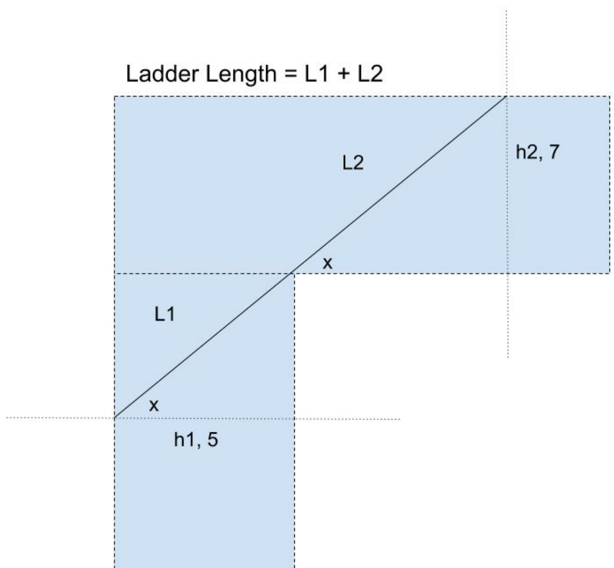
In[ ]:=

Arnav Dani – Mathematica Project 1 – P5

Original Problem

using similar triangles and a common angle to find L.
Since the bottom of the vertical alleyway and the horizontal side of the horizontal alleyway are parallel, the angle made by a horizontal line at the bottom is parallel to the angle between the ladder and the upper, horizontal alley. Using the fact that they are the same
angle, trig ratios can be used to calculate l.

x = angle shown in picture




```
(*equation*)
p[x_] = 5/Cos[x] + 7/Sin[x]
```

$$p[x\_] = \frac{5}{\text{Cos}[x]} + \frac{7}{\text{Sin}[x]}$$

```
(*finding derivative*)

f[x] = p'[x]
```

Since the ladder that touches all 3 points is the maximum ladder that can fit, the minimum value of the function that touches all 3 points - the 2 walls and the corner. Therefore I am looking for a minimum not a maximum

Finding critical values by setting derivative to 0
since it is a function of an angle, limits on the x must be placed to not get infinite solutions
the angle must obviously be greater than 0 and must be less than π/2 because that is the max angle that can be placed at a corner with a right angle
I also call sols[[1]] throughout to ensure that I call the first solution since Solve outputs an array

In[ ]:=

```
sols = Solve[ f[x] == 0 && x ≥ 0 && x ≤ π/2]
(*prints x value of minimum*)
N[sols[[1]]]
```

Once the critical value is found, plug that back into the original equation
to find the y value at the minimum

In[ ]:=

```
ans = p[x] /. sols[[1]] // FullSimplify
(* converts answer into number instead of radical expression *)
N[ans]
```

Modification — Variable Alley Width

Takes in the input from the user for the variable alley widths - I took in input at the beginning because it would not work successfully if I took the input later

In[ ]:=

```
h1 = Input["Please input a value for h1 that is greater than 0"]
h2 = Input["Input value for h2 that is greater than 0"]
```

defines the function using the same ‘similar triangle’ method as Part 1
since the angle is a right angle, the lines are parallel, so there is a common angle
so the trig ratios can be used to calculate and sum the respective hypotenuses
Since the height is supposed to be variable, the inputted h1 and h2 are used to define the function
‘x’ is the angle between a horizontal line at the bottom of the ladder and the ladder itself as shown in the diagrams

In[ ]:=
```
p[x_, h1_, h2_] = h1/Cos[x] + h2/Sin[x]


(*find derivative *)
f[x_, h1_, h2_] = D[p[x, h1, h2], x]
```

Finds critical values where the derivative is zero based on the parameters passed in through the input since I am solving for an angle at a wall, the answer must be between 0 and 90 which is why I limited the range
of possible values of x

