```
!pip3 install reeds-shepp
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-whe
Collecting reeds-shepp
  Downloading reeds_shepp-1.0.7.tar.gz (45 kB)
     |████████████████████████████████| 45 kB 4.0 MB/s
Building wheels for collected packages: reeds-shepp
  Building wheel for reeds-shepp (setup.py) ... done
  Created wheel for reeds-shepp: filename=reeds_shepp-1.0.7-cp37-cp37m-linux_x86_
  Stored in directory: /root/.cache/pip/wheels/db/8f/b0/cc244db2ac9927783f636ecb
Successfully built reeds-shepp
Installing collected packages: reeds-shepp
Successfully installed reeds-shepp-1.0.7
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call dr
```

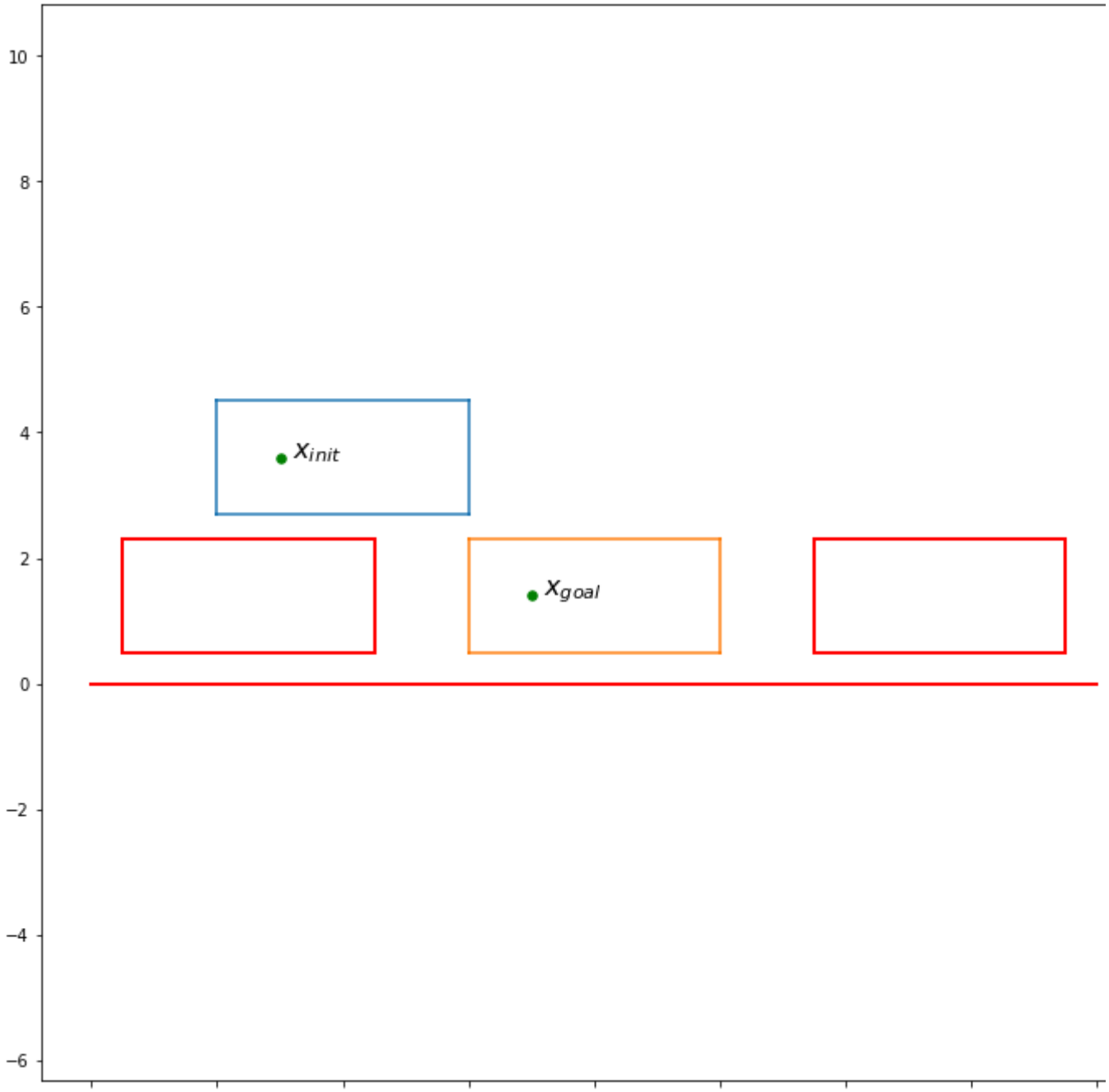
```
%cd /content/drive/My\ Drive/AA274A_HW2-master
```

```
/content/drive/My Drive/AA274A_HW2-master
```

```
# The autoreload extension will automatically load in new code 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 P4_parallel_parking import ParkingRRT

plt.rcParams['figure.figsize'] = [12, 12]  # Change default figure size
```

