Name: Fenil Desai (249812226)
Subject: RBE 550 Motion Planning

Motion Planning Assignment-2

Probabilistic Road Map (PRM):

Probabilistic Road Map planning is better than Discrete motion planning in case of high dimensional problems.

PRM consists of two phases:

1. Learning Phase
   - Construction Step
   - Extension Step
2. Query Phase

The four sampling methods that were implemented in this Programing assignment are –

1. Uniform Sampling
2. Random Sampling
3. Gaussian Sampling
4. Bridge Sampling

Let us discuss the advantages and disadvantages of all the above-mentioned sampling methods.

1. Uniform Sampling: This sampling method samples points uniformly; i.e each point is a sampled at a constant distance with respect to the previous point; samples the point in the map.
   - Advantages –
     - Computationally cheap algorithm
     - Divides the map into N equal parts
     - High probability of sampling points near to goal and start pose
     - High probability of finding path if one exists
   - Disadvantages –
     - naively divides the map into N regions
     - Most of the times it does not samples points in a narrow passage
     - Cannot find path if narrow passage exists between the start and the goal

2. Random Sampling: This sampling method randomly samples point in the entire map.
   - Advantages –
     - Computationally Cheap Algorithm

- randomly samples point in the space (requires no guiding algorithm)
- High probability of finding paths if one exists

- Disadvantages –
  - Can sample point away from start and the goal and it might become difficult to connect the start and the goal to the graph
  - Most of the times is does not sample points in a narrow passage
  - Cannot find path if narrow passage exists between the start and the goal

3. Gaussian Sampling: This sampling method first samples a point randomly in the map and then takes the sampled point as the center of the gaussian and samples another point from the gaussian. If both are neither free nor in collision than keeps the one in free space and discards the other.
   - Advantages –
     - Captures the boundaries of the map and the obstacles
     - facilitates path finding in cases of narrow passages
     - High probability of finding a path if one exists
   - Disadvantages –
     - Can sample point away from start and the goal and it might become difficult to connect the start and the goal to the graph
     - Computationally expensive algorithm
     - Requires a guiding algorithm (Gaussian) to sample point in the map

4. Bridge Sampling: This sampling method first samples a point randomly in the map and then check if the sample point is in collision. If it is in collision, then is samples another point in collision using the gaussian centered at the first point. The mid-point of the two sampled is kept as a sampled point if it is collision free.
   - Advantages –
     - Specifically captures the narrow passages.
     - facilitates path finding in cases of narrow passages
     - High probability of finding a path if one exists
   - Disadvantages –
     - Can sample point away from start and the goal and it might become difficult to connect the start and the goal to the graph
     - Computationally very expensive algorithm
     - Requires a guiding algorithm (Gaussian) to sample point in the map

Probabilistic Road Map (PRM) V/S Rapidly Exploring Random Trees (RRT):

This section of the report discusses the comparison between PRM and RRT algorithms.

1. Probabilistic Road Map (PRM):
   - Advantages –
     - Generates a compete graph
     - Multiple sampling methods are possible
     - Nicely handles the problem of narrow passage obstacle
   - Disadvantages –
     - Multi-query method
     - Computationally expensive
     - unnecessarily calculates the weights of all the edges
     - Difficult to always ensure the connection between the start and/or the goal with the rest of the graph
     - Improper connectivity and limited coverage

2. Rapidly Exploring Random Trees (RRT):
   - Advantages –
     - Computationally cheaper algorithm
     - Single-Query method
     - Does not calculate all the possible edge costs unnecessarily
     - Always ensures connection between the tree and the start and/or the goal
     - Numerous variants are available
     - A bias can be added to quickly explore the tree in the direction of the goal
     - Probabilistic Complete
   - Disadvantages –
     - No quality of paths guarantee
     - No termination when there is no solution
     - Does not generate a complete graph

RRT V/S RRT-Star (RRT*):

1. RRT: To implement RRT algorithm a point is randomly sampled in the map with bias of 5-10% of selecting the goal as this random point. Then a nearest node of the tree is found to this point. After that a node is extended in the direction of the randomly sampled point by a fixed step size. If this node is in collision free region, it is connected to the tree and the cost to travel form the nearest node to this new node is also

calculated. If by chance this nearest node is in the vicinity of the goal, the exploration of the tree is stopped and this new is directly connected to the goal. This is how RRT search finds a path from the start to the goal. This algorithm is terminated once the goal is connected to the tree.

