

--- Day 1: Historian Hysteria ---

The Chief Historian is always present for the big Christmas sleigh launch, but nobody has seen him in months! Last anyone heard, he was visiting locations that are historically significant to the North Pole; a group of Senior Historians has asked you to accompany them as they check the places they think he was most likely to visit.

As each location is checked, they will mark it on their list with a **star**. They figure the Chief Historian **must** be in one of the first fifty places they'll look, so in order to save Christmas, you need to help them get **fifty stars** on their list before Santa takes off on December 25th.

Collect stars by solving puzzles. Two puzzles will be made available on each day in the Advent calendar; the second puzzle is unlocked when you complete the first. Each puzzle grants **one star**. Good luck!

You haven't even left yet and the group of Elvish Senior Historians has already hit a problem: their list of locations to check is currently **empty**. Eventually, someone decides that the best place to check first would be the Chief Historian's office.

Upon pouring into the office, everyone confirms that the Chief Historian is indeed nowhere to be found. Instead, the Elves discover an assortment of notes and lists of historically significant locations! This seems to be the planning the Chief Historian was doing before he left. Perhaps these notes can be used to determine which locations to search?

Throughout the Chief's office, the historically significant locations are listed not by name but by a unique number called the **location ID**. To make sure they don't miss anything, The Historians split into two groups, each searching the office and trying to create their own complete list of location IDs.

There's just one problem: by holding the two lists up **side by side** (your puzzle input), it quickly becomes clear that the lists aren't very similar. Maybe you can help The Historians reconcile their lists?

For example:

3	4
4	3
2	5
1	3
3	9
3	3

Maybe the lists are only off by a small amount! To find out, pair up the numbers and measure how far apart they are. Pair up the **smallest number in the left list** with the **smallest number in the right list**, then the **second-smallest left number** with the **second-smallest right number**, and so on.

Within each pair, figure out **how far apart** the two numbers are; you'll need to **add up all of those distances**. For example, if you pair up a **3** from the left list with a **7** from the right list, the distance apart is **4**; if you pair up a **9** with a **3**, the distance apart is **6**.

In the example list above, the pairs and distances would be as follows:

- The smallest number in the left list is **1**, and the smallest number in the right list is **3**. The distance between them is **2**.

Our **sponsors** help make Advent of Code possible:

Cloudsmith - Code shapes reality; now solve your next puzzle: global artifact management. Join our journey to shape, secure, and be the world's software supply chain. We overcame the impossible and became the improbable - now, it's inevitable.

- The second-smallest number in the left list is `2`, and the second-smallest number in the right list is another `3`. The distance between them is `1`.
- The third-smallest number in both lists is `3`, so the distance between them is `0`.
- The next numbers to pair up are `3` and `4`, a distance of `1`.
- The fifth-smallest numbers in each list are `3` and `5`, a distance of `2`.
- Finally, the largest number in the left list is `4`, while the largest number in the right list is `9`; these are a distance `5` apart.

To find the **total distance** between the left list and the right list, add up the distances between all of the pairs you found. In the example above, this is `2 + 1 + 0 + 1 + 2 + 5`, a total distance of `11`!

Your actual left and right lists contain many location IDs. What is the total distance between your lists?

To begin, [get your puzzle input](#).

Answer: [\[Submit\]](#)

You can also [\[Share\]](#) this puzzle.