COL 226 Assignment

1 Prerequisites

For a given positive integer n

- $l(n)$ denotes the number of significant digits in n
- n_i denotes the i-th digit of n

So n $= \sum_{i=0}^{l(n)-1} n_i \cdot 10^i$
It may also be represented as $n_{l(n)-1}...n_1n_0$

2 Integer Square Root by Long Division

We first define a helper function *sqroot*.

SQROOT(q, r, d)
if $d = 0$ **then**
  return $(0, r)$
$r2 \leftarrow 100r + 10d_{l(d)-1} + d_{l(d)-2}$
if $l(d) = 2$ **then**
  $d2 \leftarrow 0$
else
  $d2 \leftarrow d_{l(d)-3}...d_1d_0$
$g \leftarrow$ largest non-negative integer such that $(10q + g)g \leq r2$
$(a, fr) \leftarrow$ SQROOT$(10q + 2g, r2 - (10q + g)g, d2)$
return $(g \cdot 10^{l(a)} + a, fr)$

Proposition 1. *For a positive integer n with $l(n) = 2x, x \geq 1$,* SQROOT $(0, 0, n)$ *returns its integer square root and remainder.*

Proof. When $x = 1$ it returns $(a, n - a^2)$ where a is the largest integer such that $a^2 \leq n$ so a is the integer square root.

Assume that the proposition holds for some $x = x_0$. Let n be an integer with $l(n) = 2(x_0 + 1)$. Let n' be the integer $n_{l(n)-1}...n_3n_2$ and s' be its integers square root. During the recursion for SQROOT$(0, 0, n)$, all calls before the final one have same values of q, r as calls of SQROOT$(0, 0, n')$ since the digits considered till then are same in n and n'. The penultimate call would be SQROOT$(2s', n'^2 - s'^2, n_1n_0)$. In it, we find the largest positive integer g such that

$$(20s' + g)g \leq 100n'^2 - s'^2 + 10n_1 + n_0$$

$$\Rightarrow (10s' + g)^2 \leq n$$

Since $s'^2 \leq n'^2 \Rightarrow (10s')^2 \leq n^2$, such a g would exist so $10s' + g$ is the integer square root of n which gets formed while backtracking. The r value in final call is $n - (10s' + g)^2$ which is returned as fr while backtracking.

Hence, proved by induction that the `SQROOT`$(0, 0, n)$ returns correct integer square root and remainder when $l(n) = 2x$. □

The main algorithm to find square root is following.

`ISQRTLD`(n)
if $l(n)$ *is even* **then**
    return `SQROOT` $(0, 0, n)$
else
    $g \leftarrow$ largest non-negative integer such that $g^2 \leq n_{l(n)-1}$
    if $l(n) = 1$ **then**
        $n2 \leftarrow 0$
    else
        $n2 \leftarrow n_{l(n)-2}...n_1 n_0$
    $(a, fr) \leftarrow$ `SQROOT`$(2g, n_{l(n)-1} - g^2, n2)$
    return $(g \cdot 10^{l(a)} + a, fr)$

Theorem 1. *For any positive integer n,* `ISQRTLD`(n) *returns the integer square root of n and the remainder.*

Proof. When $l(n)$ is even `SQROOT`$(0, 0, n)$ is called and it returns the integer square root according to Proposition 1.

Let $l(n) = 2x + 1, x \geq 0$. When $x = 0$, it finds the integer square root in first call and returns it.

Assume that it returns integer square root for some $x = x_0$. Let n be an integer with $l(n) = 2(x_0 + 1) + 1$. Let n' be the integer $n_{l(n)-1}...n_3 n_2$ and s' be its integers square root. During the calls of `SQROOT` due to `ISQRTLD`(n), all calls before the final one have same values of q, r as calls due to `ISQRTLD`(n') since the digits considered till then are same in n and n'. The penultimate call would be $sqroot(2s', n'^2 - s'^2, n_1 n_0)$. In it, we find the largest positive integer g such that

$$(20s' + g)g \leq 100n'^2 - s'^2 + 10n_1 + n_0$$

$$\Rightarrow (10s' + g)^2 \leq n$$

Since $s'^2 \leq n'^2 \Rightarrow (10s')^2 \leq n^2$, such a g would exist so $10s' + g$ is the integer square root of n which gets formed while backtracking. The r value in final call is $n - (10s' + g)^2$ which is returned as fr while backtracking.

Hence, proved by induction that the `ISQRTLD`(n) returns correct integer square root and remainder when $l(n)$ is odd. □