A274945: Staircase Sequence

Enrique Pérez Herrero

June 16, 2017

1 Summary

The Staircase Sequence (OEIS: A274945), present some irregularities at the beggining (below a(84)), but afterwards a general formula can be constructed.

This paper uses a Rmarkdown script using R programming code to construct, and test, a general formula for the sequence. For doing first sequence exploration R package OEIS.R is used.

A further and more formal mathematical work must be done to derive or simplify the formula.

Code to calculate sequence terms is included in the paper.

2 Packages

2.1 Installing OEIS.R

Installing `OEIS.R` package from Github.

```
library(devtools)
devtools::install_github("EnriquePH/OEIS.R")
```

2.2 Loading Packages

```
library(OEIS.R)
library(ggplot2)
```

3 Loading Staircase Sequence in R

```
id <- "A274945"
A274945 <- OEIS_sequence(id)
A274945$description
```

```
## [1] "The Staircase Sequence: numbers with at least three digits and with the property that the sum o
```

```
A274945$terms
```

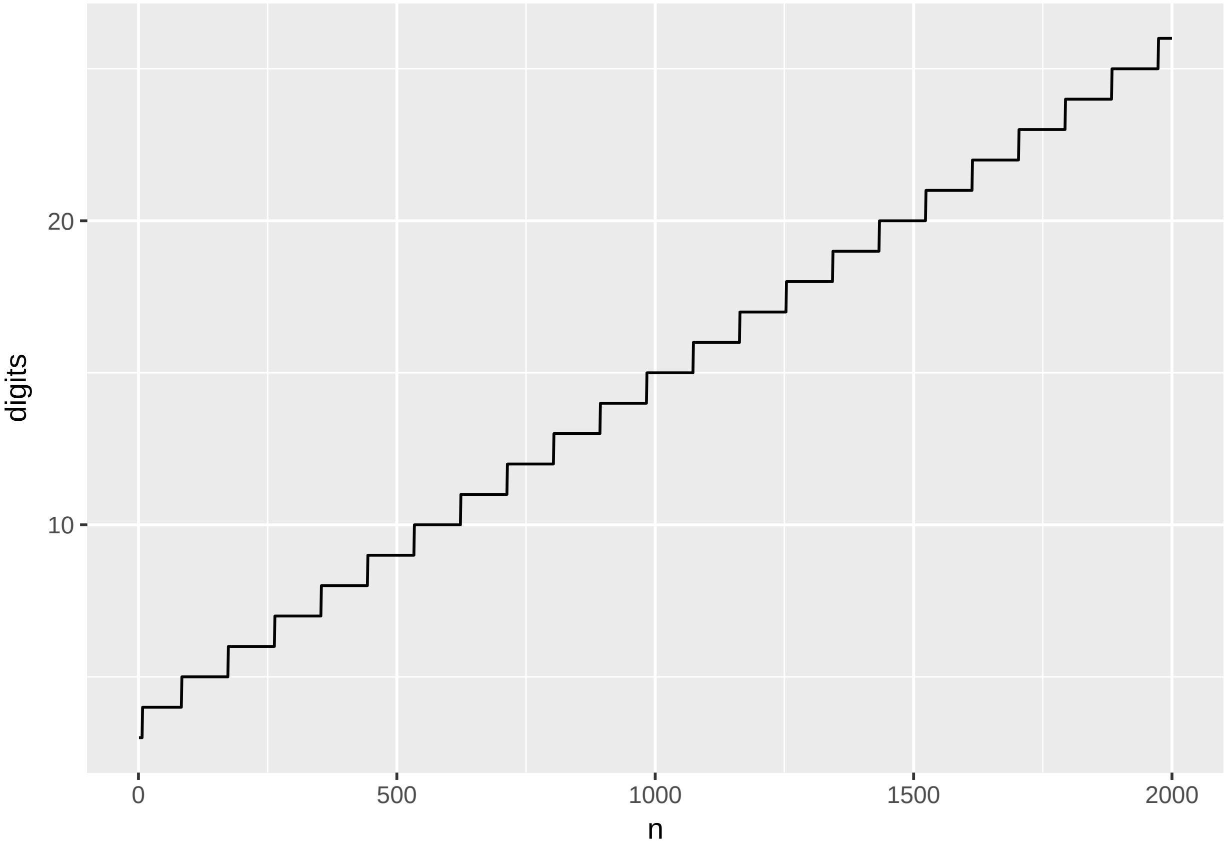
```
##  [1] "110"  "121"  "152"  "240"  "251"  "282"  "390"  "1010" "1021" "1052"
## [11] "1103" "1174" "1265" "1376" "1507" "1658" "1829" "2040" "2051" "2082"
## [21] "2133" "2204" "2295" "2406" "2537" "2688" "2859" "3090" "3101" "3132"
## [31] "3183" "3254" "3345" "3456" "3587" "3738" "3909" "4160" "4171" "4202"
## [41] "4253" "4324" "4415" "4526" "4657" "4808" "4979" "5250" "5261" "5292"
## [51] "5343" "5414" "5505" "5616"
```

