**1. What is the time, space complexity of following code:**

|  |
| --- |
| int a = 0, b = 0;  for (i = 0; i < N; i++) {  a = a + rand();  }  for (j = 0; j < M; j++) {  b = b + rand();  } |

**Options:**

1. O(N \* M) time, O(1) space
2. O(N + M) time, O(N + M) space
3. O(N + M) time, O(1) space
4. O(N \* M) time, O(N + M) space

Ans: Option 3

**2. What is the time complexity of following code:**

int a = 0;

for (i = 0; i < N; i++) {

for (j = N; j > i; j--) {

a = a + i + j;

}

}

**Options:**

1. O(N)
2. O(N\*log(N))
3. O(N \* Sqrt(N))
4. O(N\*N)

Ans: Option 4

**3. What is the time complexity of following code:**

**int i, j, k = 0;**

**for (i = n / 2; i <= n; i++) {**

**for (j = 2; j <= n; j = j \* 2) {**

**k = k + n / 2;**

**}**

**}**

**Options:**

1. O(n)
2. O(nLogn)
3. O(n^2)
4. O(n^2Logn)

Ans: Option 2

**4. What does it mean when we say that an algorithm X is asymptotically more efficient than Y?**

**Options:**

1. X will always be a better choice for small inputs
2. X will always be a better choice for large inputs
3. Y will always be a better choice for small inputs
4. X will always be a better choice for all inputs

Ans: Option 2

##### **5. What is the time complexity of following code:**

int a = 0, i = N;

while (i > 0) {

a += i;

i /= 2;

}

**Options:**

1. O(N)
2. O(Sqrt(N))
3. O(N / 2)
4. O(log N)

Ans: Option 4

**Interview Bit**

# **LOOP\_CMPL**

Ans:**O(N + M) time, O(1) space**

**Since first loop iterates to N and second loop iterates to M implies O(N+M)**

# 

# **NESTED\_CMPL:**

Ans: **O(N \* N) time, O(1) space**

**Since loop within a loop iterating to n implies O(n\*n)**

# **NESTED\_CMPL2**

Ans: O(N\*N)

Since it iterates as to be O(N(N+1)/2) which implies O(N\*N)

# **CHOOSE4**

Ans: **X will always be a better choice for large inputs //Repeated Qn**

# **WHILE\_CMPL**

Ans: O(logN) //Repeated Qn

# **NESTED\_CMPL3**

Ans: O(N)

Since it iterates as N+N/2^x implies O(N)

# **LOOP\_CMPL2**

Ans: O(NlogN)

Since it iterate as N/2\*(N/2+N/2-1+N/2-2……) implies O(NlogN)

# **GCD\_CMPL**

Ans: O(logN)

# **CHOOSE1**

**Ans: n^3 / (sqrt(n))**

**Since bounded by O(N\*N) implies O(f(n)) where f(n)<=N\*N here n^3 / (sqrt(n)) equal to n\*\*2.5 > n\*\*2**

# **CHOOSE3**

Ans: C since its time complexity is O(logN) it is faster compared to other options given

# **CHOOSE2**

Ans:**f3, f2, f4, f1**

# **REC\_CMPL1**

Ans: Didn’t get the ans thought it was O(N\*N) at first. I thought of binary search.

# **REC\_CMPL2**

Ans: Didn’t get the the answer

# **REC\_CMPL3**

Ans: Didn’t get the the answer

# **AMORTIZED1**

Ans: O(N)