You want to approximate the value of a function, in this case the square root of a positive integer, *x*.

You want to approximate the value of a function, in this case the square root of a positive integer, *x*.

In order to accomplish this goal, you decide to use bisection search.

You want to approximate the value of a function, in this case the square root of a positive integer, *x*.

In order to accomplish this goal, you decide to use bisection search.

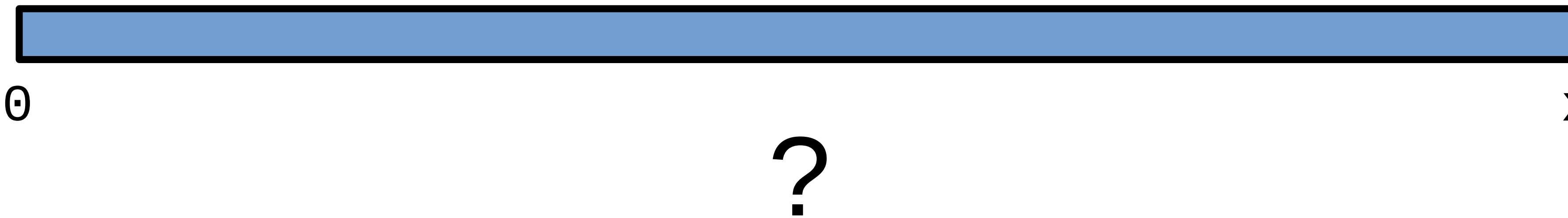
The bisection search algorithm says place your guess at the halfway point of the interval and evaluate it.

You want to approximate the value of a function, in this case the square root of a positive integer, *x*.

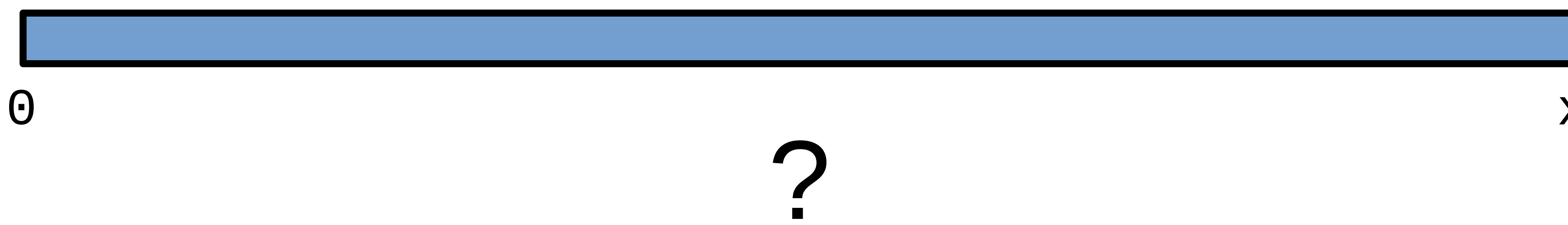
In order to accomplish this goal, you decide to use bisection search.

The bisection search algorithm says place your guess at the halfway point of the interval and evaluate it.

How do we get started?

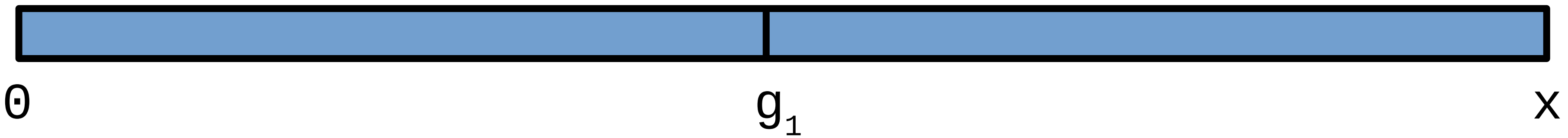


Since we're looking at square root of a positive integer, we know that the value of the square root of *x* must be between zero and *x*.