2. RRT*: The initial steps are identical to the RRT algorithm. Once a new node is added to the tree, a list of neighbors of the new node is created and rewiring operation is carried out on all the neighbors of the new node. Rewiring essentially means disconnecting and reconnecting the node to the tree in such a way that the cost to reach that particular node is minimum, here the cost used is the Euclidean distance. This algorithm is not terminated even after it finds one path from the start to the goal, as rewiring can help find shorter path if RRT* algorithm is made to run a little longer.

The RRT algorithm has more spread-out branches of the trees and has smaller trees as the algorithm is terminated as it finds one path, whereas the RRT* has more of a directional kind of graph but has larger trees since it is not terminated as soon as it finds the first path.

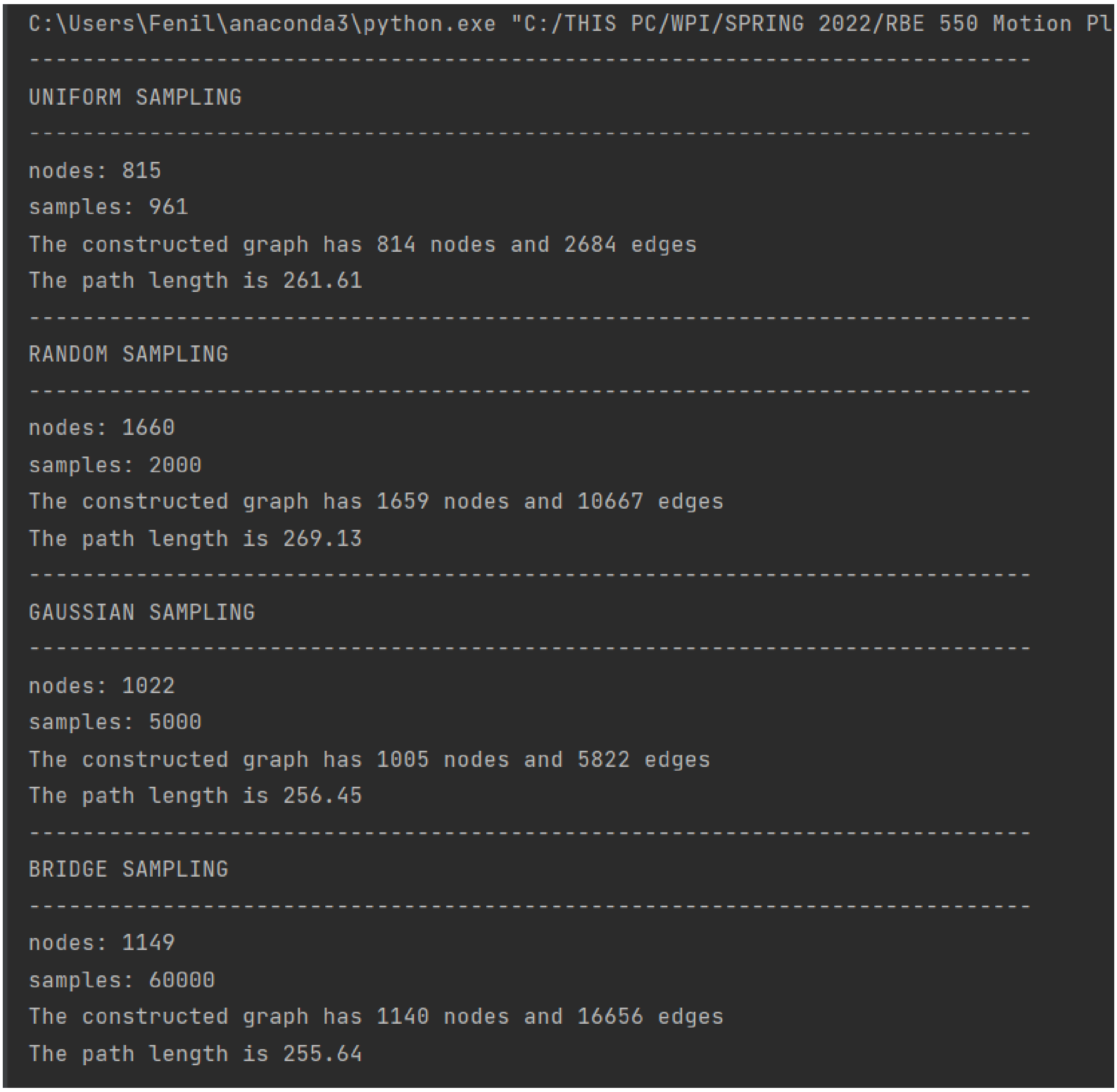
This helps us conclude that RRT algorithm is more efficient since it finds a feasible path in little computation however it does not guarantee optimality. On the other hand, RRT* has higher computational complexity but it has the potential to find a shorter path and thus produce more optimal path as compared to RRT algorithm. However, it still does not guarantee to produce the best possible path in the map.

