Ordering and total order

You've got two arithmetic sequences:
First:
11 12 20 6 9 7 14 1 16 13 19 8 2 4 18 10 17 3 5 15

Second:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

It's easier to retrieve and remember information from the second sequence. The main reason for that is **ordering**.

Total order

For natural numbers, the smallest number is *1,* then come *2, 3, 4, etc*. This order has strict rules that we can describe formally.

We say that the sequence has a **total order** with the relation *less or equal* when for each *a*, *b* and *c*
from this sequence the following statements are true:

- if $a \leq b$ and $b \leq a$ then a =b
- If a <= b and b <= c then a <= c
- a <= b or b <= a

You can look at the second sequence above and check if it is possible to apply these rules there.

Apart from numbers, we can put any other types of data in total order, all we need is to define the operation *less or equal*.

Lexicographical order

The way we compare strings differs from how we compare numbers.

The more digits a positive integer has, the greater it is, but that's not true for strings.

Their order is similar to the alphabetic one for the same case words.

The particular trait of the alphabetic order appears when we are comparing lowercase and uppercase letters.

For example, in ASCII encoding, all Latin-script uppercase letters are less than

any lowercase letter, which means that Z < a.

Moreover, we can compare any symbols with letters because all of them have the mapping to integer values, and we know how to compare numbers.

To compare two strings for the *less or equal* relation, do it step by step:

1. Compare each word letter by letter from the first position. When two letters in the same position differ, the order of words matches the order of these letters: cord < core.
2. When there are no letters left in one of the words, that word is always *less than or equal* to the other one, so ball < ballet.

Note that the empty string is a prefix for any other string, so it's the smallest string due to the lexicographical order.

Chronological order

Can you tell what day was earlier: *1969-06-20* or *1965-03-18*? The order of dates is straightforward as we compare them by year, then by month, and finally by day. When we meet the first difference, we can identify which date is less.

Almost nothing changes when you add time to date. You compare two timestamps step by step:

- by year
- by month
- by day
- by hour
- by minute
- by second

Just like in real life, the future dates are greater than the current time, and the past is less.

Application of ordering

Let's recall the sequence from the beginning of the topic:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
The ordering has several applications based on its useful properties, so, looking at this sequence, we can:

- compress the information (say, we have elements from 1 to 20)
- look for elements faster (it's easy to find elements in it and answer what elements are not present)
- analyze the structure of the data (by the glimpse, we can find the minimum and maximum elements in the sequence)

You can find applications for ordering in compression algorithms, search

engines, databases, analytics tools, and many other programs.