Source 1
I found this arduino solution with three Ultrasonic sensors:

Build Idea
I will use this build as a starting point:

I have decoded the arduino code from source 1 into the following algorithm:

Loop Forever
  // get readings
  distFront = front ultrasonic distance
  distLeft = left ultrasonic distance
  distRight = right ultrasonic distance

  // clear in front, move forward
  if distFront > 20
    forward 1 sec
/ if wall is between 10 and 20 on the left, move forward
  if distLeft > 10 and distLeft < 20
    forward 1 sec
// if move away from the wall on the left, steer gradually back
  if distLeft >= 20
    spin left a bit 30ms
    forward 1 sec
// if too close to wall on left, drive right a bit
  if distLeft < 10 and distLeft > 0
    spin right a bit 30ms
    forward 1 sec
// wall in front but clear to the right
  if distFront <= 10 and distRight > 20
    stop 300ms
    spin right 90 degrees
// both front and right have a wall spin right 180 degrees
  if distFront <= 20 and distRight < 20
    stop 1 sec
    spin right 180 degrees

---

**test 1** - forward and stop at wall in front
- forward
- stop

**test 2**
- forward and stop at wall in front
- stick to left hand wall
- turn left if clear
This works OK, but is a bit jerky. A PID control would be good.
Test 4 -
- forward and stop at wall in front
- stick to left hand wall
- turn left if clear
- wall in front but clear to the right
- dead end - both front and right have a wall spin right 180 degrees
All went well when there were dead ends to the left, as this is a left hand rule algorithm. However, dead-ends were skipped when running the maze the other way and dead-ends were on the right. This actually would not be a problem if the maze is closed and the entry and exit is the same.
For mazes where the entry and exit are in different places, the solution would be to have a left or right program loaded up and make sure that you are traversing the maze in the correct direction.

Of all the plans, so far, this one has been the most effective.

Source 2

Here is another idea. I haven't coded or tested this out. Give it a go!

1st refinement

Set dis to 25
Loop forever
  Check left, front, right distances
  If wall on all 3 sides, dead end
    Turn right until 180 degrees the other way
  If wall on right and front sides
    Turn left
  If wall on left and front sides
    Turn right
  If wall on front side
    Turn right
  If wall on left side
    Turn right
  If wall on right side
    Turn left
  Go forward
  Else
    Go forward
2nd Refinement

\[ \text{dis} = 25 \]
Loop forever

Check left, front, right distances

If \( \text{frontSensor} < \text{dis} \) AND \( \text{rightSensor} < \text{dis} \) AND \( \text{leftSensor} < \text{dis} \)

\hspace{1cm} // \text{wall in front of all 3 sides, dead end}

\hspace{1cm} \text{spin right 180 degrees}

Else if \( \text{frontSensor} < \text{dis} \) AND \( \text{rightSensor} < \text{dis} \) AND \( \text{leftSensor} > \text{dis} \)

\hspace{1cm} // \text{wall on right and front sides}

\hspace{1cm} \text{spin right 90 degrees}

Else if \( \text{frontSensor} < \text{dis} \) AND \( \text{rightSensor} > \text{dis} \) AND \( \text{leftSensor} < \text{dis} \)

\hspace{1cm} // \text{wall on left and front sides}

\hspace{1cm} \text{spin right 90 degrees}

Else if \( \text{frontSensor} < \text{dis} \) AND \( \text{rightSensor} > \text{dis} \) AND \( \text{leftSensor} > \text{dis} \)

\hspace{1cm} // \text{wall on front side}

\hspace{1cm} \text{spin right 90 degrees}

Else if \( \text{frontSensor} < \text{dis} \) AND \( \text{rightSensor} > \text{dis} \) AND \( \text{leftSensor} < \text{dis} \)

\hspace{1cm} // \text{wall on left side; follow wall and correct}

\hspace{1cm} \text{spin right 180 ms}

\hspace{1cm} \text{forward}

Else if \( \text{frontSensor} > \text{dis} \) AND \( \text{rightSensor} > \text{dis} \) AND \( \text{leftSensor} < \text{dis} \)

\hspace{1cm} // \text{wall on right side; steer away}

\hspace{1cm} \text{spin left 180 ms}

\hspace{1cm} \text{forward}

Else

\hspace{1cm} \text{forward}

3rd refinement

GENERATE