Problem 13 – Taking a Closer Look

After you are given the number of testcases that you will be entering, the next set of input are the two integers that you will be testing.

Your job is to figure out the smallest number of steps to take to get from the first number (X) to the second number (Y). Your requirements from the program are:

1. The length of the first step you take AND the last step you take must be 1.
2. The length of each step can be:
   1. One step larger than the previous step
   2. Equal to the previous step
   3. One step smaller than the previous step

How would you attack this?

Let’s walk through an example to help you think about everything you need.

Let’s say you enter the following integers:

**40 50**

First, let’s look at the two integers on a line:

Shape

Description automatically generated

If you look at the integers on this line, what you can tell is that there is a midpoint to these numbers. On the left side of the midpoint, you have the possibility to take larger steps. On the right side of the midpoint, you have the possibility to take smaller steps.

On the line above, the midpoint is (Y + X )/2 = (50 + 40 )/2 = 90/2 = 45.

Chart, box and whisker chart

Description automatically generated

Now, we know that the first step we need to take must be a length of 1, so that leaves us with the following:

1. You have ((midpoint – x) – 1) = ((45 – 40) -1) = **4** steps to take to reach the midpoint, AND
2. You have ((Y – X) – 1) = ((50 – 40) – 1) = (10 – 1) = **9** steps to reach the second integer

Chart, box and whisker chart

Description automatically generated

**Question**: Should you increase the size of your step??

**Answer**: Yes!

Based on requirement (2a) above, you can only increase the size of your step by 1. So, you add 1, which increases the size of your step to 2.

So far, you have taken 2 steps, and you must review two things:

1. You have ((midpoint – x) – 3) = ((45 – 40) – 3) = **2** steps to reach the midpoint, AND
2. You have ((Y – X) – 3) = (10 – 3) = **7** steps to reach the second integer

**Question**: Should you increase the size of your step??

**Answer**: Based on (B), yes – it is possible to increase the size again.

Again, based on requirement (2a) at the top, you can only increase the size of your step by 1, so add 1 again, increasing the size of the step to 3.

Chart

Description automatically generated

So far, you have taken 3 total steps, and you have two things to review:

1. You have ((midpoint – X) – 6) = ((45 – 40) – 6) = **-1** steps to reach the midpoint, AND
2. You have ((Y – X) – 6) = (10 – 6) = **4** steps to reach the second integer

**Question**: Should we increase the size of our step??

**Answer**: **No.** You have now passed the midpoint, so it is time to start decreasing the size of the steps being taken. Also, because of requirement (2c) above, you can only decrease the size of the step by 1, so subtract 1 from the size, which results in a step size of 2.

Chart

Description automatically generated

So far, you have taken 4 steps, and you only have one thing to review:

1. You have ((Y – X) – 8) = (10 – 8) = **2** steps to reach the second integer

At this point, you need to ask a new question:

**Question**: Should we decrease the size of our step, OR stay the same?

**Answer**: Because of requirement (1), we cannot keep the same step size, since we only have *two steps to go* before reaching the second integer, and the last step MUST have a size of 1. So, you subtract 1, giving a step size of 1.

Chart

Description automatically generated

So far, you have taken 5 steps, and you have one thing to review:

1. We have ((Y – X) – 9) = (10 – 9) = **1** step to reach the second integer

Based on requirement (1), you are on track because your last step ***must be a size of 1***.

Your output is the minimum number of steps you took, which in this example is 6.

*Not giving you all the code*, please consider:

* What values were you keeping track?
* What items did you have to review?
* What questions did you have to ask at each part of the process?
* How many of these do you have to do (based on the very first input)?

Hopefully this was helpful….