# RRT Sampling-Based Motion Planning

In [11]:
```
# The autoreload extension will automatically load in new c
ode as you edit files,
# so you don't need to restart the kernel every time
%load_ext autoreload
%autoreload 2

import numpy as np
import matplotlib.pyplot as plt
from P2_rrt import *

plt.rcParams['figure.figsize'] = [8, 8] # Change default fi
gure size
```

```
The autoreload extension is already loaded. To reload it, u
se:
  %reload_ext autoreload
```

## Set up workspace

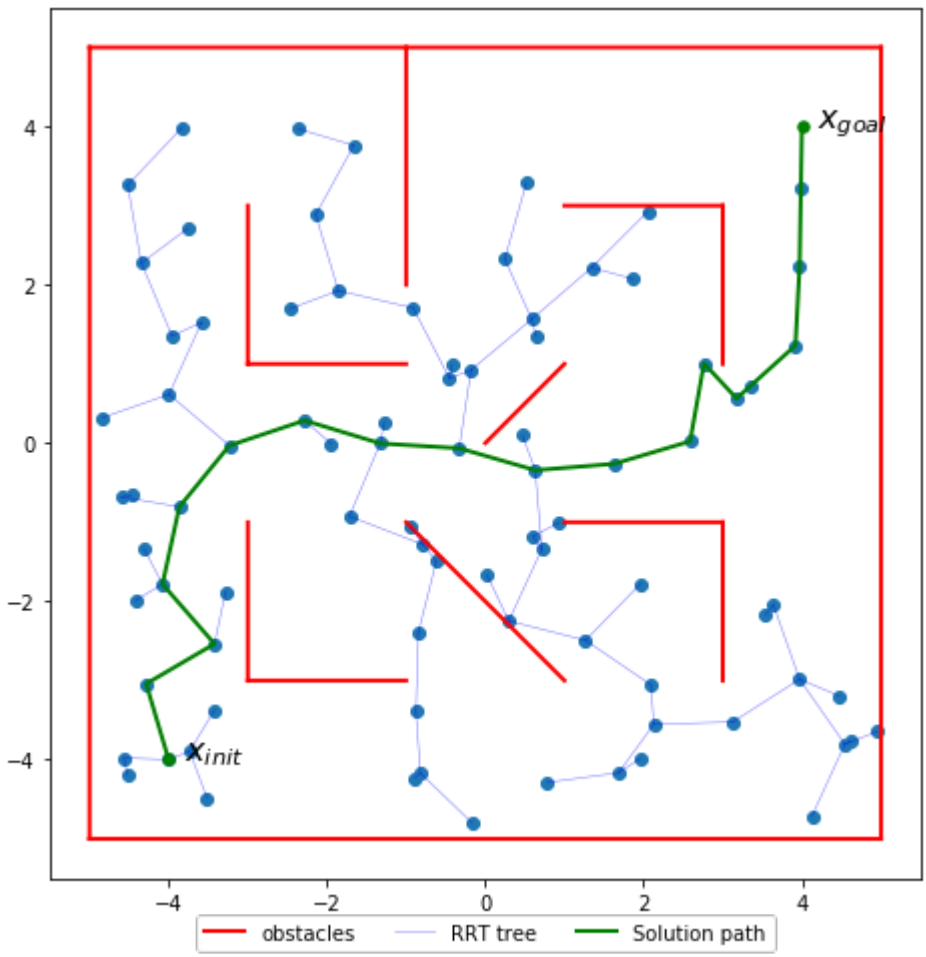
In [12]:
```
MAZE = np.array([
    (( 5, 5), (-5, 5)),
    ((-5, 5), (-5,-5)),
    ((-5,-5), ( 5,-5)),
    (( 5,-5), ( 5, 5)),
    ((-3,-3), (-3,-1)),
    ((-3,-3), (-1,-3)),
    (( 3, 3), ( 3, 1)),
    (( 3, 3), ( 1, 3)),
    (( 1,-1), ( 3,-1)),
    (( 3,-1), ( 3,-3)),
    ((-1, 1), (-3, 1)),
    ((-3, 1), (-3, 3)),
    ((-1,-1), ( 1,-3)),
    ((-1, 5), (-1, 2)),
    (( 0, 0), ( 1, 1))
])

# try changing these!
x_init = [-4,-4] # reset to [-4,-4] when saving results for
submission
x_goal = [4,4] # reset to [4,4] when saving results for sub
mission
```

# Geometric Planning

In [13]:
```
grrt = GeometricRRT([-5,-5], [5,5], x_init, x_goal, MAZE)
grrt.solve(1.0, 2000)
```