4 A274945 terms with the same number of digits

```
# A274945 data from OEIS bfile
data <- A274945$bfile$data
# Adding number of decimal digits by term
data$digits <- nchar(data$A274945)
```

4.1 Plotting decimal digits

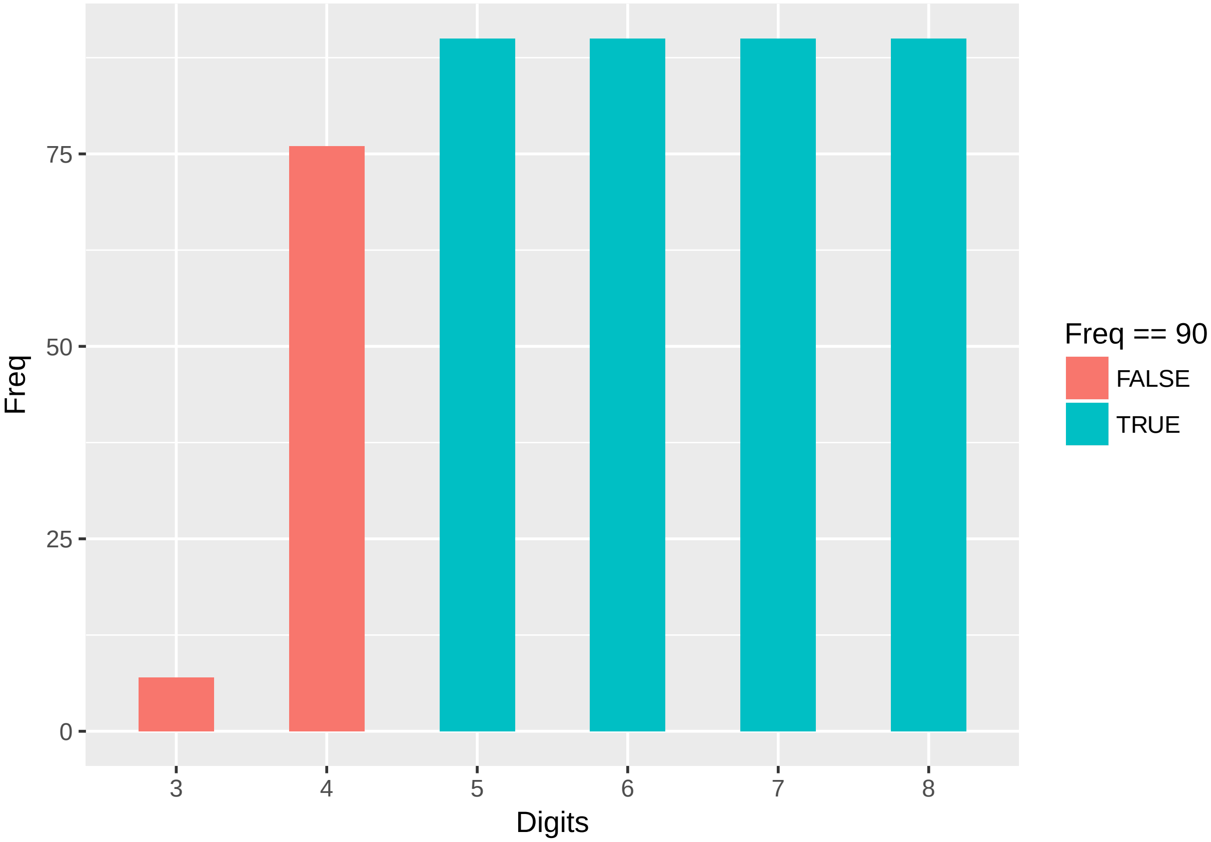
```
ggplot(head(data, 2000), aes(x = n, y = digits)) +
  geom_line()
```



4.2 Plotting decimal digits frequency

```
# Digits Frecuency
df <- data.frame(table(data$digits))
names(df) <- c("Digits", "Freq")

ggplot(head(df), aes(x = Digits, y = Freq, fill = Freq  == 90)) +
  geom_col(width = 0.5)
```



Observation:

- There are 90 terms of the sequence with the same number of decimal digits, except when the number of digits is below 5.
- Thus, the sequence is infinite.