Algorithm Results and Explanation:

1. Probabilistic Road Map:
Three experiments were conducted for this implementation.
The important parameters for the implementation of PRM algorithms are as follows:
- Steps for collision check = 50
- cost = Euclidean Distance
- Standard Deviation for Gaussian and Bridge sampling = 30 in both row and column directions
- Radius for K-D tree implementation = 15
- Radius to connect start to the graph = 35
- Radius to connect start to the graph = 65
- Start Position = (200,75)
- Goal Position = (30,250)

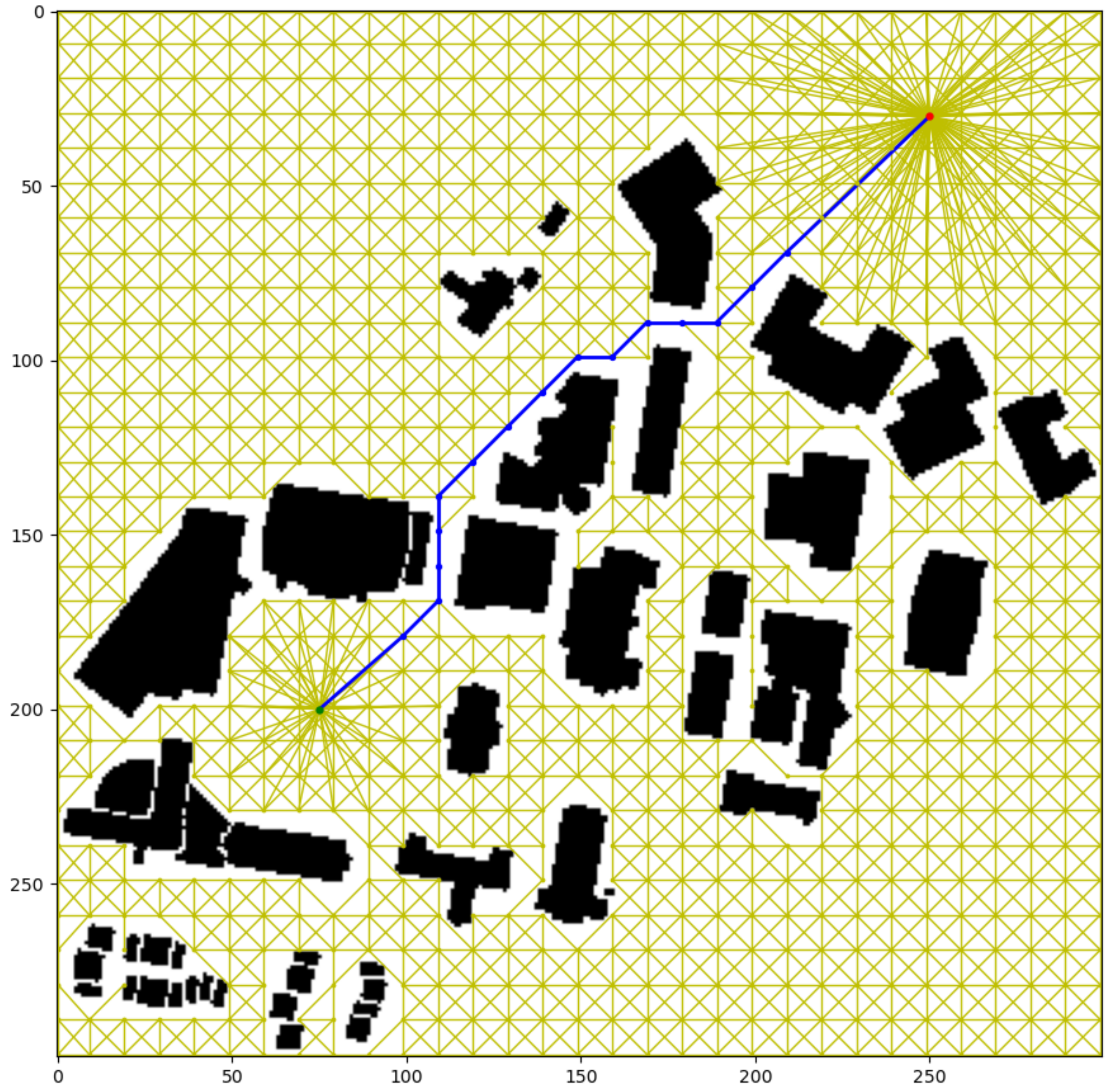
- Experiment 1 –
Number of sampling points for Uniform Sampling = 1000
Number of sampling points for Random Sampling = 2000
Number of sampling points for Gaussian Sampling = 5000
Number of sampling points for Bridge Sampling = 30000

