1. What is the time complexity of following code:

```
int a = 0;
for (i = 0; i < N; i++) { //O(N) * O(N) ⇒ O(N*N)
   for (j = N; j > i; j--) { // O(N)
      a = a + i + j;
   }
}
```

   Options:

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

2. What is the time complexity of following code:

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) { // O(N) * O(LogN) ⇒ O(NLogN)
   for (j = 2; j <= n; j = j * 2) { // O(NLogN)
      k = k + n / 2;
   }
}
```

   Options:

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

3. 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

4. What is the time complexity of following code:

```
int a = 0, i = N;
while (i > 0) {
   a += i;
```

```
    i /= 2; // O(LogN)
}
```

Options:

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

5.