The sequence definition implies that the terms are of the form:

$$a(n) = i(n) \cdot 10^{(D_{10}(n)-1)} + 10 \cdot (i(n)^2 + j(n)^2) + j(n)$$

Where:

- $a(n)$ is n-th term of `A274945` sequence.
- $D_{10}(n)$ is the number of decimal digits of $a(n)$
- $i(n)$ is the first decimal digit of $a(n)$.
- $j(n)$ is the last decimal digit of $a(n)$.

This is easy to justify because:

$$i(n) \in \{1, 2, \ldots, 9\}$$
$$j(n) \in \{0, 1, \ldots, 9\}$$

And then $9 \cdot 10 = 90$

5 Number of decimal digits.

The first term with n digits is of the form 100. . . 0010, with zeroes filling the figure.

```
head(data[grep("1.*10$", data$A274945), ])
```

```
##       n  A274945 digits
## 1     1      110      3
## 8     8     1010      4
## 84   84    10010      5
## 174 174   100010      6
## 264 264  1000010      7
## 354 354 10000010      8
```

This implies that the function $D_{10}(n)$ some rounding function like:

$$D_{10}(n) = \left\lfloor \frac{n+366}{90} \right\rfloor - [n < 8]$$

If n < 8 then $D_{10}(n) = 3$

The symbol [] is 0 if the condition inside is false and 1 if true.

```
D_10 <- function(n) {
  floor((n + 366) / 90) + ifelse(n < 8, -1, 0)
}
```

5.1 Testing Decimal digits formula.

```
data$D10 <- D_10(data$n)
sum(data$D10 == data$digits) / length(data$digits)
```

```
## [1] 1
```

It holds for all data.

6 Last decimal digit

$j(n)$ can be found easily from the last digit of n, when n > 83.

$$j(n) = (n-4) - 10 \cdot \left\lfloor \frac{n-4}{10} \right\rfloor (n \geq 84)$$

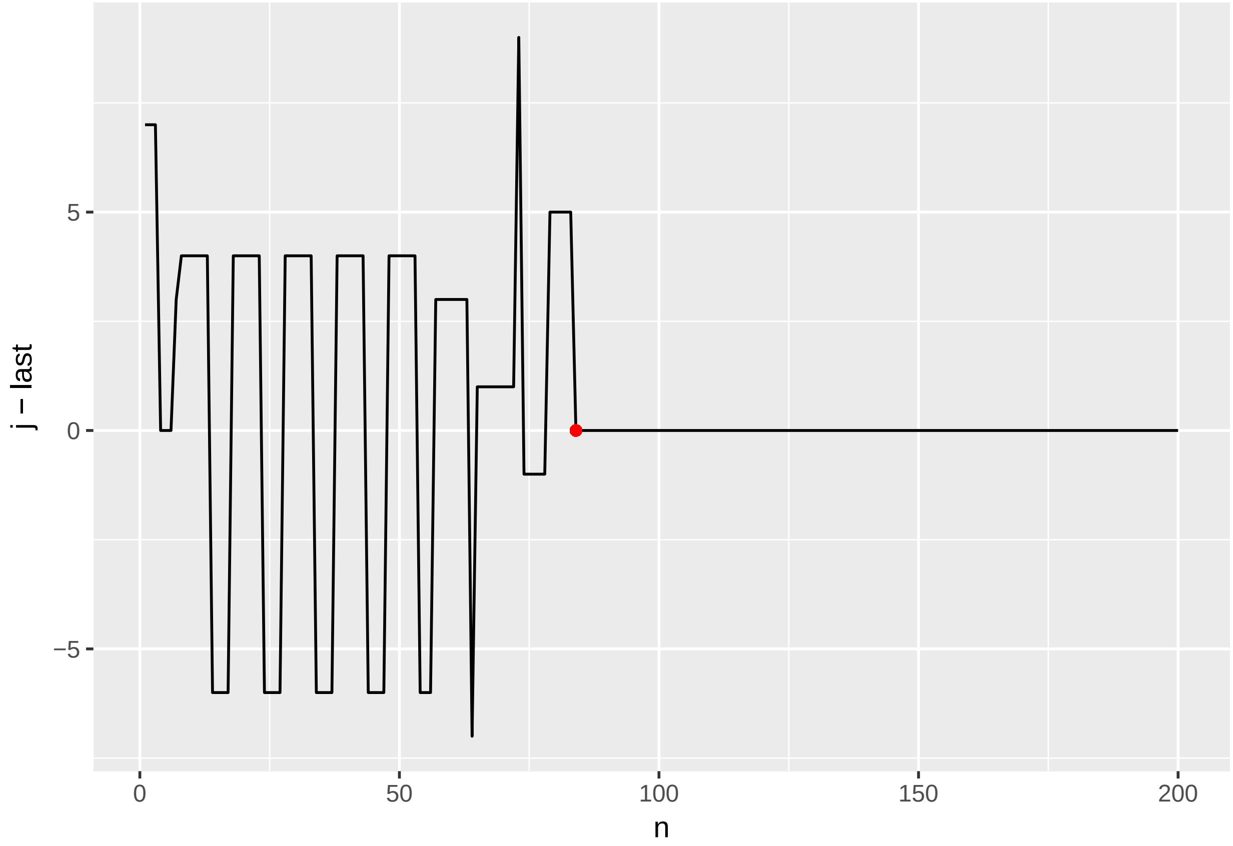
```
j <- function(n) {
  (n - 4) %% 10
}

data$j <- j(data$n)
```

