



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 ALGO- RITHMS PSEUDOCODE

ANT

Copyright © 2018 Hengfeng Wei

PUBLISHED BY ANT

TUFTE-LATEX.GOOGLECODE.COM

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*First printing, April 2018*

# *Contents*

*Sorting*      15

*Dynamic Programming*      17

*Bibliography*      19



## *List of Figures*





## *List of Tables*



*Dedicated to those who appreciate  $\text{\LaTeX}$   
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.

---

```
1: procedure SELECTION-SORT( $A, n$ )
2:   for  $i \leftarrow 1$  to  $n - 1$  do
3:     for  $j \leftarrow i + 1$  to  $n$  do
4:       if  $A[j] < A[i]$  then
5:         SWAP( $A[j], A[i]$ )
```

---





# Dynamic Programming

---

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

---

```
1: procedure BINOM( $n, k$ )                                ▷ Required:  $n \geq k \geq 0$ 
2:   if  $k = 0 \vee n = k$  then
3:     return 1
4:   return BINOM( $n - 1, k$ ) + BINOM( $n - 1, k - 1$ )
```

---

---

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

---

```
1: procedure BINOM( $n, k$ )                                ▷ Required:  $n \geq k \geq 0$ 
2:   for  $i \leftarrow 0$  to  $n$  do
3:      $B[i][0] \leftarrow 1$ 
4:      $B[i][i] \leftarrow 1$ 
5:   for  $i \leftarrow 2$  to  $n$  do
6:     for  $j \leftarrow 1$  to  $k$  do
7:        $B[n][k] \leftarrow B[n - 1][k] + B[n - 1][k - 1]$ 
8:   return  $B[n][k]$ 
```

---



## *Bibliography*