2. THE COLUMNSORT ALGORITHM

The description of the basic Columnsort algorithm appears in Leighton [6]. Columnsort begins with $n = r \cdot s$ values placed in an $r \times s$ matrix, where $r \,\%\, s = 0$ and $r \geq 2(s-1)^2$. When complete, Columnsort will produce an $r \times s$ matrix of values. When accessed in column–major order, the values will be in ascending order.

Eight steps (numbered 1 — 8) comprise Columnsort. The odd–numbered steps sort each column of the matrix, while the even–numbered steps relocate the values. The step details are summarized in Table 1. The algorithm's steps are not complex, but their actions leave room for plenty of implementation creativity.

The following example clarifies most of the ambiguities introduced by the brevity of Table 1. Consider the collection of twenty-seven values (from 1 to 27, inclusive) arranged in the 9×3 matrix shown in Figure 1. This arrangement is acceptable because $27 = 9 \cdot 3$, $9 \,\%\, 3 = 0$, and $9 \geq 2(3-1)^2$.

$$\xrightarrow{\text{Start}} \begin{bmatrix} 14 & 25 & 9 \\ 3 & 1 & 27 \\ 21 & 12 & 7 \\ 24 & 23 & 16 \\ 8 & 13 & 18 \\ 26 & 4 & 5 \\ 19 & 17 & 22 \\ 10 & 15 & 11 \\ 2 & 20 & 6 \end{bmatrix}$$

Figure 1: Initial collection of values to be sorted

Step 1 sorts each of the three columns of values. Thanks to the $r \,\%\, s = 0$ rule, the transposition of Step 2 can be performed without special cases. After the columns are sorted in Step 3, Step 4 rearranges the values from rows back into columns. Figure 2 shows the effects of these actions.

The conclusion of the sort is depicted in Figure 3. After the columns are sorted in Step 5, the values are very close to being in their final positions. However, some of the values at the bottom of column c should be at or near the top of column $c+1$, and vice-versa. The shifting and unshifting operations of Steps 6 and 8 will take care of those stragglers. In Step 6, the values are shifted by $\lfloor \frac{r}{2} \rfloor = \lfloor \frac{9}{2} \rfloor = 4$ places, sufficient to place the appropriate out–of–position values together in the same columns. The sorts of Step 7 and the reversal of the shift produces the final sorted matrix.

An informal proof of the correctness of Columnsort is provided in [6].

Step	Action
1	Sort the values in each column.
2	*Transpose*: Access the values in column–major order (CMO); place them back in the the matrix in row major order (RMO).
3	Sort the values in each column.
4	*Untranspose*: The reverse of Step 2; access in RMO and reinsert in CMO.
5	Sort the values in each column.
6	*Shift*: In a CMO sense, shift each value forward by $\lfloor \frac{r}{2} \rfloor$ positions. This creates an extra column; fill the vacated positions with $-\infty$ and the excess positions in the new column with $+\infty$.
7	Sort the values in each column.
8	*Unshift*: Undo Step 6; that is, shift each value back by $\lfloor \frac{r}{2} \rfloor$ positions, eliminating the extra column and the $-\infty$ and $+\infty$ values.

Table 1: The Eight Steps of Columnsort

$$\xrightarrow{\text{Step 1}} \begin{bmatrix} 2 & 1 & 5 \\ 3 & 4 & 6 \\ 8 & 12 & 7 \\ 10 & 13 & 9 \\ 14 & 15 & 11 \\ 19 & 17 & 16 \\ 21 & 20 & 18 \\ 24 & 23 & 22 \\ 26 & 25 & 27 \end{bmatrix} \quad \xrightarrow{\text{Step 2}} \begin{bmatrix} 2 & 3 & 8 \\ 10 & 14 & 19 \\ 21 & 24 & 26 \\ 1 & 4 & 12 \\ 13 & 15 & 17 \\ 20 & 23 & 25 \\ 5 & 6 & 7 \\ 9 & 11 & 16 \\ 18 & 22 & 27 \end{bmatrix}$$

$$\xrightarrow{\text{Step 3}} \begin{bmatrix} 1 & 3 & 7 \\ 2 & 4 & 8 \\ 5 & 6 & 12 \\ 9 & 11 & 16 \\ 10 & 14 & 17 \\ 13 & 15 & 19 \\ 18 & 22 & 25 \\ 20 & 23 & 26 \\ 21 & 24 & 27 \end{bmatrix} \quad \xrightarrow{\text{Step 4}} \begin{bmatrix} 1 & 9 & 18 \\ 3 & 11 & 22 \\ 7 & 16 & 25 \\ 2 & 10 & 20 \\ 4 & 14 & 23 \\ 8 & 17 & 26 \\ 5 & 13 & 21 \\ 6 & 15 & 24 \\ 12 & 19 & 27 \end{bmatrix}$$

Figure 2: Results of Steps 1 through 4

$$\xrightarrow{\text{Step 5}} \begin{bmatrix} 1 & 9 & 18 \\ 2 & 10 & 20 \\ 3 & 11 & 21 \\ 4 & 13 & 22 \\ 5 & 14 & 23 \\ 6 & 15 & 24 \\ 7 & 16 & 25 \\ 8 & 17 & 26 \\ 12 & 19 & 27 \end{bmatrix} \quad \xrightarrow{\text{Step 6}} \begin{bmatrix} -\infty & 6 & 15 & 24 \\ -\infty & 7 & 16 & 25 \\ -\infty & 8 & 17 & 26 \\ -\infty & 12 & 19 & 27 \\ 1 & 9 & 18 & +\infty \\ 2 & 10 & 20 & +\infty \\ 3 & 11 & 21 & +\infty \\ 4 & 13 & 22 & +\infty \\ 5 & 14 & 23 & +\infty \end{bmatrix}$$

$$\xrightarrow{\text{Step 7}} \begin{bmatrix} -\infty & 6 & 15 & 24 \\ -\infty & 7 & 16 & 25 \\ -\infty & 8 & 17 & 26 \\ -\infty & 9 & 18 & 27 \\ 1 & 10 & 19 & +\infty \\ 2 & 11 & 20 & +\infty \\ 3 & 12 & 21 & +\infty \\ 4 & 13 & 22 & +\infty \\ 5 & 14 & 23 & +\infty \end{bmatrix} \quad \xrightarrow{\text{Step 8}} \begin{bmatrix} 1 & 10 & 19 \\ 2 & 11 & 20 \\ 3 & 12 & 21 \\ 4 & 13 & 22 \\ 5 & 14 & 23 \\ 6 & 15 & 24 \\ 7 & 16 & 25 \\ 8 & 17 & 26 \\ 9 & 18 & 27 \end{bmatrix}$$

Figure 3: Results of Steps 5 through 8