6.1 Test Validity of j(n)

```
data$last <- as.numeric(
  lapply(data$A274945, function(x)
    substr(x, nchar(x) , nchar(x))))

# Plot calculated data less sequence data
ggplot(head(data, 200), aes(x = n, y = j - last)) +
  geom_line() +
  geom_point(x = 84, y = 0, color = "red")
```



7 First decimal digit

$$i(n) = 1 + \left\lfloor \frac{n - 90 \cdot D_{10}(n) + 366}{10} \right\rfloor (n \geq 79)$$

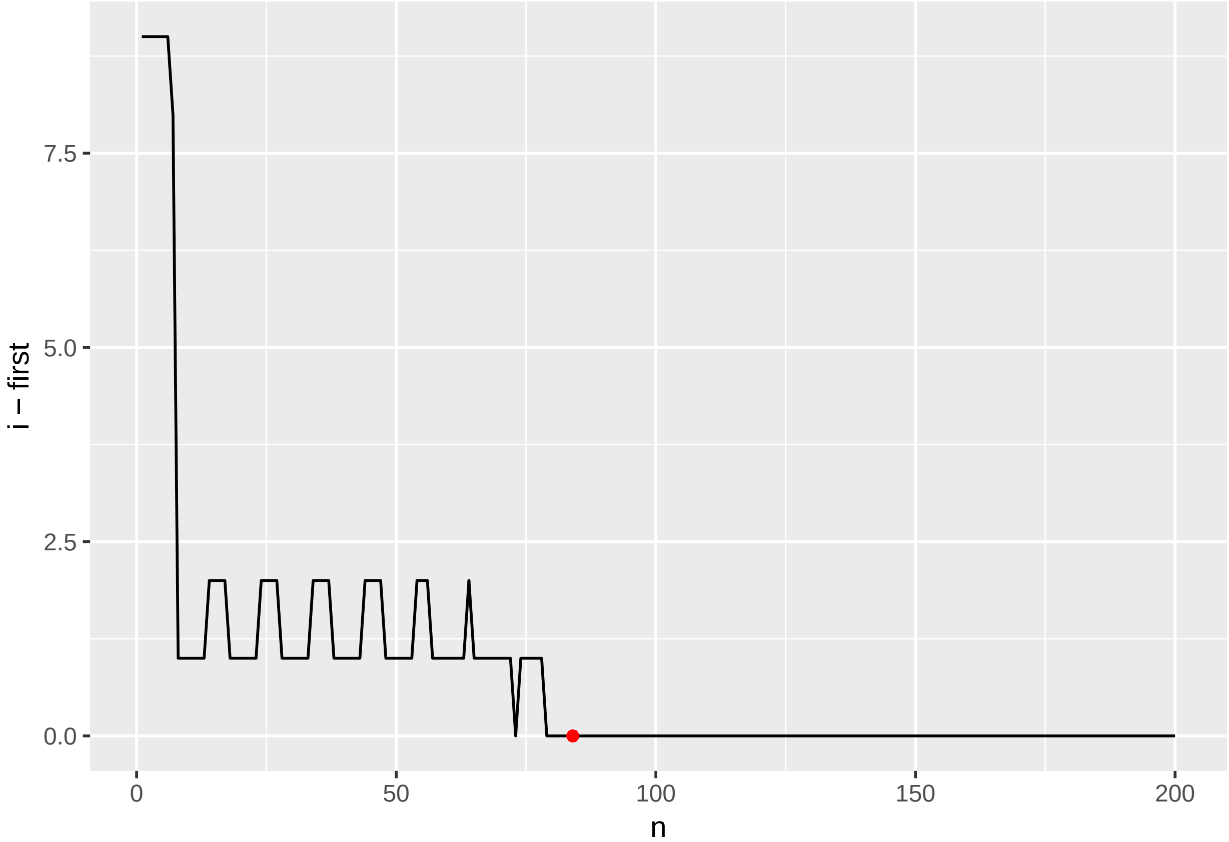
```
i <- function(n) {
  1 + floor(((n - 90 * D_10(n) + 366) / 10))
}

data$i <- i(data$n)
```

