

# How to Sort Numbers in Excel Using the SORT Formula

Authored by  
**stats writer**

February 21, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Sort Numbers in Excel Using the SORT Formula*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=131899>

## The Strategic Importance of Data Sorting in Microsoft Excel

In the contemporary landscape of **data management**, the ability to organize information efficiently is paramount for any professional utilizing a **spreadsheet**. **Microsoft Excel** provides a robust suite of tools designed to transform raw data into actionable insights, and among the most fundamental of these capabilities is the ability to sort numbers. Whether you are dealing with financial records, scientific measurements, or inventory counts, arranging data in a logical sequence allows for quicker identification of trends, outliers, and key performance indicators. By mastering the use of a **formula** to sort numbers, users can move beyond static manual sorting and embrace a more dynamic approach to **data analysis**.

While **Excel** offers built-in sorting buttons within its interface, utilizing a **formula** provides a level of flexibility and automation that manual tools simply cannot match. When data is sorted via a **formula**, the resulting list can update automatically as the source data changes, ensuring that your reports remain current without constant manual intervention. This is particularly useful in complex workbooks where data is being pulled from various sources or where multiple users are inputting information simultaneously. Understanding the underlying logic of these formulas is the first step toward becoming a power user in the spreadsheet environment.

The primary functions involved in these sorting operations are the **SMALL** and **LARGE** functions, which work in tandem with the **ROWS** function to create an incrementing index. This combination allows **Excel** to evaluate a **range** of cells and extract values one by one in a specific order. By leveraging these functions, you can create a secondary list that mirrors your primary data but presents it in a perfectly ordered fashion. This method is highly regarded for its reliability and its ability to maintain **data integrity** across large datasets, making it a staple technique for advanced data science tasks within the office suite.

## Understanding the SORT Function for Dynamic Arrays

Before diving into traditional functions, it is essential to acknowledge the **SORT function**, a modern addition to the Excel engine designed for **dynamic arrays**. This function simplifies the sorting process significantly by allowing users to arrange an entire **array** or **range** with a single, concise **formula**. The **SORT function** is particularly powerful because it can handle multiple columns and sort by specific indices, providing a streamlined alternative to older, more complex nesting techniques. It is the go-to solution for users on **Microsoft 365** or **Excel 2021** and beyond, where efficiency and speed are prioritized.

The syntax for the **SORT function** involves identifying the **range** of data you wish to organize and then specifying the sort order. Specifically, the **formula** uses "1" to denote **ascending order** and "-1" to signify **descending order**. Because this function is part of the **dynamic array** family, the

results "spill" into adjacent cells automatically, removing the need to drag the **formula** down manually. This "spill" behavior is a revolutionary change in how **Excel** handles calculations, ensuring that the output **range** always matches the input **range** in size and scope.

Despite the convenience of the **SORT function**, there are many scenarios where users might still prefer or require the use of **SMALL** and **LARGE**. For instance, in legacy versions of **Excel** or in specific templates where dynamic arrays are restricted, the traditional **formula** approach remains the gold standard. Furthermore, using **SMALL** and **LARGE** gives the user granular control over exactly which values are returned and where they are placed, which can be advantageous in highly customized dashboard designs. Regardless of the specific tool chosen, the goal remains the same: achieving a clear, ordered view of numerical information.

## The Logic of the SMALL Function for Ascending Sorts

To sort numbers from the **smallest to the largest**, we rely on the **SMALL function**. In technical terms, the **SMALL function** returns the k-th smallest value in a data set. For example, if you want the absolute minimum value, you would set 'k' to 1. If you wanted the second smallest, you would set 'k' to 2. When we want to sort an entire column, we need a way to make 'k' increase automatically as we copy the **formula** down the column. This is where the **ROWS function** becomes indispensable, acting as a dynamic counter that starts at 1 and grows with each subsequent row.

The construction of the **ascending sort formula** involves a careful use of **absolute references** and **relative references**. By locking the source **range** with dollar signs (e.g.,  $\$A\$2:\$A\$13$ ), we ensure that **Excel** always looks at the same set of numbers. Meanwhile, the **ROWS function** uses a reference like  $\$A\$2:A2$ , where the first part is locked and the second part is mobile. As you drag the **formula** down, the reference expands to  $\$A\$2:A3$ , then  $\$A\$2:A4$ , causing the **ROWS function** to return 1, then 2, then 3. This sequence tells the **SMALL function** exactly which rank of number to retrieve next.

This method is exceptionally robust because it handles duplicates gracefully. If the number "5" appears twice in your list, the **SMALL function** will return "5" for both the k-th and (k+1)-th smallest positions, maintaining the original frequency of the data. This level of accuracy is critical in financial reporting and **statistical analysis**, where every data point must be accounted for. By understanding this logic, users can troubleshoot errors and adapt the **formula** to fit various **dataset** sizes and configurations.

