1.2 Show that any binary integer is at most four times as long as the corresponding decimal integer. For veri large numbers, what is the ratio of these two lengths, approx.?

For any decimal number of length n, the ratio of its length in binary to its length in decimal is $\frac{\lceil \log_2 10^n - 1 \rceil}{n}$. To prove that the ratio is at most 4, prove by contradiction.

$$\frac{\lceil \log_2 10^n - 1 \rceil}{n} \leq 4 \xrightarrow{\text{Contradiction}} \frac{\lceil \log_2 10^n - 1 \rceil}{n} > 4$$

$\log_2 10^n - 1 > 4n$

$10^n - 1 > 2^{4n}$

$10^n - 1 > 16^n$

$10^n > 16^n + 1$ ← This is false when $n \geq 0$.

So the original statement, $\frac{\lceil \log_2 10^n - 1 \rceil}{n} \leq 4$ must be true.

To find the ratio for very large numbers, find $\lim_{n \to \infty} \frac{\lceil \log_2(10^n - 1) \rceil}{n}$

$$\lim_{n \to \infty} \frac{\log_2(10^n - 1)}{n}$$

$$\lim_{n \to \infty} \frac{\log_2(10^n)}{n}$$

$\log_2 10$. For very large numbers, the ratio is $\log_2 10$ approximately.

1.3 A d-ary tree is a rooted tree in which each node has at most d children. Show that any d-ary tree with n nodes must have a depth of $\Omega(\log n / \log d)$.

Depth is the length of the longest path from the root to leaf node.
The minimum depth of a d-ary tree is when the tree is complete and that all depths that exist should be fully populated before beginning to populate the next level.

In this case, the tree has a balanced structure, and the minimum depth can be calculated using logarithmic functions.
In the worst case scenario, each node, n, has d children.
Number of nodes at level i is d^i
Since the tree has n nodes, we can find the maximum level of the tree using: $n = d^h$

$$\log(n) = h \cdot \log(d)$$

$$h = \frac{\log(n)}{\log(d)}$$

Then, we can say that the depth is $\Omega(\log(n) / \log(d))$

The precise formula for the minimum depth is $\lfloor \log(n) \rfloor$

1.4 Show that $\log(n!) = \Theta(n \log n)$

Upper bound

Since $n! \leq n^n$ for very large $n \to \log(n!) = O(\log n^n)$

which means $\log(n!) = O(n \log n)$.

lower bound

Since $n! > \left(\frac{n}{2}\right)^{n/2}$ for very large $n \to \log(n!) = \Omega\left(\left(\left(\frac{n}{2}\right)^{n/2}\right)\right)$

which means $\log(n!) = \Omega(n \log(n))$.

Therefore, $\log(n!) = \Theta(n \log n)$

1.10 Show that if $a \equiv b \pmod N$ and if M divides N then $a \equiv b \pmod M$

$a \equiv b \pmod N$

$N = KM$ for any integer K

$a \equiv b \pmod{KM}$ is the same as saying

$a \equiv b \pmod M$

1.16 The algorithm for computing $a^b \bmod c$ by repeated squaring does not necessarily lead to the minimum number of multiplications. Give an example of $b > 10$ where the exponentiation can be performed using fewer multiplications, by some other method.

In the case $71^{500} \pmod{35}$. Simplify $71 \bmod 35$ will give you $71^{500} \equiv 1^{500} \pmod{35}$

1.18 Compute gcd$(210, 588)$

Factorization: $210 = 2 \times 3 \times 5 \times 7$

$588 = 2^2 \times 3 \times 7^2$

Common primes: $2, 3, 7$ gcd$(210, 588) = 2 \times 3 \times 7 = 42$

Euclid's algorithm: $588 = \underline{210} \cdot 2 + \underline{168}$

$210 = \underline{168} \cdot 1 + \underline{42}$ 42 is the gcd

$168 = 4 \cdot 42 + 0$

1.18 Euclid's extended algorithm

$588 = \underline{210} \cdot 2 + \underline{168}$ $\leftarrow$ $168 = 588 - 210 \cdot 2$

$210 = \underline{168} \cdot 1 + \boxed{42}$ $42 = 210 - 168 \cdot 1$

$168 = 4 \cdot 42 + 0$

$= 210 + 168(-1)$

$= 210 + (588 - 210(2))(-1)$

$= 210 + (588 + 210(-2))(-1)$

$= 210 + 588(-1) + 210(2)$

$42 = 210(3) + 588(-1)$

1.26 What is the least significant decimal digit of $17^{17^{17}}$?

$17^{17^{17}} \pmod{10}$

$10 = 2 \cdot 5$ (2,5 are primes)

$p = 2 \quad q = 5 \quad a = 17$

$a^{(p-1)(q-1)} \equiv 1 \pmod{pq}$

$17^{1 \cdot 4} \equiv 1 \pmod{10}$

$17^{17} = (4^2+1)^{17} = 4 \cdot C + 1$ C is a constant

$17^{17^{17}} \pmod{10} = 17^{4 \cdot C} \pmod{10} \cdot 17 \pmod{10}$

$= 7$