithm with time and space complexity analysis. (Best and worst-case complexity)

Q#4: Bubble sort Algorithm with time and space complexity analysis. (Best and worst-case complexity)

Q#5: Selection sort Algorithm with time and space complexity analysis. (Best and worst-case complexity)