Summary of Experiment 1 -

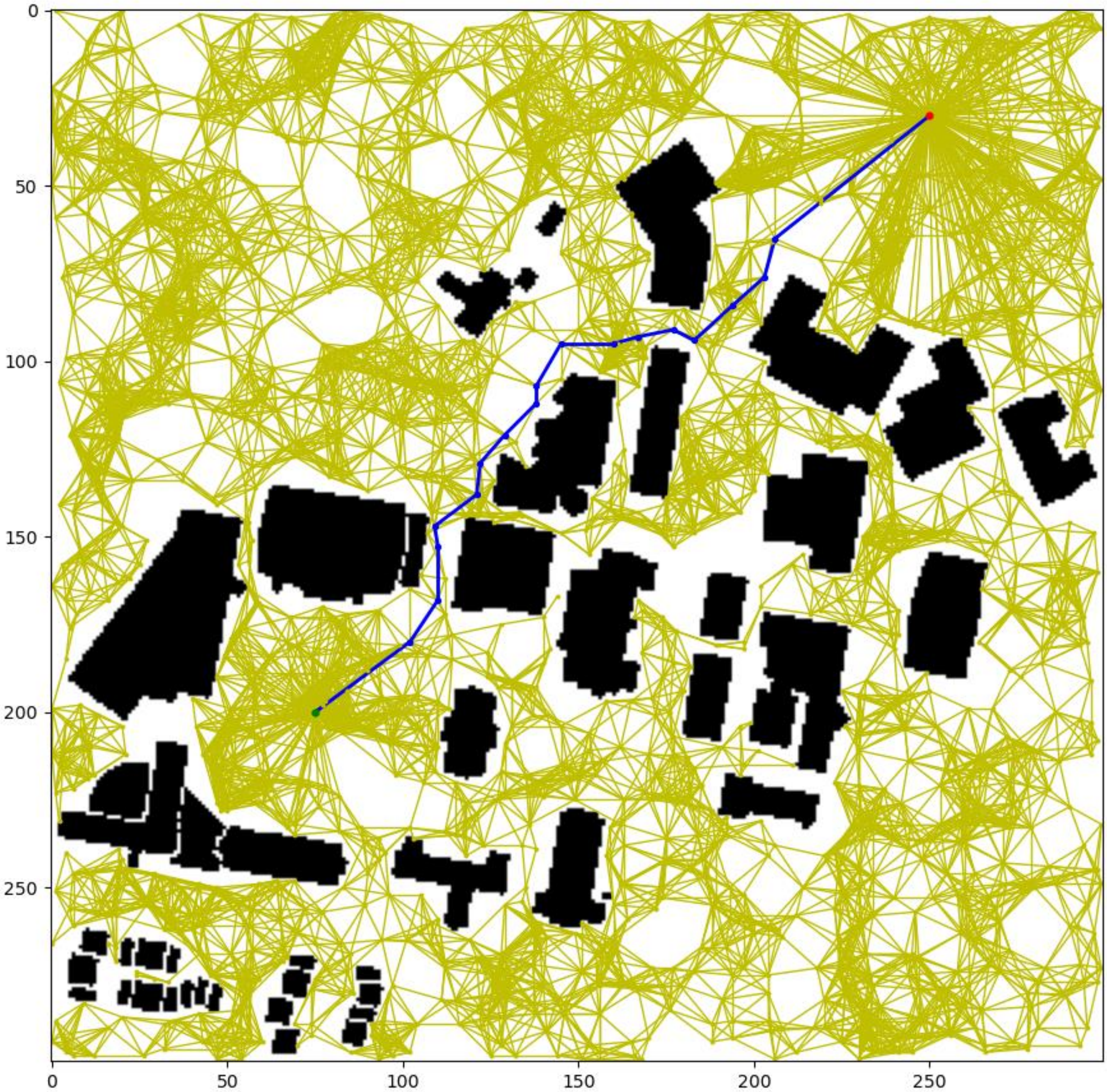


```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion Pl
---------------------------------------------------------------------------
UNIFORM SAMPLING
---------------------------------------------------------------------------
nodes: 815
samples: 961
The constructed graph has 814 nodes and 2684 edges
The path length is 261.61
---------------------------------------------------------------------------
RANDOM SAMPLING
---------------------------------------------------------------------------
nodes: 1660
samples: 2000
The constructed graph has 1659 nodes and 10667 edges
The path length is 269.13
---------------------------------------------------------------------------
GAUSSIAN SAMPLING
---------------------------------------------------------------------------
nodes: 1022
samples: 5000
The constructed graph has 1005 nodes and 5822 edges
The path length is 256.45
---------------------------------------------------------------------------
BRIDGE SAMPLING
---------------------------------------------------------------------------
nodes: 1149
samples: 60000
The constructed graph has 1140 nodes and 16656 edges
The path length is 255.64
```