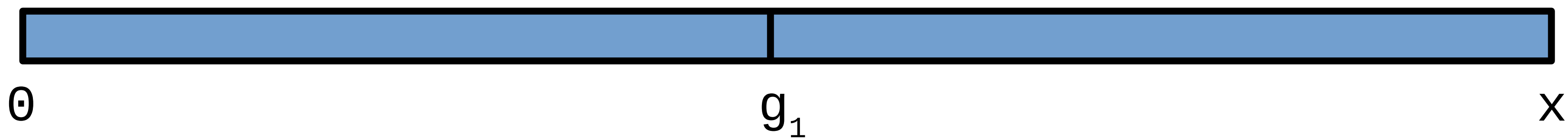


Since we're looking at square root of a positive integer, we know that the value of the square root of *x* must be between zero and *x*.

So we use the bisection search algorithm to come up with our first guess, g_1, by dividing x by 2.

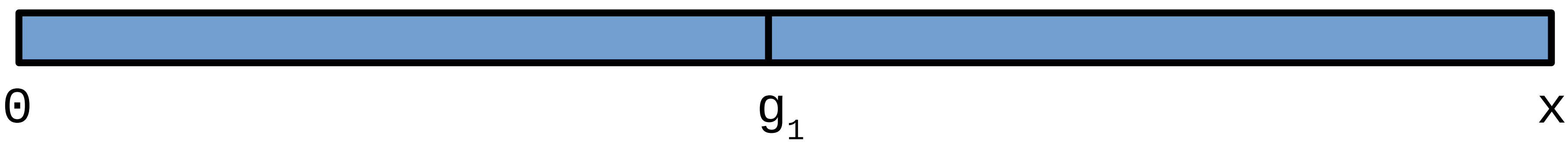


Like this ...



Like this ...

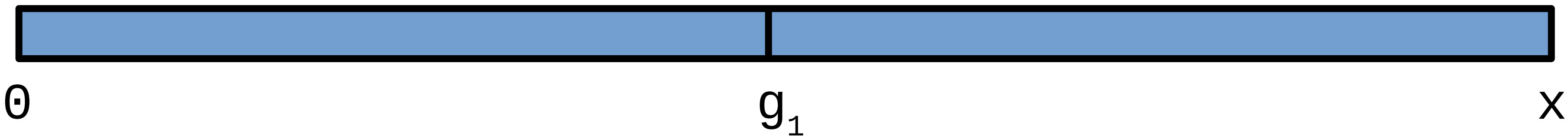
Now we have to evaluate how good our current guess, g_1, is.



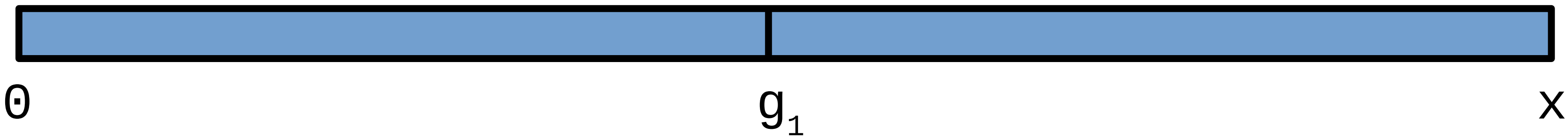
Like this ...

Now we have to evaluate how good our current guess, g_1, is.

if g_1 is our guess of the square root,
then the square root squared (g_1^2) is what
we should check against our initial value, x.

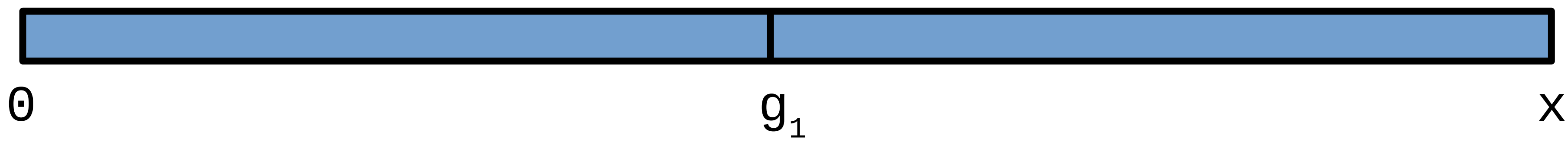


There are 2 options for g_1^2 relative to x:



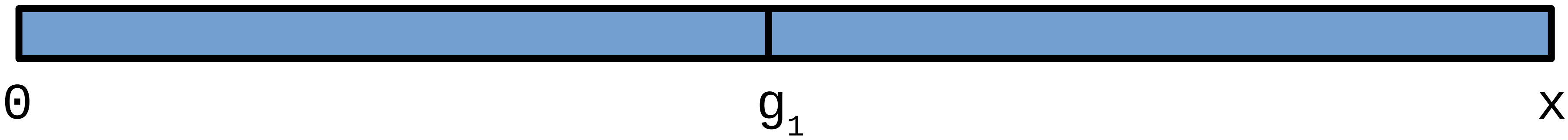
There are 2 options for g_1^2 relative to x:

Why check g_1^2? Remember, if g_1 is our guess of the square root, then the square root squared (g_1^2) is what we should check against our initial value, x.