## Implementing Formula 1: Ascending Order (Smallest to Largest)

To begin organizing your data in **ascending order**, you must first identify the **range** containing your numerical values. In the following examples, we assume your data resides in cells **A2** through

**A13.** You can use the following **formulas** to sort numbers in **Excel** in **ascending** or **descending order**:

**Formula 1: Sort Numbers in Ascending Order (Smallest to Largest)**

**=SMALL(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))**

**Formula 2: Sort Numbers in Descending Order (Largest to Smallest)**

**=LARGE(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))**

Both of these **formulas** will sort the numbers in the **range A2:A13** in either **ascending** or **descending order**, depending on whether you use the **SMALL** or **LARGE** functions in **Excel**. The following examples show how to use each **formula** in practice with the following column of numbers in **Excel**:

	A	B	C	D	E	F
1	<b>Numbers</b>					
2	14					
3	19					
4	30					
5	18					
6	12					
7	9					
8	4					
9	45					
10	47					
11	35					
12	13					
13	16					
14						
15						
16						
17						
18						

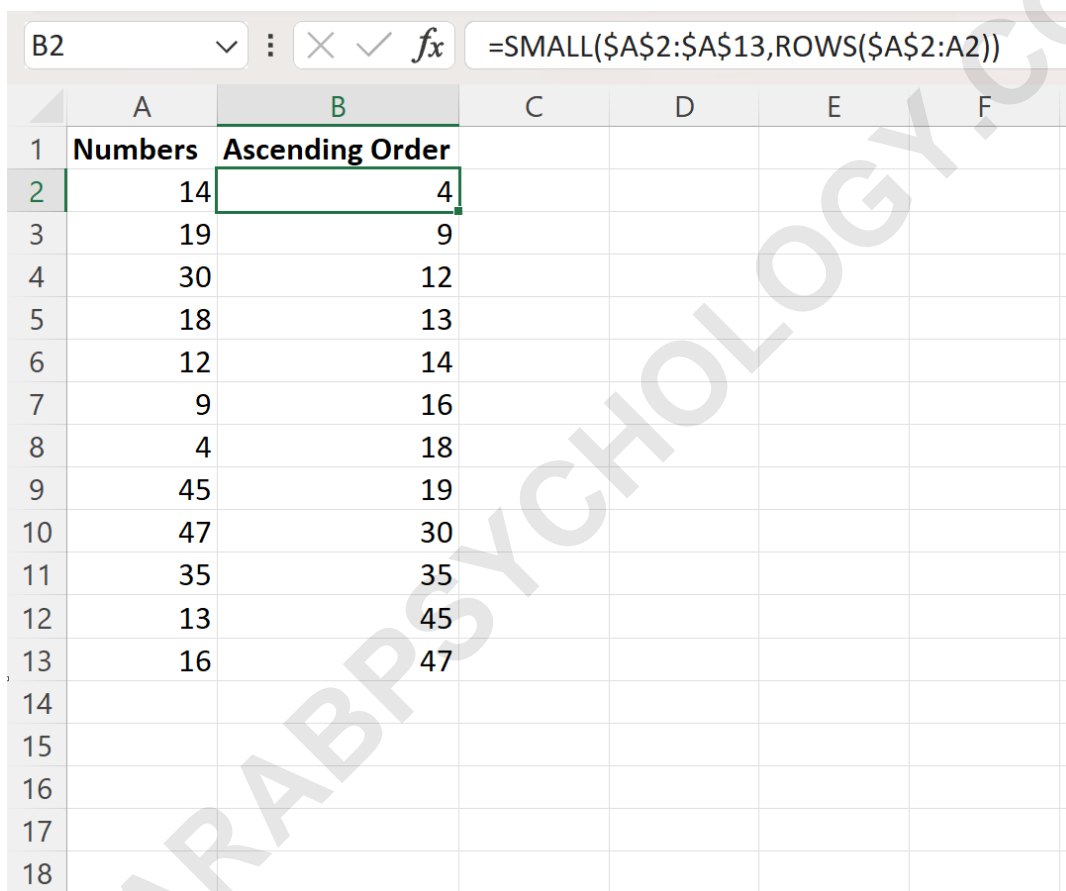
### Step-by-Step Guide: Sorting Numbers in Ascending Order

Executing an **ascending sort** involves a few simple steps. We can type the following **formula** into **cell B2** to return the smallest value from the **range A2:A13**. This initial step establishes the

starting point for our sorted list, identifying the minimum value within the **dataset**:

**=SMALL(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))**

Once the **formula** is entered into the first cell, we can then click and drag this **formula** down to each remaining cell in **column B** to return the numbers from the **range A2:A13** in **ascending order**. As the **formula** is replicated, the **ROWS function** increments, prompting the **SMALL function** to find the second smallest, then the third smallest, and so on, until the entire list is populated.