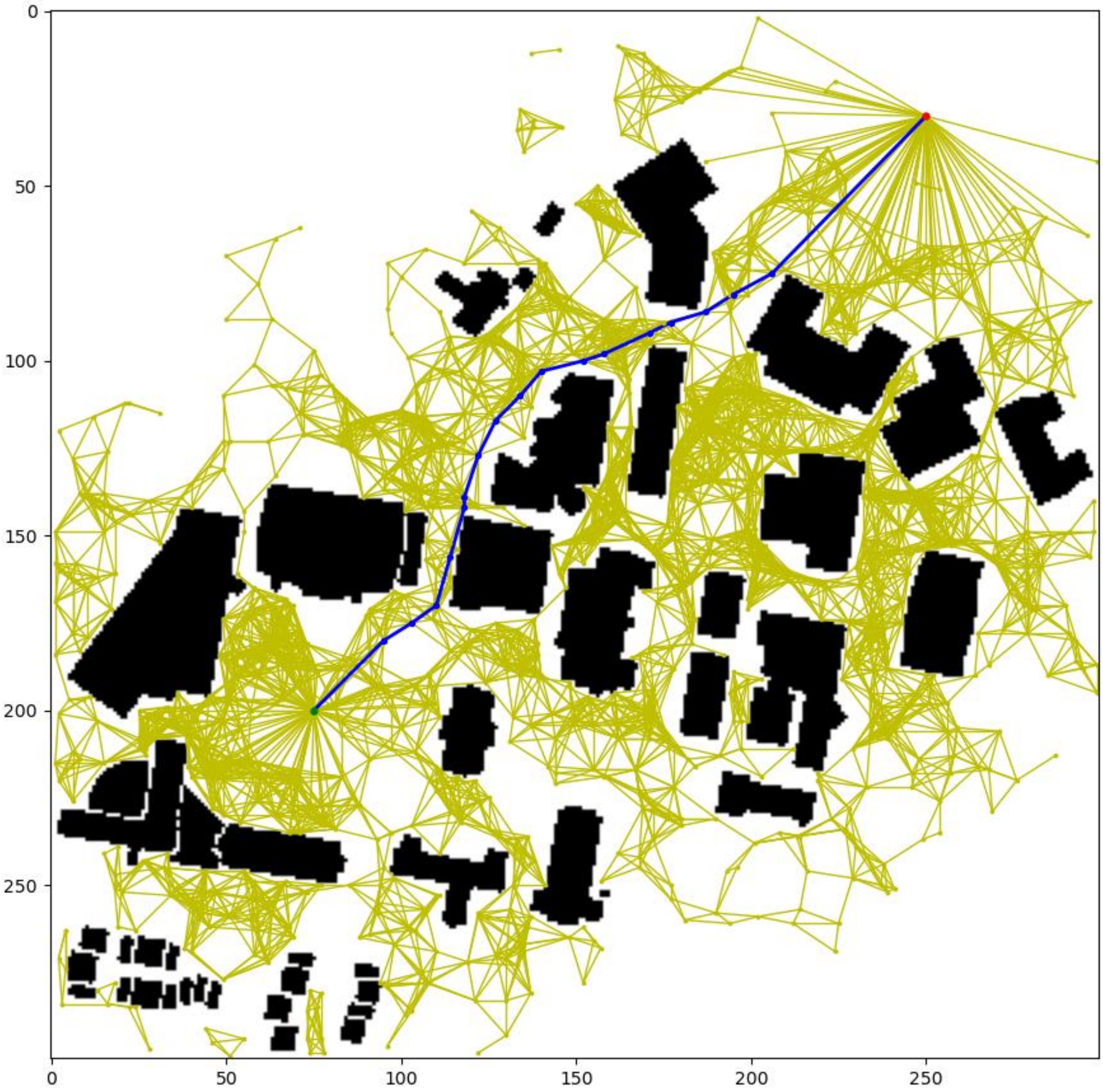
Uniform Sampling –



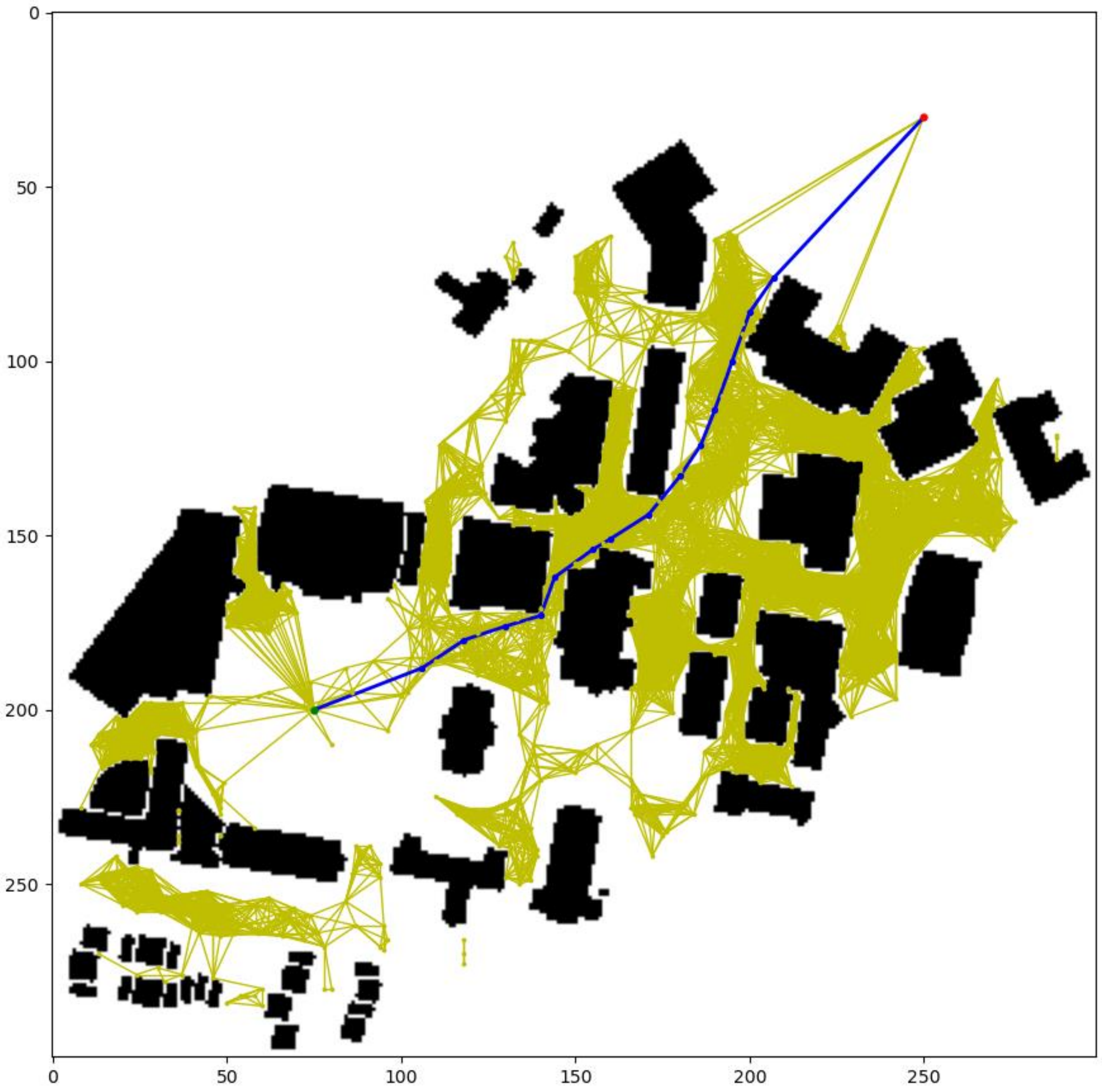
Random Sampling –



Gaussian Sampling –

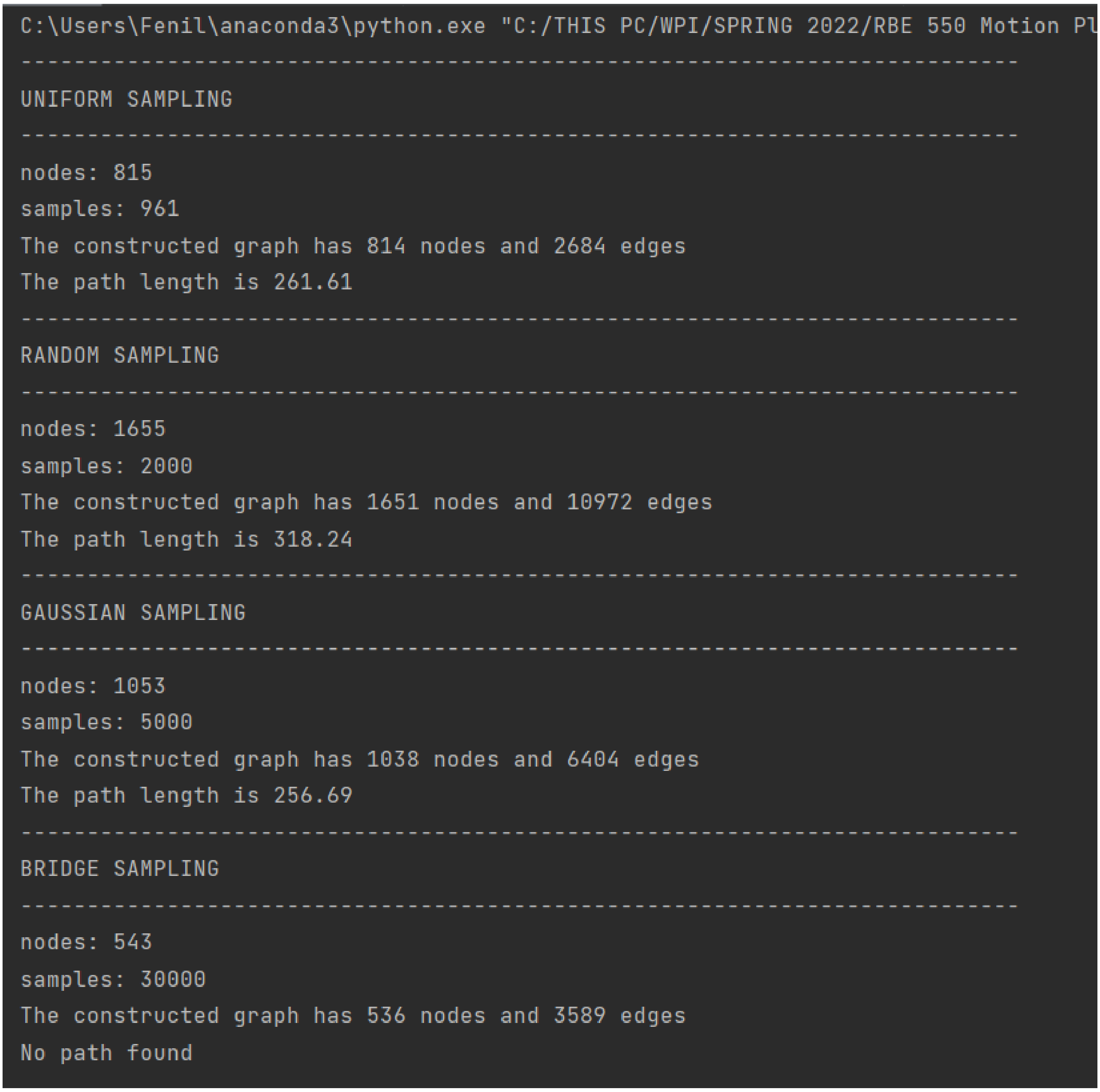


Bridge Sampling –



