Sorting Pairs in Bins—an Improved Upper Bound

Felix Kälberer*, Matthias Nieser*, Ulrich Reitebuch*

December 29, 2008 (German original†)

English translation by Felix Kälberer, July 8, 2010

Abstract

The following pages describe new insights for the *Sorting Pairs in Bins* problem. We can't give an optimal algorithm, but we improve the currently known upper bound to $\lfloor 2/3\, n^2 - 2/3 \rfloor$ swaps. The specified algorithm is at most $n/3$ swaps from the optimum. For $n = 2^j + 1$, $j \in \mathbb{N}$, the algorithm is optimal and uses only $2/3n^2 - n/3$ swaps.

1 Notation

Given n pots, which are numbered in ascending order from 0 to $n-1$. Each pot i contains two balls with the numbers $2n-2i-1$ and $2n-2i-2$. The goal is to sort the balls so that pot i contains the balls numbered $2i$ and $2i+1$. This numbering differs slightly from the original numbering in the *Krawattenrätsel*[1] (there are two balls with the same number in each pot), but the problem is equivalent.

For a clearer description of the algorithm, each pot is divided in the two halves *over* (o) and *under* (u), in each of which is a ball. A permutation is represented by two tuples of the form $(i, o/u) \leftrightarrow (i+1, o/u)$. Here, the upper/lower ball from pot i is exchanged with the upper/lower ball of pot $i+1$. The abbreviating notation

$$i \xrightarrow{o/u} j$$

Means that the upper (or lower) ball from pot i is pushed along the upper (lower) row to pot j. This operation needs $|j-i|$ swaps.

In some places, the operation `flip(i,j)` is used. It will swap the upper vs. the lower ball in pots i to j. This operation is not counted as a swap operation, since it does not matter if a ball is on top or at the bottom. Its only for a clearer notation.

2 The Algorithm

The algorithm consists of a main method `krawattenloeser(n)` and a recursive in `invert(n)` method. The main method inverts order of the balls the $2n$ pots. It calls the recursive method, that solves the respective subproblem. The detailed operation of the methods are explained in Sections 3 and 4. The pseudo-code of the methods is as follows:

*Freie Universität Berlin, Institut fur Mathematik und Informatik
supported through DFG Research Center Matheon *Mathematics for key technologies*
`<kaelberer,nieser,reitebuch>@mi.fu-berlin.de`

†The original version was submitted to on December 30, 2008 to a prize competition held by Spektrum Akademischer Verlag and the German Mathematical Society (DMV), under technical supervision of Prof. Stefan Felsner, Techn. Univ. Berlin and Prof. Volker Kaibel, Univ. Magdeburg. This submission won one of two first prizes.

[1]For unknown reasons, the problem *Sorting Pairs in Bins* goes by the name *Krawattenrätsel* in German, which translates to *necktie problem*. This might be due to a wrong translations of one of the many meanings of *tie*.

Algorithm 1 Krawattenlöser(n)

if $n == 1$ **then**
  return
end if
Swap $0 \xrightarrow{u} n-1$
Call `invert`$(n-1)$

Algorithm 2 invert(k) for even k

Swap $(0, o) \leftrightarrow (1, u)$
Swap $1 \xrightarrow{u} k-1$
Swap $(k-1, u) \leftrightarrow (k, o)$
for $j = 1$ to $k/2 - 1$ **do**
  Swap $(j, o) \leftrightarrow (j+1, u)$
  Swap $j+1 \xrightarrow{u} k-j$
end for
flip$(0, k-1)$
Call `invert`$(k/2)$
flip$(0, k/2)$
for $j = 0$ to $k/2 - 2$ **do**
  Swap $k/2 - j \xrightarrow{u} k-1-2j$
  Swap $(k/2 - j - 1, o) \leftrightarrow (k/2 - j, u)$
  Swap $k/2 - j \xrightarrow{u} k-2-2j$
end for

Algorithm 3 invert(k) for odd k

if $k == 1$ **then**
  Swap $(0, o) \leftrightarrow (1, o)$
  return
end if
Swap $(0, o) \leftrightarrow (1, u)$
Swap $1 \xrightarrow{u} k-1$
Swap $(k-1, u) \leftrightarrow (k, o)$
for $j = 1$ to $(k-1)/2$ **do**
  Swap $(j, o) \leftrightarrow (j+1, u)$
  Swap $j+1 \xrightarrow{u} k-j$
end for
flip$(0, k-1)$
Call `invert`$((k-1)/2)$
for $j = 0$ to $(k-3)/2$ **do**
  Swap $((k-1)/2 - j, o) \leftrightarrow ((k-1)/2 - j + 1, u)$
  Swap $(k-1)/2 - j + 1 \xrightarrow{u} k-1-2j$
  Swap $(k-1)/2 - j - 1 \xrightarrow{u} k-2-2j$
end for
flip$(0, 0)$

3 Krawattenlöser

At the beginning the balls in the n are distributed as follows:

2n-2	2n-4	...	2	0
2n-1	2n-3	...	3	1

A call of `krawattenloeser(`n`)` makes it:

1	3	...	2n-3	2n-1
0	2	...	2n-4	2n-2

Lines 1–3 :
Treats the special case for $n = 1$.