There are 2 options for g_1^2 relative to x:

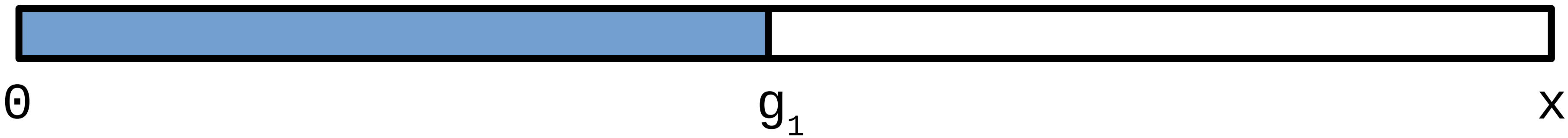
1. $g_1^2 > x$
2. $g_1^2 < x$



There are 2 options for g_1^2 relative to x:

1. $g_1^2 > x$
2. $g_1^2 < x$

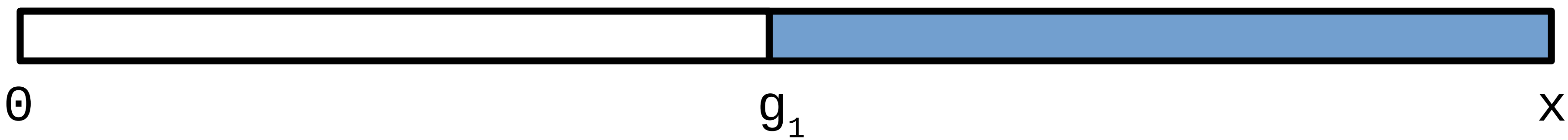
OK, there are 3 options, g_1^2 could == x,
but we don't need to worry about that right now.



There are 2 options for g^2 relative to x:

1. $g_1^2 > x$

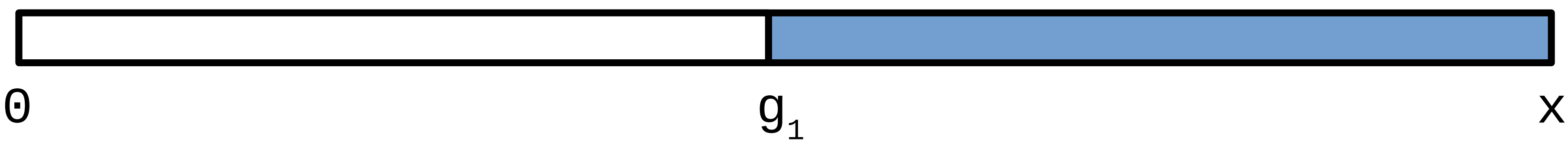
   If $g_1^2 > x$ then g_1 is larger than the square root of x, so we know the square root is in the interval between 0 and g_1 (shown in blue)



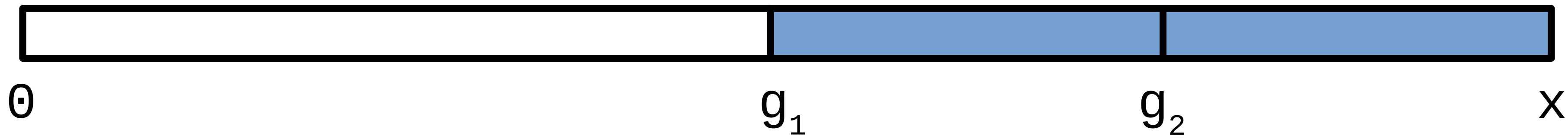
There are 2 options for g_1^2 relative to x:

2. $g_1^2 < x$

   If $g_1^2 < x$ then g_1 is smaller than the square root of x, so we know the square root is in the interval between g_1 and x (shown in blue)

