Bisection search behind the scenes with diagrams!

Bisection search is an amazing algorithm (I know loved it when I first saw it in the course), it can find a value much faster than a simple approach. In these diagrams you can experience how this algorithm works.

I've used the same code from the lecture, but in this case, ***we find the square root of 9 with an epsilon 0.08***, and I've added some print statement to help us see what's going on for each step. With a low starting point from 0.0 and a high point of 9, we start our algorithm.... This is what we see on the shell:

- First, the algorithm asks if the answer found (the middle point of the search space) squared is close enough to the answer we are looking for (if the answer squared is within epsilon of the value for which we are calculating the square root) **below you will find a visual explanation of this process...**

```
Are we close enough? False
low = 0.0 high = 9 ans = 4.5
The difference is still  11.25  and it has to be less than  0.08


Are we close enough? False
low = 0.0 high = 4.5 ans = 2.25
The difference is still  3.9375  and it has to be less than  0.08


Are we close enough? False
low = 2.25 high = 4.5 ans = 3.375
The difference is still  2.390625  and it has to be less than  0.08


Are we close enough? False
low = 2.25 high = 3.375 ans = 2.8125
The difference is still  1.08984375  and it has to be less than  0.08


Are we close enough? False
low = 2.8125 high = 3.375 ans = 3.09375
The difference is still  0.5712890625  and it has to be less than  0.08


Are we close enough? False
low = 2.8125 high = 3.09375 ans = 2.953125
The difference is still  0.279052734375  and it has to be less than  0.08


Are we close enough? False
low = 2.953125 high = 3.09375 ans = 3.0234375
The difference is still  0.14117431640625  and it has to be less than  0.08


Yes! The difference is now  0.0701751708984375  which is less than 0.08
numGuesses: 7
2.98828125 is close to square root of 9
```

- We start with a search space from 0.0 up to 9. We know that the square root of 9 is 3, so we have an idea of where the algorithm will find the answer (this will help us through the explanation).
- On each step, the algorithm determines if the answer is close enough to the answer we are looking for, within a margin of error (`epsilon`).

But what exactly is `epsilon`?...

If we square the answer (the middle point of the search space) we get a number. Although this number may be either larger or smaller than our initial number 9, it may still be the answer we are looking for if that difference is smaller than `epsilon`. (For example, if we say epsilon is 0.5, and we want the square root of 9, **if our answer squared gives us 8.6 or 9.4, those are "acceptable" answer because they are within that margin of error we've set**, that is the meaning of epsilon, but the algorithm will select the first answer it finds within epsilon)

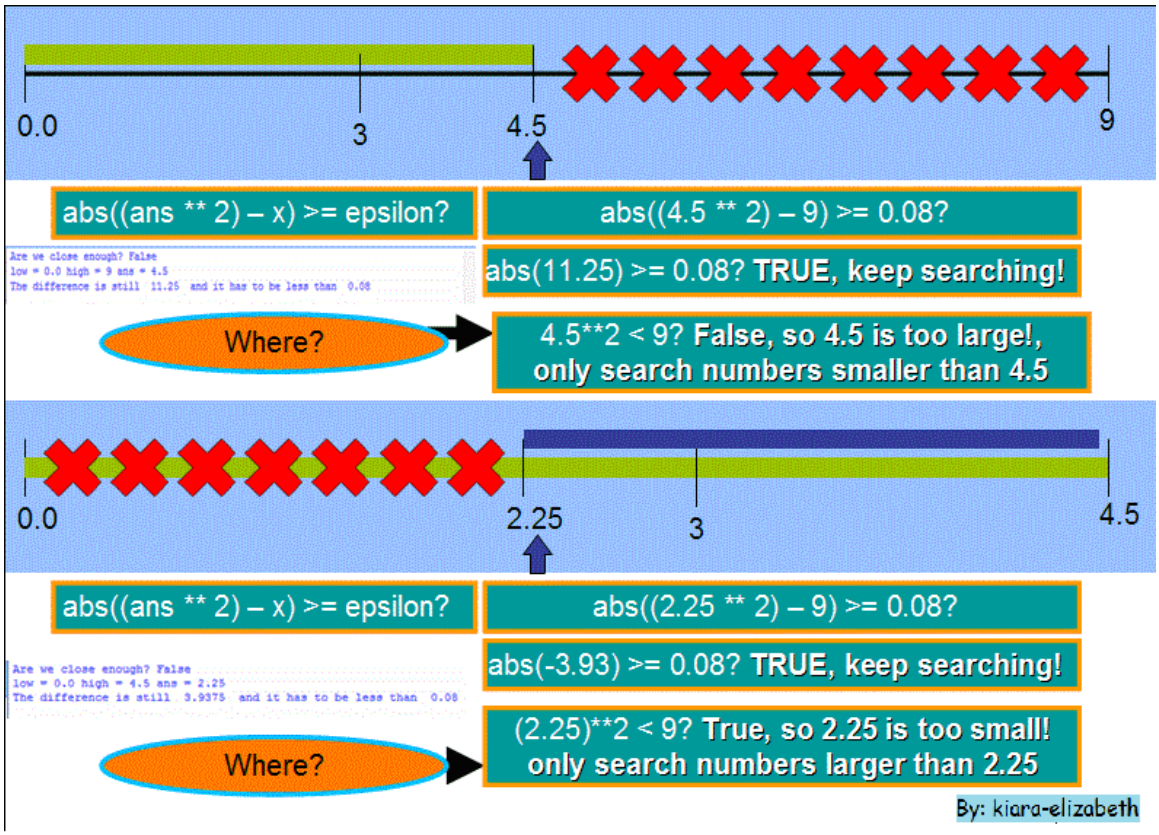
If it's not within epsilon, it means it's not precise enough, and we have to reduce our search space even further.

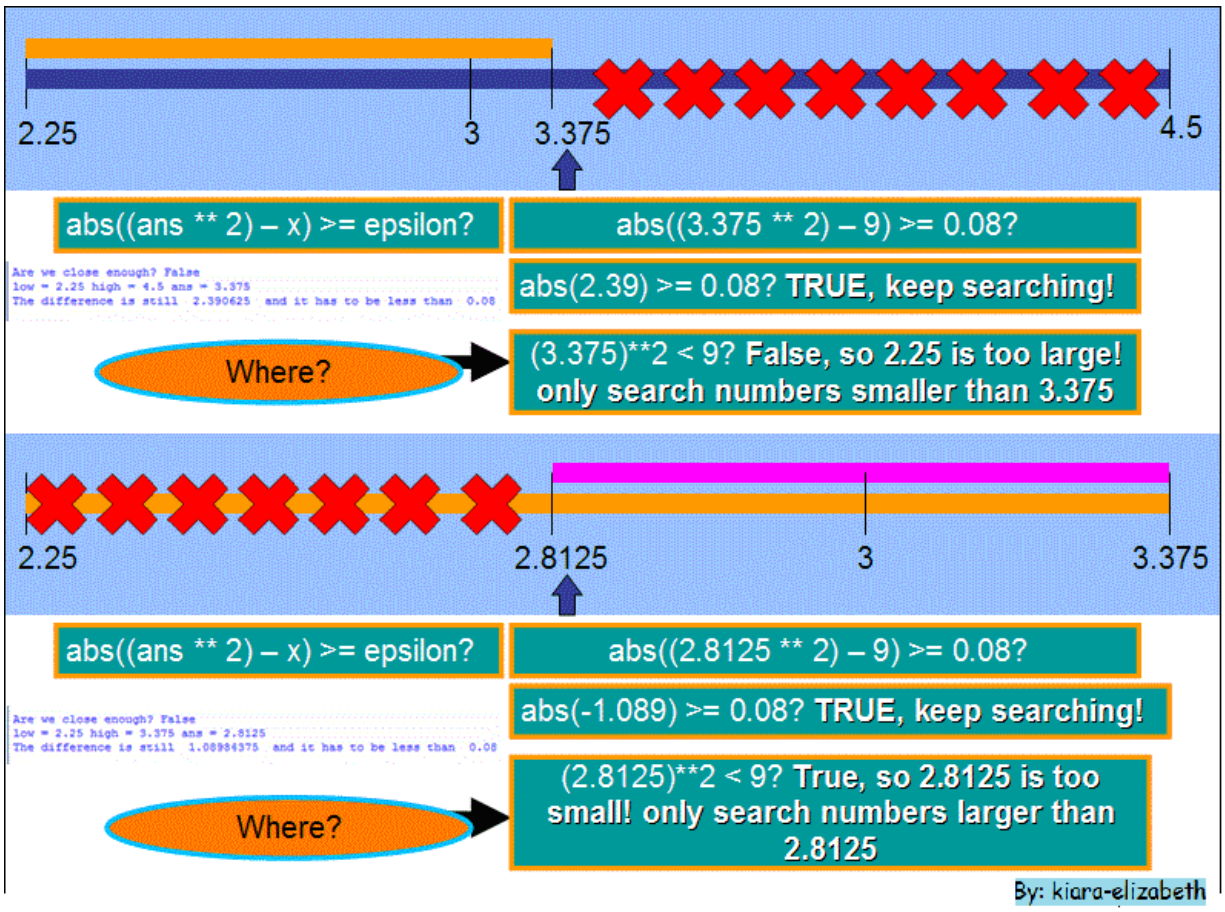
But how do we determine where to keep looking? **We can reduce our search space BY HALF!**

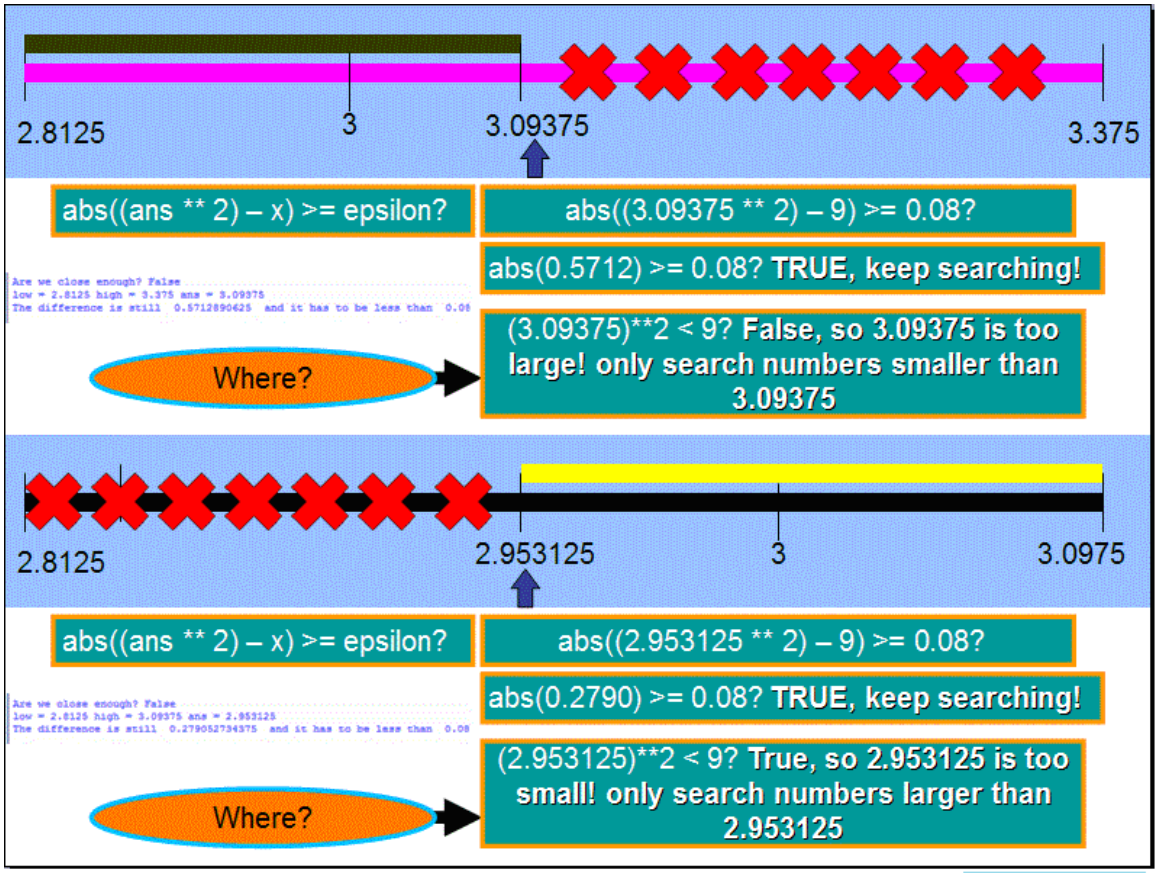
- We can reduce our search space BY HALF! on each step by noting if the answer squared is too small or too large to be the right answer.
    - If our answer squared is **too small**, the algorithm will *discard the half of the search space that contains numbers smaller than our answer*, and our new search space will be the other half that contains numbers larger than our previous answer. Our new Low will be the previous answer (the middle point of the previous search space) and our High point will remain the same.
    - On the contrary, if our answer squared is **too large** to be the right answer, the algorithm will *discard the HALF of the search space that contains numbers Larger than our previous answer*. Our new High point will be the previous answer (the middle point of the previous search space) and our Low point will remain the same.

These steps will repeat until the algorithm finds an answer that, if squared, is close enough to the initial number (9), and so we will find its approximate square root.