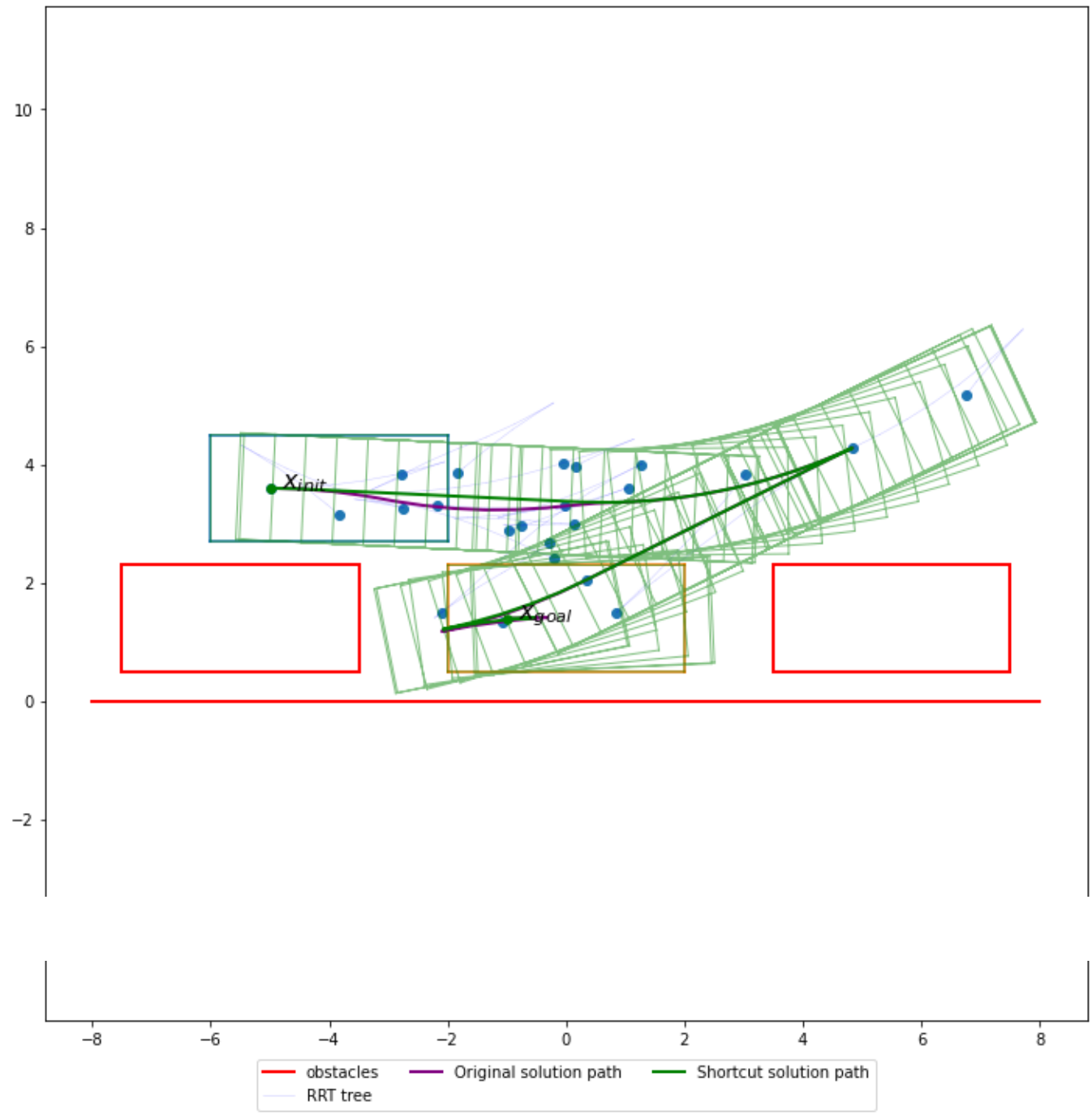
```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

```
x_init = [-5, 3.6, 0]
x_goal = [-1, 1.4, 0]
CAR = np.array([[[-2, -0.9], [2, -0.9]], [[2, -0.9], [2, 0.9]], [[2, 0.9], [-2, 0.9]],
PARKING_SPOT = np.concatenate([CAR + np.array([5.5, 1.4]), CAR + np.array([-5.5, 1.4]
                               np.array([[[-8, 0], [8, 0]]])], 0)

pp_rrt = ParkingRRT([-5, 0, -np.pi / 3], [5, 4, np.pi / 3], x_init, x_goal, PARKING_SP
pp_rrt.plot_problem()
```




```
# RRT is a randomized algorithm; even though this planning problem is feasible, with a
# success is not guaranteed (though we see that with 1000 samples it seems to work mor
# to see the different solutions RRT comes up with, but for debugging you may wish to
# np.random.seed(1235)
pp_rrt.solve(5.0, 1000, shortcut=True)
```

```
True
```


10
8
6
4
2
0
−2
−8
−6
−4
−2
0
2
4
6
8
X_{init}
X_{goal}
obstacles
Original solution path
Shortcut solution path
RRT tree

Colab paid products - Cancel contracts here

✓ 1s completed at 5:02 PM