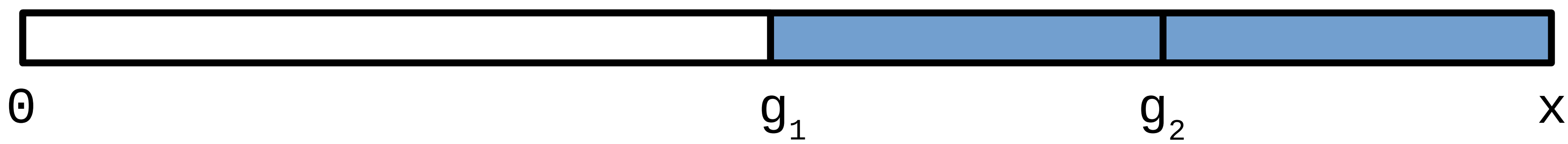


This “check” tells us which half of the bisected interval is closer.
(shown in blue)



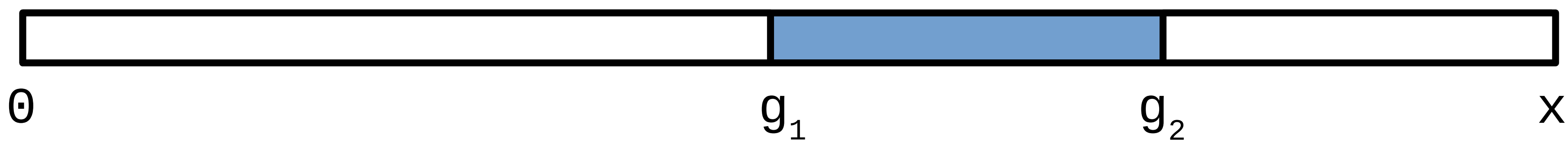
This “check” tells us which half of the bisected interval is closer.

Now we “bisect” again, making a new guess, g_2, and check again.
(In this case we have assumed that $g_1^2 < x$)



And repeat the evaluation process:
There are 2 options for g_2^2 relative to x:

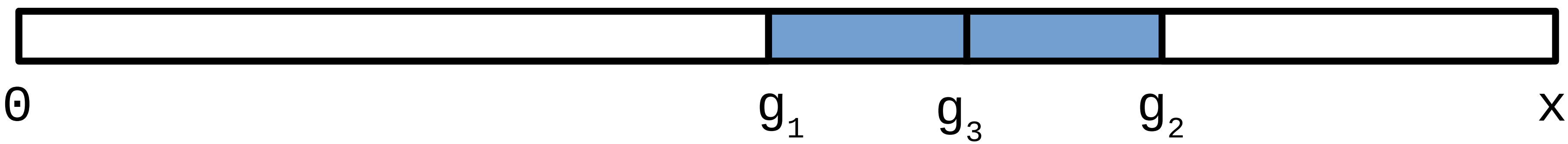
1. $g_2^2 > x$
2. $g_2^2 < x$



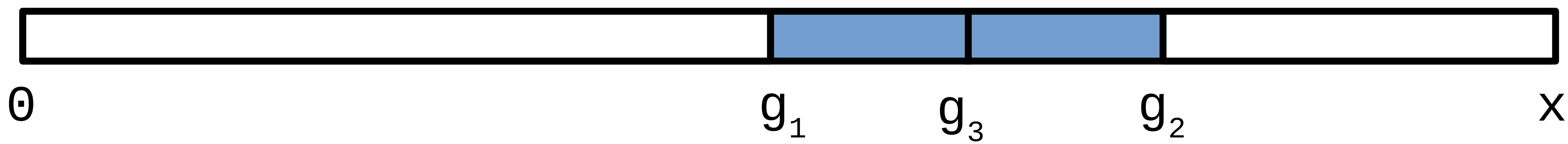
And repeat the evaluation process:
Assuming that:

$$g_2^2 > x$$

We “bisect” again...



And make a new guess, g_3, based on our analysis



And repeat the evaluation process:
There are 2 options for g_3^2 relative to x:

1. $g_3^2 > x$
2. $g_3^2 < x$

etc. , etc., etc.

Each time, we have a smaller range of values, and over time, the approximation g_n gets closer to the real value of the square root of *x*.

an example

You want to approximate the value of the square root of 2.