Similar to above - the graph/derivative will show all the ladder lengths that go beyond the bounds of the alley corner
so I am looking for the minimum in the function . Since there is only 1 critical value between 0 and $\pi/2$, that must be the
minimum and the value of the max length that will fit. Since the Solve method returns an array, I have to always call the
first index of that array which is why I use sols[[1]]

```
ans = p[x /. sols[[1]], h1, h2] // FullSimplify

(*converts the answer into a number output instead of a radical expression *)
N[ans]
```

In[ ]:=

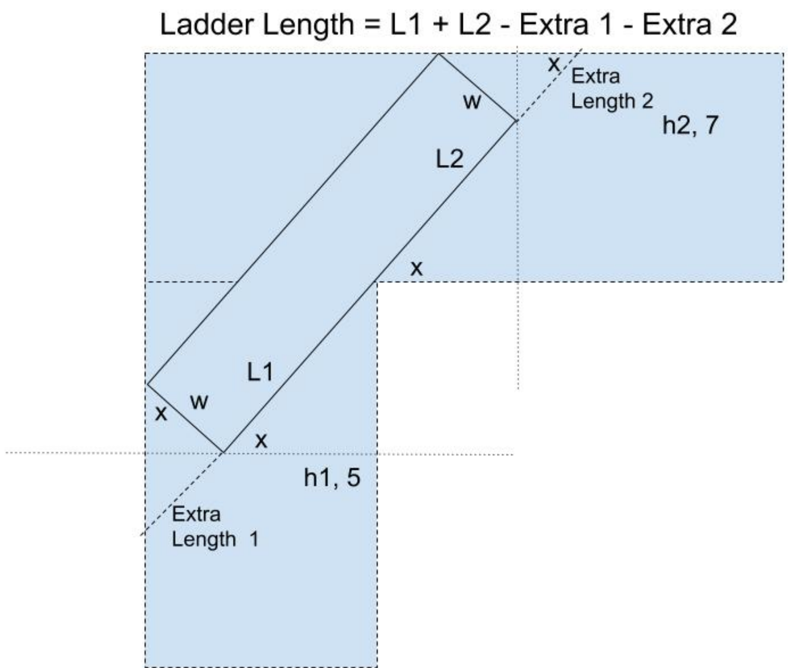
Modification – Variable Ladder width

Takes in the input from the user for the variable alley widths - I took in input at the beginning because it would not work successfully if I took the input later

In[ ]:=
```
w = Input["Please input a ladder width between 1 and 3 (inclusive)"]
```

Because the ladder is wider, the max width is reduced because the closer one gets to the corner, the less room there is
to solve this, I said that the hypotenuse “l” of the original ladder is equal to the hypotenuse of the ladder minus the
parts that are cut out on each end due to the width of the ladder. By using parallel lines and subtraction, I found
the angles for all the triangles and then that the length left over on the bottom was w*tan(x) and the length left
over at the top was w/tan(x). Since I knew how to calculate the total length, I used those values and modified the
initial formula to subtract the excess length.

x is the angle shown in the picture



In[ ]:=
```
p[x_, w_] = 5/Cos[x] + 7/Sin[x] - w * Tan[x] - w/Tan[x]

(*finds derivative *)
f[x_, w_] = D[p[x, w], x]
```

$$p[x, w] = \frac{5}{\mathrm{Cos}[x]} + \frac{7}{\mathrm{Sin}[x]} - w * \mathrm{Tan}[x] - \frac{w}{\mathrm{Tan}[x]}$$

Finds critical values where the derivative is zero based on the parameter passed in through the input since I am solving for an angle at a wall, the answer must be between 0 and 90 degrees which is why I limited the range
of possible values of x. I am looking for the minimum of the graph because it is the only length that doesn't go
beyond the restrictions.

```
sols =
     Solve[
      f[x, w] == 0 && x ≥ 0 && x ≤ π/2]

(*solves and prints final value*)
ans = p[x /. sols[[1]], w] // FullSimplify
N[ans]
```