Line 4 :
Pushes the ball $2n - 1$ along the bottom row to pot $n - 1$ (needs $n - 1$ swaps):

2n-2	2n-4	...	2	0
2n-3	2n-5	...	1	2n-1

Line 5 :
Calls the recursive method `invert(`$n - 1$`)`, that sorts the balls in pots 0 to $n - 2$, and the upper ball of Pot $n - 1$ correctly (see Section 4).

4 The method `invert(`k`)`

The main part of the Krawattenlöser is the recursive method `invert(`k`)`. It operates only on the balls in pots from 0 to $k - 1$, and on the upper ball of pot k. The configuration when calling the method is as follows:

2k	2k-2	...	2	0
2k-1	2k-3	...	1	

The method makes it:

0	2	...	2k-2	2k
1	3	...	2k-1	

The method is different for even and odd k. The function of the two algorithms is described in the following sections.

4.1 `invert(`k`)` for even k

Lines 1–3 :
Push the ball $2k$ along the bottom row to their final position in pot k.

2k-3	2k-2	2k-4	...	4	2	2k
2k-1	2k-5	2k-7	...	1	0	

This needs k swaps.

Lines 4–7 :

All other upper balls in pots 1 to $k/2-1$ are now successively pushed along the bottom row to their final positions in pots $k-1$ to $k/2+1$. First of all ball $2k-2$:

2k-3	2k-7	2k-4	2k-6	...	6	4	2	2k
2k-1	2k-5	2k-9	2k-11	...	1	0	2k-2	

This needs $k-2$ swaps.

Then ball $2k-4$ ($k-4$ swaps), and so on. The ball $k+2$ is the last in this loop and will be moved two positions to the right. The distribution is then:

2k-3	2k-7	2k-11	...	1	k	k-2	...	4	2	2k
2k-1	2k-5	2k-9	...	3	0	k+2	...	2k-4	2k-2	

The number of permutations in lines 7-10 is:

$$\sum_{i=1}^{k/2-1} (k-2i) = \frac{1}{4}k^2 - \frac{1}{2}k.$$

Line 8 :

The upper and lower balls in pots 0 to $k-1$ are swapped. This doesn't count as a swap operation, as the upper and lower positions are equivalent. This happens only to provide an easier description.

2k-1	2k-5	2k-9	...	3	0	k+2	...	2k-4	2k-2	2k
2k-3	2k-7	2k-11	...	1	k	k-2	...	4	2	

Line 9 :

Calls the method recursively. This reverses the order of the balls in pots 0 to $k/2-1$, and the upper ball in pot $k/2$.

0	3	7	...	2k-5	2k-1	k+2	...	2k-4	2k-2	2k
1	5	9	...	2k-3	k	k-2	...	4	2	

Line 10 :

Swaps the upper and lower balls in pots 1 to $k/2$.

1	5	9	...	2k-3	k	k+2	...	2k-4	2k-2	2k
0	3	7	...	2k-5	2k-1	k-2	...	4	2	

Lines 11–15 :

This loop successively moves the balls $2k-1$,$2k-3$,...,5 along the lower row to their final position. First the ball $2k-1$ (using $k/2-1$ swaps).

1	5	9	...	2k-3	k	k+2	...	2k-4	2k-2	2k
0	3	7	...	2k-5	k-2	k-4	...	2	2k-1	

Then ball $2k-3$ ($k/2-1$ swaps):

1	5	9	...	2k-7	k-2	k	k+2	...	2k-6	2k-4	2k-2	2k
0	3	7	...	2k-9	2k-5	k-4	k-6	...	2	2k-3	2k-1	

Balls $2k$ to $2k-4$ are now at their final position. The left $k-2$ pots together with the upper ball from pot $k-2$ are exactly in the same configuration as the situation after line 10, except that now k is two less.

So when the lines 12 to 14 in the loop are executed, all balls are successively sorted into their proper position. The last action is to swap ball 5 with ball 2 in one operation. After that, all balls are sorted in ascending order.