7.1 Test Validity of i(n)

```
data$first <- as.numeric(
  lapply(data$A274945, function(x)
    substr(x, 1 , 1)))

# Plot calculated data less sequence data
ggplot(head(data, 200), aes(x = n, y = i - first)) +
  geom_line() +
  geom_point(x = 84, y = 0, color = "red")
```



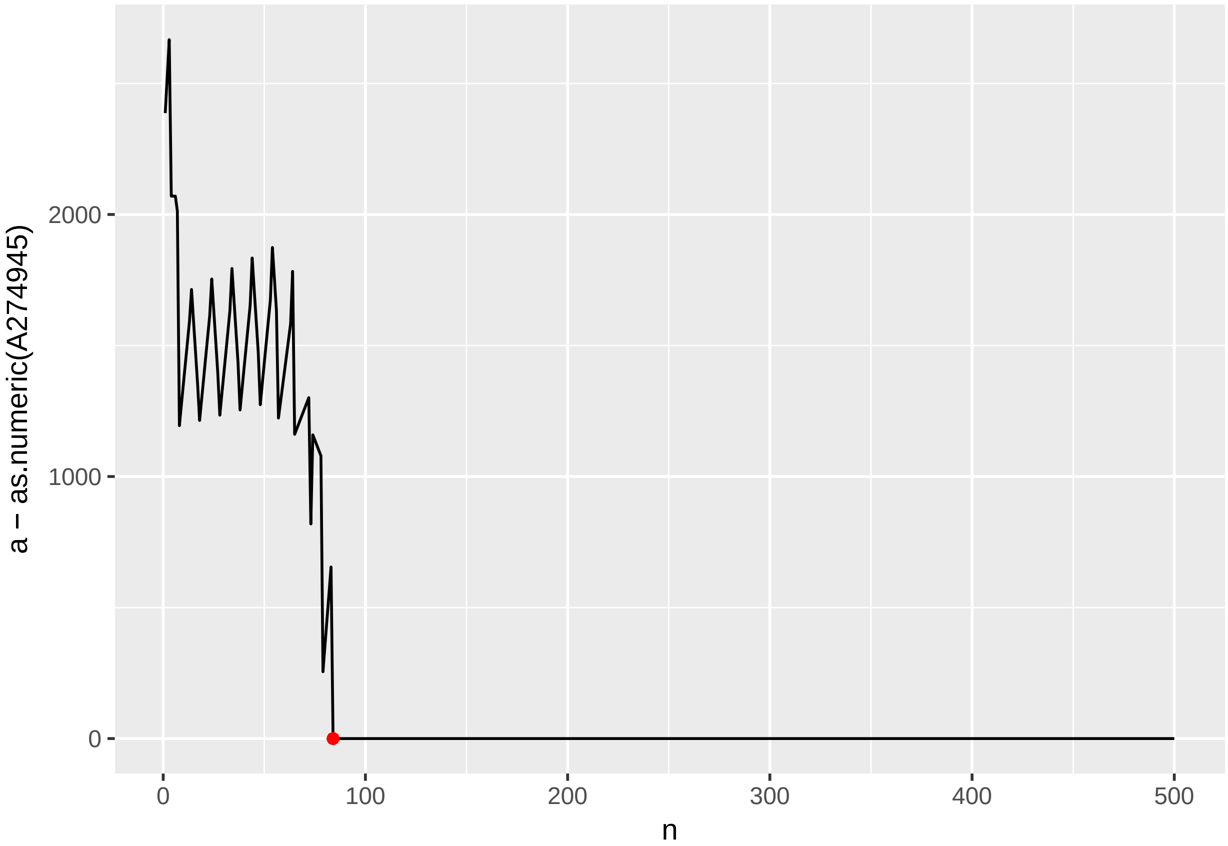
8 Formula for the sequence

```
a <- function(n) {
  i(n) * 10 ^ (D_10(n) - 1) + 10 * (i(n)^2 + j(n)^2) + j(n)
}

data$a <- a(data$n)
```

8.1 Test Validity of a(n)

```
# Plot calculated data less sequence data
ggplot(head(data, 500), aes(x = n, y = a - as.numeric(A274945))) +
  geom_line() +
  geom_point(x = 84, y = 0, color = "red")
```



9 Generate Sequence as string for larger numbers

The method to speed up calculations and to aviod handling multiprecision numbers is to calculate all sequence terms using strings and not numbers.

The strategy is as follows:

1) Find all terms bellow $a(84) = 10010$ by brute force: because, at the beginning, the staircase sequence is very irregular.
2) Construct all terms that have more than 4 decimal digits using character strings
3) Bind everything all together.

9.1 Code for covenience functions

```r
# Check if number n belogs to staircase sequence
# This formula is used to brute force calculations.
isA274945 <- function(n) {
  n_string <- as.character(n)
  digits_number <- nchar(n_string)
  first_digit <- substr(n_string, start = 1L, stop = 1L)
  last_digit <-
    substr(n_string, start = digits_number, digits_number)
  inner_digits <- substr(n_string, start = 2, digits_number - 1L)
  first_digit <- as.integer(first_digit)
  last_digit <- as.integer(last_digit)
  inner_digits <- as.integer(inner_digits)
  answer <- first_digit ^ 2 + last_digit ^ 2 == inner_digits
  ifelse(is.na(answer), FALSE, answer)
}

# Order of the first sequence term with k decimal digits
# This function is used to generate terms.
first_k_digits_term <- function(k) {
  if (k < 5) {
    stop("To find terms with less digits than 5, use bruteforce")
  }
  (k - 5) * 90 + 84
}

# Construct a data.frame with the 90 terms that have k decimal digits
staircase_k_digits <- function(k) {
  A <- expand.grid(i = 1:9, j = 0:9)
  A$n <- first_k_digits_term(k) + 0:89
  A$central_digits <-
    mapply(function(x, y)
      as.character(x ^ 2 + y ^ 2), A$i, A$j)
  A$central_digits_nchar <- nchar(A$central_digits)
  A$central_zeroes <-
    sapply(k - 2 - A$central_digits_nchar, function(x)
      paste0(rep("0", times = x), collapse = ""))
  A$A274945 <- paste0(A$i, A$central_zeroes, A$central_digits, A$i)
  data.frame("n" = A$n, "A274945" = A$A274945)
}

# Find all terms that are in the sequence less than x, by brute force.
bruteforce_find_terms <- function(x) {
  if (x < 110) {
    stop("no terms below x value, try a higher x")
  }
  A <- 110:x
  A <- A[isA274945(A)]
  data.frame("n" = 1:length(A), "A274945" = A)
}


# Find all terms that have k or less decimal digits
staircase_up_to_k_digits <- function(k) {
  first_terms <- bruteforce_find_terms(10010 - 1)
  digits <- 5:k
  last_terms <- lapply(digits, function(k) staircase_k_digits(k))
  last_terms <- do.call(rbind, last_terms)
  rbind(first_terms, last_terms)
}
```

9.2 Find all terms that have less than a given number of digits.

```
k_digits <- 6
sequence_A274945 <- staircase_up_to_k_digits(k_digits)
```

9.2.1 First terms

```
head(sequence_A274945)
```

```
##   n A274945
## 1 1     110
## 2 2     121
## 3 3     152
## 4 4     240
## 5 5     251
## 6 6     282
```

9.2.2 Last terms

```
tail(sequence_A274945)
```

```
##       n A274945
## 258 258  400974
## 259 259  501065
## 260 260  601176
## 261 261  701307
## 262 262  801458
## 263 263  901629
```