Practice Worksheet

Ques.1 Let w(n) and A(n) denote respectively, the worst case and average case running time of an algorithm executed on an input of size n. which of the following is ALWAYS TRUE?

(a) $A(n) = \Omega(W(n))$ (b) $A(n) = \Theta(W(n))$ (c) $A(n) = O(W(n))$ (d) None of the above

Ques.2 Arrange these functions by order of growth from highest to lowest

$100*n^2$, 1000, 2^n, $10*n$, n^3, $2*n$

$2^n > n^3 > 100n^2 > 10n > 2n > 1000$

Ques.3 What is the time complexity of the following code fragments?

(a)
```
int fun(int n)
{
int count = 0;
for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
                for(k = 0; k < n; k++)
                        count += 1;
return count;
}
```
$O(n^3)$

(b)
```
int a = 0;
for (i = 0; i < N; i++) {
        for (j = N; j > i; j--) {
                a = a + i + j;
        }
}
```
$O(N^2)$

(c)
```
int i, j, k = 0;
for (i = n / 2; i <= n; i++)
{
        for (j = 2; j <= n; j = j * 2)
        {
                k = k + n / 2;
        }
}
```
$O(n \log n)$

(d)
```
int fun(int n)
{
        for (int i = 1; i <= n; i++)
        {
                for (int j = 1; j < n; j += i)
                {
                        // Some O(1) task
                }
        }
}
```
$\Theta(\log n)$

(e)
```
void fun()
{
int i, j, count = 0;
for ( i = n/2; i <= n; i++)
        for ( j = 1; j <= n; j = j * 2)
                for( k = 1; k <= n; k = k * 2)
                        count++;
}
```
$O(n \log n)$

(f)
```
void fun(int n, int k)
{
    for (int i=1; i<=n; i++)
    {
            int p = pow(i, k);
            for (int j=1; j<=p; j++)
            {
                    // Some O(1) work
            }
    }
}
```
$O(n^K)$

(g)
```
fun(int n)
{
     for(i = 1; i <= n; i = i*2)
     {
          for(j = 1; j <= i; j = j*2)
               printf(" Hii ");
     }
}
```
$O(\log(n))^2$ or $O(\log^2(n))$

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

(i)
```
void function(int n)
{
   int count = 0;
   for (int i=n/2; i<=n; i++)          // n/2 times
     for (int j=1; j+n/2<=n; j = j++)  // n/2 times
       for (int k=1; k<=n; k = k * 2)  // log n times
         count++;
}
```
$O(n^2 \log n)$

Ques4. For the functions, n^k and c^n, what is the asymptotic relationship between these functions? Assume that k >= 1 and c > 1 are constants.

$n^K = O(c^n)$, for $K = 2$ and $c = 2$

Ques5. Decide whether these statements are True or False:

1. If f(n) = Θ(g(n)) and g(n) = Θ(h(n)), then h(n) = Θ(f(n)) True
2. If f(n) = O(g(n)) and g(n) = O(h(n)), then h(n) = Ω(f(n)) True
3. If f(n) = O(g(n)) and g(n) = O(f(n)) then f(n) = g(n) False [for $f(n) = n$ & $g(n) = n+1$]
4. $\frac{n}{100} = \Omega(n)$ True for $c = 1/200$ [$\frac{n}{100} < c*n$ for $c = 1/200$]

Ques6. Find the complexity of below recurrence:

$$T(n) = \begin{cases} 1, & n = 0 \\ 3T(n-1), & n > 0 \end{cases}$$

$T(n) = O(3^n)$

Ques7. Find the complexity of below recurrence:

$$T(n) = \begin{cases} 1, & n = 0 \\ 2T(n-1) - 1, & n > 0 \end{cases}$$

$T(n) = O(1)$

Constant Time

Ques8. Find the complexity of below recurrence:

$$T(n) = \begin{cases} 1, & n = 0 \\ 7T(n/2) + 3n^2 + 2, & n > 0 \end{cases}$$

$T(n) = \theta(3n^2 + 2)$

Ques. 3

(c) O(n*logn)

Explaination:-

If you notice, j keeps doubling till it is less than or equal to n. Several times, we can double a number till it is less than n would be log(n).
Let's take the examples here.
for n = 16, j = 2, 4, 8, 16
for n = 32, j = 2, 4, 8, 16, 32
So, j would run for O(log n) steps.
i runs for n/2 steps.
So, total steps = O(n/ 2 * log (n)) = **O(n*logn)**

(d) Θ (logn)

Explaination:- Visit this link

https://www.geeksforgeeks.org/interesting-time-complexity-question/

(e) O(n*logn)

Explaination:- The outer loop will run for n/2 times and for each iteration of the outer loop, inner loop will run $\log_2 n$ times.

(f) $O(n^k)$

Explaination: Visit this link

https://www.geeksforgeeks.org/time-complexity-of-loop-with-powers/

Ques. 5

1. True. Θ is transitive
2. True. O is transitive, and h(n) = Ω(f(n)) is the same as f(n) = O(h(n))

Ques.8 $\Theta(3n^2 + 2)$

Explaination: Visit this link

https://www.geeksforgeeks.org/practice-set-recurrence-relations/