The number of swaps of the whole loop is:

$$\sum_{i=0}^{k/2-2} \left(\left(\frac{1}{2}k - 1 - i \right) + \left(\frac{1}{2}k - 1 - i \right) \right) = \frac{1}{4}k^2 - \frac{1}{2}k$$

4.2 *invert(k)* for odd k

Lines 1–4 :
Treats the special case for $k = 1$, to end the recursion.

Lines 5–11 :
Just as for odd k. The upper balls from pots 0 to $(k-1)/2$ will be pushed along the lower row into pots k to $(k+1)/2$. The last ball $k+1$ is swapped with one operation to position $(k+1)/2$.

2k-3	2k-7	2k-11	...	3	0	k-1	k-3	...	4	2	2k
2k-1	2k-5	2k-9	...	5	1	k+1	k+3	...	2k-4	2k-2	

The number of swap operations in lines 5–11 is:

$$\sum_{i=0}^{(k-1)/2} (k - 2i) = \frac{1}{4}k^2 + \frac{1}{2}k + \frac{1}{4}.$$

Line 12 :
Swaps the upper and lower balls in the pots from 0 to $k-1$.

2k-1	2k-5	2k-9	...	5	1	k+1	k+3	...	2k-4	2k-2	2k
2k-3	2k-7	2k-11	...	3	0	k-1	k-3	...	4	2	

Line 13 :
Recursive application of the algorithm to the left $(k-1)/2$ pots (and the opper ball of pot $(k-1)/2$).

1	5	...	2k-9	2k-5	2k-1	k+1	k+3	...	2k-4	2k-2	2k
3	7	...	2k-7	2k-3	0	k-1	k-3	...	4	2	

Lines 14–18 :

The balls $2k-1$,$2k-3$,...,3 are pushed along the bottom rows to their final positions. First the ball $2k-1$ (using $(k-1)/2$ swaps).

1	5	...	2k-9	2k-5	k-1	k+1	k+3	...	2k-4	2k-2	2k
3	7	...	2k-7	2k-3	0	k-3	k-5	...	2	2k-1	

Then balls $2k-3$ ($(k-3)/2$ swaps):

1	5	...	2k-9	2k-5	k-1	k+1	k+3	...	2k-6	2k-4	2k-2	2k
3	7	...	2k-7	0	k-3	k-5	k-7	...	2	2k-3	2k-1	

Balls $2k$ to $2k-4$ are now in their correct position. The left $k-2$ pots together with the upper ball from pot $k-2$ resemble the same situation as after line 13, except that k is two less.

So now if lines 14–18 in the loop are executed, all balls are successively to their right position. The last ball is ball 3 in pot 0 that is exchanged with ball 0 in pot 1 with one swap operation.

1	2	4	...	k-3	k-1	k+1	k+3	...	2k-2	2k
0	3	5	...	k-2	k	k+2	k+4	...	2k-1	

The number of swap operations in the whole loop is:

$$\sum_{i=0}^{(k-3)/2} \left(\left(\frac{k-1}{2} - i \right) + \left(\frac{k-1}{2} - i \right) \right) = \frac{1}{4}k^2 - \frac{1}{4}.$$

Line 19 :

Swaps the upper and lower ball in pot 0.

5 Counting the swaps

Adding the swaps in Section 4, one obtains the total cost of the `invert` method:

for even k :

$$i(k) = \frac{1}{2}k^2 + i\left(\frac{k}{2}\right). \tag{1}$$

for odd k :

$$i(k) = \frac{1}{2}k^2 + \frac{1}{2}k + i\left(\frac{k-1}{2}\right). \tag{2}$$

The total cost of the *Krawattenlöser* is as follows:

Case 1: $n = 2^j + 1, j \in \mathbb{N}$

In this case the method `invert(`$k = 2^j$`)` is called. Thus, k is even for all recursive calls. With complete induction by the recursion formula (1), the following equation for the cost is easily veryfied:

$$i(k) = \frac{2}{3}k^2 + \frac{1}{3}.$$

Therefore, the *Krawattenlöser* needs the following number of steps:

$$k(n) = n - 1 + i(n-1) = \frac{2}{3}n^2 - \frac{1}{3}n$$

This matches the lower bound and is thus optimal.

Case 2: n arbitrary

For arbitrary n, the algorithm is not necessarily optimal.

One can derive with formulas (1) and (2) via complete induction:

$$i(k) \leq \frac{2}{3}k^2 + \frac{1}{3}k.$$

Thus, the *Krawattenlöser* needs at most

$$k(n) = n - 1 + i(n-1) \leq \frac{2}{3}n^2 - \frac{2}{3}$$

swaps.