Name: Aryan Srivastava Roll no.: 27
Sec: D51

10) n^k c^n

$k >= 1$ $c > 1$

$k = 1$ $c = 2$

n^1 2^n

n 2^n

Q1) Asymptotic notations are used in computer science and mathmatics to describe the behaviours or algorithms or functions as their input size approaches infinity. The three main notations are

i) Big O(o):
Represent ~~another part~~ an upper bound on a growth rate of a function.

ii) Omega (Ω): Represent the lower bound and the growth of wait function.

iii) Theta (Θ): Represents both upper and lower bond indicating tight ~~one~~ bond on the equation

A2)
```
for (i=1 to n)
    {
        i = i*2;
    }
```

The time complexity of the given loop O(log n) this is because variable i is in multiple by 2.

Date
Page
Extra fine paper

Q3) The recursion relation suggest ~~that~~ that
$T(n) = 3^k T(n-k)$ until $n-k=1$
$k = 3^k$
$T(n) = 3^{(n-1)}$
$T(n) = 3^{(n-1)} = O(3^n)$
so, time complexity is $O(3^n)$

Q4) $T(n) = 2^n - (2^0 + 2^1 + 2^{(n-1)})$
$T(n) = 2^n - (2^n - 1)$
$T(n) = 1$
So time complexity is constant, we can express it as $O(1)$.

Q5) The time complexity is $O(\sqrt{n})$ as the loop trying approxe $\sqrt{n}$ relations where n is the input parameter.

Q6) The time complexity of the given code is $O(\sqrt{n})$, This is beause the loop iterats until $i*i$ exceeds n, and n each execution is increased as the loop ~~eq~~ stop which $i*i$ becomes greter than n.

Q7) The outer loop sums, $i = n/2$ to ~~O~~n it iterates $(n - n/2) + 1$ times, which is app. $n/2$ times.
The second loop runs from $j=1$ to n. So its run appox. $\log(n)$ time.
The third loop is similar to second loop i.e. $\log(n)$ times.
Since, they are nested we multiply them.

$$O(n) \times O(\log n) \times O(\log n) = O(n \log^2 n)$$

Q5>

Ans: The inner loop for (j=1 to n) 'run n times.
The outer loop 'for (i=1 to n) run n times.
The recursive call function(n-3) reduced size of n by 3 in each recursive call until n = 1.

$T(n) = n \times n + T(n-3)$

$T(n) = n^2 + T(n-3)$

Solving this recurrence is bit complex, but we can approximate it to $O(n^2)$.

Q5>

Ans: The outer loop runs from i = 1 to n
the inner loop runs from j = 1 to n but increment is i.e. $T = j+i$.
when i = 1, it runs n times
when i = 2, it runs n/2 times
when i = 3, the inner loop runs n/3 times.
when i = n, the inner loop runs only once.

$$1 + \frac{n}{2} + \frac{n}{3} + \cdots + 1$$

$\therefore O(\log n)$ Ans