Problem 3 :

T.S.T. divide(x, y) returns (q, r), such that $x = qy + r$, $q \geq 0$ and $0 \leq r < y$.

Proof by PMI on x.

Basis : $x = 0$

divide$(0, y) = (0,0) = (q, r)$.

I.H. : for some $0 \leq z \leq x$

divide$(z, y) = (q, r)$

such that $z = qy + r$.

I.S : Case I :

$x = 2k$

divide$(x \text{ div } 2, y) = (q, r)$.

$k = qy + r$.

$q_1 = 2q$ $\quad$ $0 \leq r < y$.

$r_1 = 2r$

if $r_1 < y \rightarrow (q_1, r_1)$ as reqd since $(r_1 < y)$

if $r_1 > y \rightarrow (q_1 + 1, r_1 - y) = (2q+1, 2r - y)$.

since $r < y$, $\Rightarrow 2r - y < y$.

$\therefore x = (q_1 + 1) y + (r_1 - y)$ which is same by algo.

Case II :

$n = 2k + 1$.

divide$(n \text{ div } 2, y) = (q, r)$

$k = qy + r$.

$q_1 = 2q$ $\qquad 0 \le r < y$

$r_1 = 2r$. $\qquad r_2 = r_1 + 1 = 2r + 1$

if $r_2 < y \longrightarrow (q_1, r_2)$ as req since $r_2 < y$.

if $r_2 > y \longrightarrow (q_1 + 1, r_2 - y)$

$= (2q + 1, 2r + 1 - y)$

now, $(2r + 1 - y) < y$ since $r < y$.

Thus, $n = (q_1 + 1) y + (r_2 - y)$

which is same as algo.

$T(n) = T\left(\frac{n}{2}\right) + 1$

$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + 1$

...

$T(n) = \log_2 n + c$.

$T(n) = \boxed{O(\log_2 n)}$

since no. of digits in n is n, n will grow as 10^n.

$\Rightarrow T(n) = O(\log_2 10^n)$

$= O(n \log_2 10)$

$= \boxed{O(n)}$

Problem 4 :

```
fun isqrt(n) =
    let fun maxpow(n) =
            if (n div 4 = 0) then 1
            else 4 * maxpow (n div 4);

        fun iter(i, n, c) =
            if (c = 0) then i
            else if (2*i + 1) * (2*i+1) > (n div c)
                then iter (2*i, n, c div 4)
            else iter (2*i +1, n, c div 4);

    in
        iter( 0, n, maxpow(n))
    end;
```

Since $n \text{ div } 4^{k_0} > 0$

and $n \text{ div } 4^{k_0+1} = 0$

$T(N) = T\left(\frac{N}{4}\right) + 1$

$T\left(\frac{N}{4}\right) = T\left(\frac{N}{4^2}\right) + 1$

$\boxed{O(\log_4 N)}$

$\vdots$

$T(N) = (\log_4 N) + c$