Question **1**

Incorrect

Mark 0.00 out of 1.00

Flag question

What is the time complexity of the below function?

```
void fun(int n, int arr[])
{
   int i = 0, j = 0;
   for (; i < n; ++i)
      while (j < n && arr[i] < arr[j])
         j++;
}
```

- a. $O(n * \log(n)^2)$
- b. $O(n)$
- c. $O(n * \log(n))$
- d. $O(n^2)$ ✖

Your answer is incorrect.

The correct answer is: O (n)

Question **2**

Correct

Mark 1.00 out of 1.00

Flag question

In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed. Consider the following for loops:

A) for(i = 0; i < n; i++)

B) for(i = 0; i < n; i += 2)

C) for(i = 1; i < n; i *= 2)

D) for(i = n; i <= n; i /= 2)

If n is the size of input (positive), which function is most efficient (if the task to be performed is not an issue)?

- a. A
- b. B
- c. C ✔
- d. D

Question **3**

Incorrect

Mark 0.00 out of 1.00

Flag question

What is time complexity of fun()?

```
int fun(int n)
{
   int count = 0;
   for (int i = n; i > 0; i /= 2)
      for (int j = 0; j < i; j++)
         count += 1;
   return count;
}
```

- a. O (n^2)
- b. O (n *log (n* log (n)))
- c. O (n)
- d. O (n * log (n))

Your answer is incorrect.

The correct answer is: O (n * log (n))

Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

Consider the following two functions fun1 and fun2. What are time complexities of the functions fun1 and fun2?

```
int fun1(int n)
{
   if (n <= 1) return n;
   return 2*fun1(n-1);
}
int fun2(int n)
{
   if (n <= 1) return n;
   return fun2(n-1) + fun2(n-1);
}
```

- a. O (n) for both fun1 ( ) and fun2 ( )
- b. O (n) for fun1 ( ) and O (2 ^ n) for fun2 ( )
- c. O (2 ^ n) for fun1 ( ) and O (n) for fun2 ( )
- d. O (2 ^ n) for both fun1 ( ) and fun2 ( )

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

Consider the following function:

```
int unknown(int n) {
  int i, j, k = 0;
  for (i  = n/2; i <= n; i++)
    for (j = 2; j <= n; j = j * 2)
      k = k + n/2;
  return k;
}
```

What is the time complexity of the function?

- a. n ^ 3 log n
- b. n ^ 3
- c. n ^ 2
- d. n log n ✔

Your answer is correct.

The correct answer is: n log n