HackerRank For Work

INFO6205_Fall2018_Section... 30 minutes

Question - 1

SCORE: **6 points**

Question 1

What is the time complexity of following the function *fun()*?
Assume that *log(x)* returns log value in base 2.

```
void fun()
{
    int i, j;
    for (i=1; i<=n; i++)
        for (j=1; j<= log (i); j++)
            printf ( "TimeComplexity" );
}
```

- ~ n ln(n)
- ~ ln(n)
- ~ log(n)
- ~ n log(n)

Question - 2

SCORE: **20 points**

Question 2

By induction show that $5^n - 1$ is divisible by 4 for all values of $n \geq 1$

Question - 3

SCORE: **6 points**

Question 3

What is complexity of 3-sum problem in terms of Big Omega?

- O(N^2)
- Ω(N^2)
- O(N^3)
- Ω(N^3)

Question - 4

SCORE: **6 points**

Question 4

What is the least and greatest number of key compares that a Binary Search algorithm might use for a sorted array of size N.
Assume that it is possible that they key you seek is not in the array at all.

- Least : 1, Most:logN
- Least : 1, Most: logN+1
- Least : 1, Most: N
- None of the above

Question - 5

SCORE: **6 points**

Question 5

True or False: it's always safe to use mid = (low + high) / 2 in binary search.

- True
- False

Question - 6

SCORE: **6 points**

Question 6

Select the right order depending on complexity.

- O(log N) < O(N) < O(N log N) < O(2^N) < O(N^2)
- O(log N) < O(N log N) < O(N) < O(N^2) < O(2^N)
- O(log N) < O(N) < O(N log N) < O(N^2) < O(2^N)
- O(log N) < O(N log N) < O(N) < O(N^2) < O(2^0)