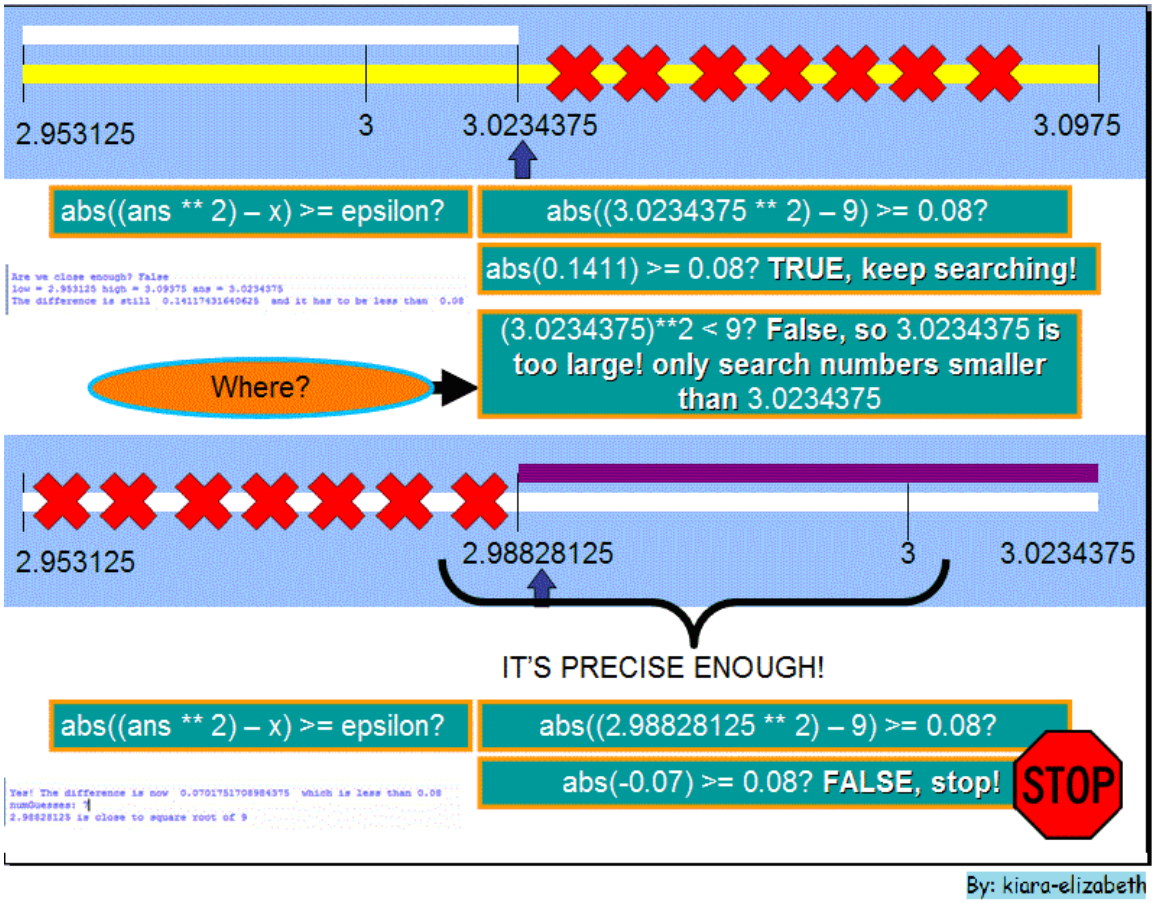
With these diagrams you can follow the shell step by step, finding the square root of 9, within epsilon 0.08:




2.25
3
3.375
4.5
abs((ans ** 2) – x) >= epsilon?
abs((3.375 ** 2) – 9) >= 0.08?
Are we close enough? False
low = 2.25 high = 4.5 ans = 3.375
The difference is still 2.390625 and it has to be less than 0.08
abs(2.39) >= 0.08? TRUE, keep searching!
Where?
(3.375)**2 < 9? False, so 2.25 is too large! only search numbers smaller than 3.375
2.25
2.8125
3
3.375
abs((ans ** 2) – x) >= epsilon?
abs((2.8125 ** 2) – 9) >= 0.08?
Are we close enough? False
low = 2.25 high = 3.375 ans = 2.8125
The difference is still 1.08984375 and it has to be less than 0.08
abs(-1.089) >= 0.08? TRUE, keep searching!
Where?
(2.8125)**2 < 9? True, so 2.8125 is too small! only search numbers larger than 2.8125
By: kiara-elizabeth



By: kiara-elizabeth



Hope it helps! If you have any questions please post them on the forums! :)

Estefania (kiara-elizabeth).