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	<b>Numbers</b>	<b>Ascending Order</b>				
2	14	4				
3	19	9				
4	30	12				
5	18	13				
6	12	14				
7	9	16				
8	4	18				
9	45	19				
10	47	30				
11	35	35				
12	13	45				
13	16	47				
14						
15						
16						
17						
18						

The formula bar at the top shows the formula: `=SMALL($A$2:$A$13,ROWS($A$2:A2))`

After the process is complete, **Column B** now displays the numbers in **column A** in **ascending order** (smallest to largest). Note that this **formula** uses the **SMALL function** in **Excel** to return the nth smallest value in a **range** and then uses the **ROWS** function to specify the value for n. For example, the first value in **column B** represents the 1st smallest value in **column A**. Then, the second value in **column B** represents the 2nd smallest value in **column A**, ensuring a perfect progression.

## The Logic of the LARGE Function for Descending Sorts

When the objective is to display data from **largest to smallest**, we pivot to the **LARGE function**. Similar to its counterpart, the LARGE function is designed to retrieve a value based on its rank, but it starts from the top of the numerical scale. This is particularly useful for leaderboards, high-priority task lists, or financial reports focusing on top-performing assets. By specifying 'k' as the rank, the **LARGE function** allows for a seamless **descending sort** that mirrors the complexity and accuracy of the **ascending** method.

The integration of the ROWS function remains identical in this context. The **ROWS function** serves as the engine that drives the iteration. By starting the count at 1 and increasing it as you move down the **spreadsheet**, you effectively ask **Excel** to "give me the #1 largest value," then "give me the #2 largest value," and so forth. This systematic approach ensures that the resulting list is a complete and perfectly reversed reflection of the original data **range**.

One of the significant advantages of using the **LARGE function** in a **formula** is that it remains unaffected by the physical location of the data. Whether your numbers are scattered or already partially sorted, the **formula** will scan the entire defined **range** and extract the values based purely on their numerical magnitude. This creates a high level of **data reliability**, which is essential for any professional-grade **spreadsheet** application or **database** management task.

### Step-by-Step Guide: Sorting Numbers in Descending Order

To implement a **descending sort**, we follow a nearly identical procedure to the ascending method, but with the **LARGE function** as our primary tool. We can type the following **formula** into **cell B2** to return the largest value from the **range A2:A13**. This immediately identifies the peak value in your **dataset** and places it at the head of your new list:

```
=LARGE($A$2:$A$13,ROWS($A$2:A2))
```

After establishing the first cell, we can then click and drag this **formula** down to each remaining cell in **column B** to return the numbers from the **range A2:A13** in **descending order**. The visual transformation of the data is immediate, providing an organized view of the numbers from high to low, which is often the preferred format for executive summaries and performance tracking.

	A	B	C	D	E	F
1	<b>Numbers</b>	<b>Descending Order</b>				
2	14	47				
3	19	45				
4	30	35				
5	18	30				
6	12	19				
7	9	18				
8	4	16				
9	45	14				
10	47	13				
11	35	12				
12	13	9				
13	16	4				
14						
15						
16						
17						

Upon completion, **Column B** now displays the numbers in **column A** in **descending order** (largest to smallest). Note that this **formula** uses the **LARGE** function in **Excel** to return the nth largest value in a **range** and then uses the **ROWS** function to specify the value for n. For example, the first value in **column B** represents the 1st largest value in **column A**. Then, the second value in **column B** represents the 2nd largest value in **column A**, and so on, until the list is fully populated.

## Advanced Tips for Professional Data Organization

While the **formulas** provided are powerful, there are several advanced considerations that can further enhance your **Excel** experience. For example, handling empty cells or non-numeric data within your **range** is a common challenge. If your **range** includes blanks, the **SMALL** and **LARGE** functions might return errors. To circumvent this, you can wrap your **formula** in an **IFERROR** function, which allows you to specify a custom message or a blank space if a calculation fails, maintaining the aesthetic **cleanliness** of your **spreadsheet**.

Additionally, for users working with massive **datasets**, it is worth exploring the **SORTBY** function. Unlike the standard **SORT** tool, **SORTBY** allows you to arrange one **range** based on the values in another **range**. This is incredibly useful for lists where you want to sort names alphabetically based

on their corresponding sales figures or dates. Understanding the hierarchy of these **functions** allows you to select the most efficient tool for your specific **data architecture**, saving time and reducing computational overhead.

Finally, always remember the importance of **absolute references** (the dollar signs). Forgetting to lock your **range** is the most common cause of **formula** failure when sorting. If the **range** shifts as you drag the **formula** down, your sorted list will be incomplete and inaccurate. By double-checking your **syntax** and ensuring that your **source data** is properly defined, you can create robust, automated systems that handle your **data organization** needs with precision and ease. The following tutorials explain how to perform other common tasks in **Excel**:

**How to use the VLOOKUP function for data retrieval.**

**Creating dynamic charts with sorted data ranges.**

**Using Pivot Tables for advanced data summarization.**

**Automating tasks with Excel Macros and VBA.**