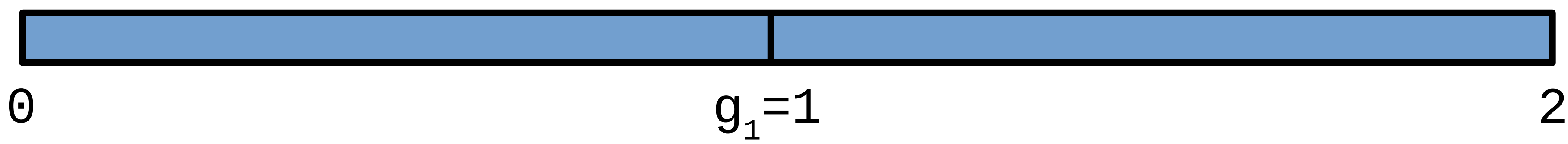
```
FYI:
math.sqrt(2) = 1.4142135623730951
```

Since we're looking at square root of a positive integer, we know that the value of the square root of *x* must be between zero and *x*.

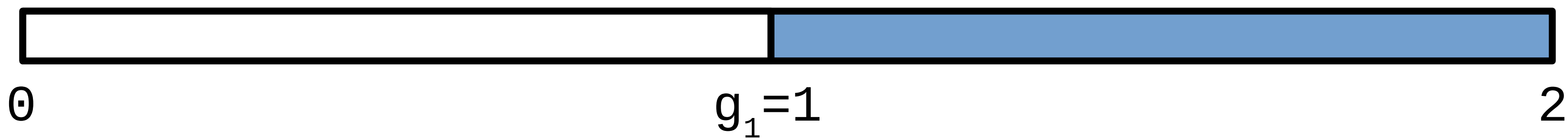
Since we're looking at square root of a positive integer, we know that the value of the square root of 2 must be between zero and 2.

Since we're looking at square root of a positive integer, we know that the value of the square root of 2 must be between zero and 2.

So we use the bisection search algorithm to come up with our first guess, g_1, by dividing 2 by 2.



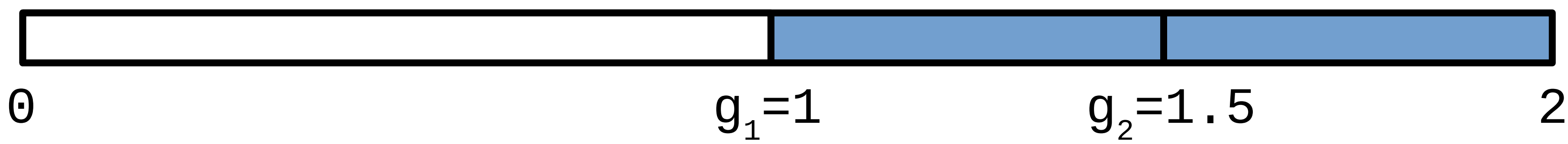
$$g_1 = 2/2 = 1$$



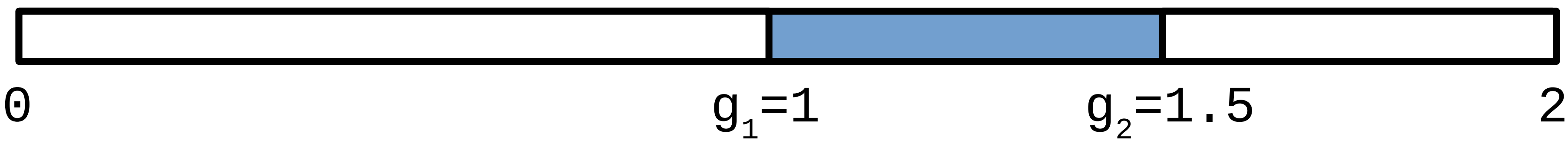
$g_1^2 = 1^2 = 1$ so relative to x:

- $g_1^2 < 2$
  - Since $g_1^2 < 2$, g_1 is smaller than the square root of 2, so we know the square root is in the interval between g_1 and x (shown in blue)

```
FYI:
math.sqrt(2) = 1.4142135623730951
```