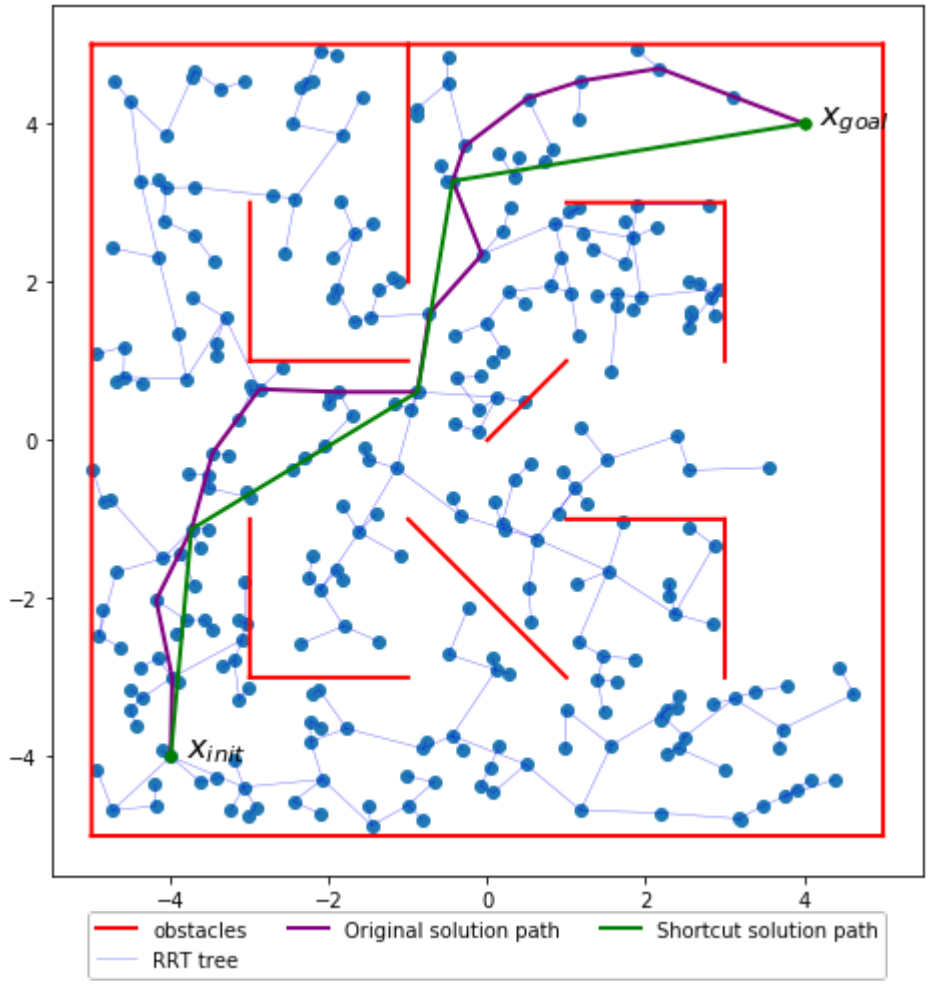
Out[13]: True




## Adding shortcutting

In [14]:
```
grrt.solve(1.0, 2000, shortcut=True)
```

Out[14]: True

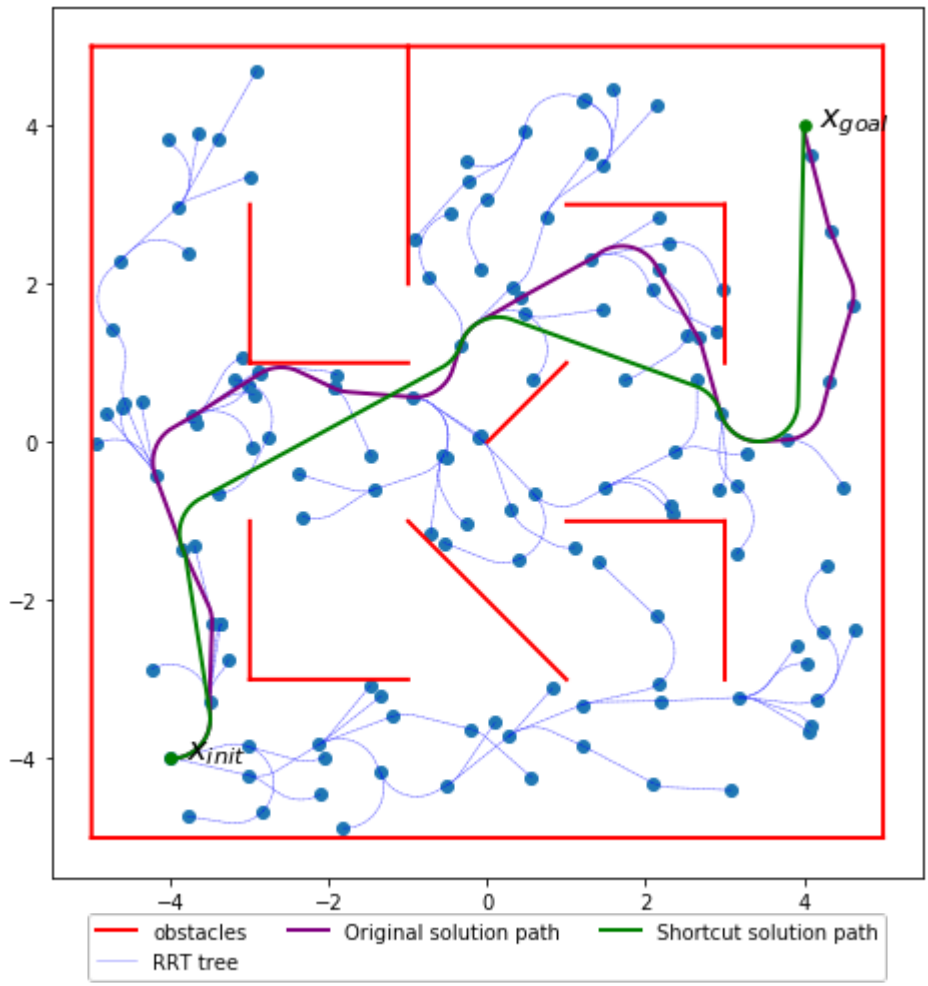



## Dubins Car Planning

In [28]:
```
x_init = [-4,-4,0]
x_goal = [4,4,np.pi/2]

drrt = DubinsRRT([-5,-5,0], [5,5,2*np.pi], x_init, x_goal,
MAZE, .5)
drrt.solve(1.0, 1000, shortcut=True)
```

Out[28]: True




In [26]:
```
x1=[2.99, 3, 0]
x2=[3.,  3.,  0]
eps=1.0
drrt.steer_towards(x1,x2,eps)
```

```
[(2.99, 3.0, 0.0)]
```

Out[26]: [3.0, 3.0, 0]

In [ ]: