THE PUBLIC IS MORE FAMILIAR WITH BAD DESIGN THAN GOOD DESIGN. IT IS, IN EFFECT, CONDITIONED TO PREFER BAD DESIGN, BECAUSE THAT IS WHAT IT LIVES WITH. THE NEW BECOMES THREATENING, THE OLD REASSURING.

PAUL RAND

A DESIGNER KNOWS THAT HE HAS ACHIEVED PERFECTION NOT WHEN THERE IS NOTHING LEFT TO ADD, BUT WHEN THERE IS NOTHING LEFT TO TAKE AWAY.

ANTOINE DE SAINT-EXUPÉRY

. . . THE DESIGNER OF A NEW SYSTEM MUST NOT ONLY BE THE IMPLEMENTOR AND THE FIRST LARGE-SCALE USER; THE DESIGNER SHOULD ALSO WRITE THE FIRST USER MANUAL. . . IF I HAD NOT PARTICIPATED FULLY IN ALL THESE ACTIVITIES, LITERALLY HUNDREDS OF IMPROVEMENTS WOULD NEVER HAVE BEEN MADE, BECAUSE I WOULD NEVER HAVE THOUGHT OF THEM OR PERCEIVED WHY THEY WERE IMPORTANT.

DONALD E. KNUTH

HENGFENG WEI

COLLECTION OF ALGORITHMS PSEUDOCODE

ANT

PUBLISHED BY ANT

TUFTE-LATEX.GOOGLECODE.COM

First printing, April 2018

Contents

List of Figures

List of Tables

Dedicated to those who appreciate LaTeX

and the work of Edward R. Tufte and Donald E. Knuth.

Introduction

This is a collection of psuedocode for classic algorithms.

Sorting

Algorithm 1 Selection Sort.

procedure SELECTION-SORT(A, n)
    for $i \leftarrow 1$ **to** $n - 1$ **do**
        for $j \leftarrow i + 1$ **to** n **do**
            if $A[j] < A[i]$ **then**
                SWAP$(A[j], A[i])$

Dynamic Programming

Algorithm 2 Computing $\binom{n}{k}$.

```
procedure BINOM(n, k)                         ▷ Required: n ≥ k ≥ 0
    if k = 0 ∨ n = k then
        return 1
    return BINOM(n − 1, k) + BINOM(n − 1, k − 1)
```

Algorithm 3 Computing $\binom{n}{k}$.

```
procedure BINOM(n, k)                         ▷ Required: n ≥ k ≥ 0
    for i ← 0 to n do
        B[i][0] ← 1
        B[i][i] ← 1
    for i ← 2 to n do
        for j ← 1 to k do
            B[n][k] ← B[n − 1][k] + B[n − 1][k − 1]
    return B[n][k]
```

Bibliography