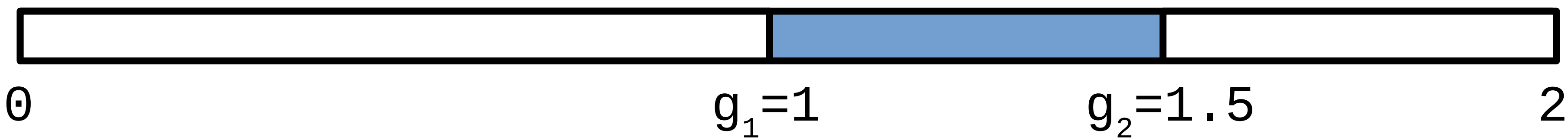
Now we “bisect” again, making a new guess, g_2 = 1.5, and check again.



And repeat the evaluation process:

since:

$$g_2^2 = 1.5^2 = 2.25$$

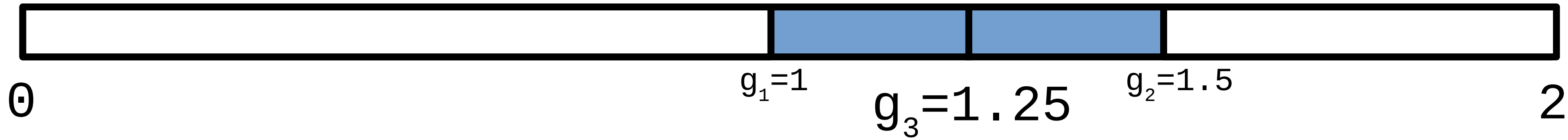


And repeat the evaluation process:

since:

$$g_2^2 = 1.5^2 = 2.25 > x$$

```
FYI:
math.sqrt(2) = 1.4142135623730951
```

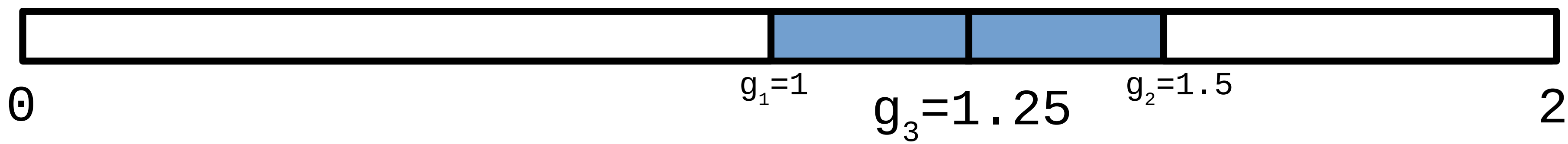


We “bisect” again,

And make a new guess, g_3, between g_1 and g_2

So, $g_3 = g_1 + g_2 = 1.25$

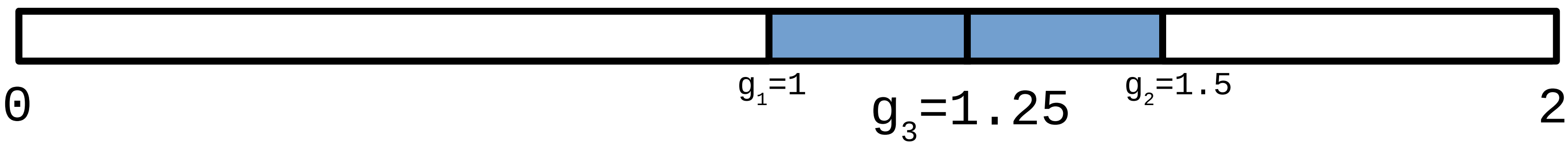
FYI:
`math.sqrt(2) = 1.4142135623730951`



We “bisect” again,
And make a new guess, g_3, between g_1 and g_2
So, $g_3 = g_1 + g_2 = 1.25$

etc. , etc., etc.

Each time, we have a smaller range of values, and over time, the approximation g_n gets closer to the real value of the square root of *2*.



We “bisect” again,
And make a new guess, g_3, between g_1 and g_2
So, $g_3 = g_1 + g_2 = 1.25$

etc. , etc., etc.

Each time, we have a smaller range of values, and over time, the approximation g_n gets closer to the real value of the square root of *2*.

After 50 iterations the approximation is `1.414213562373095`

FYI:
`math.sqrt(2) = 1.4142135623730951`