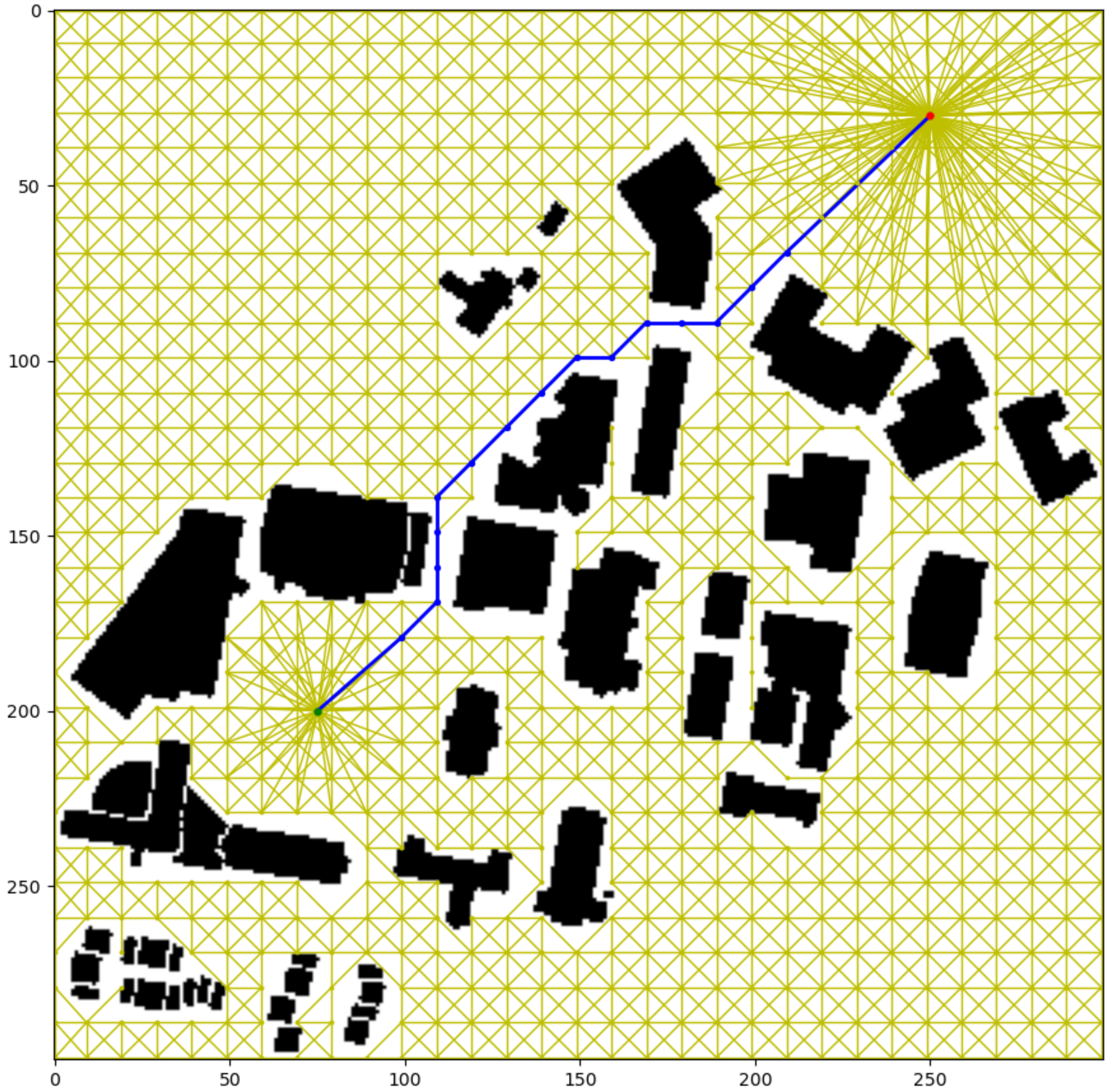
- Experiment 2 –
  Same parameters were used as in case of Experiment 1
  We can see from the summary of Experiment 1 and Experiment 2 that even though same parameters were used different paths were obtained in case of Random Sampling, Gaussian Sampling and Bridge Sampling. And we can also conclude that this algorithm does not perform one better than the other always as the points are randomly sampled in all the cases, sometimes one algorithm performs better than the other in the same problem setting with same start and the end goals. However, uniform sampling produces the same results in always in same problem setting.

Summary of Experiment 2 –

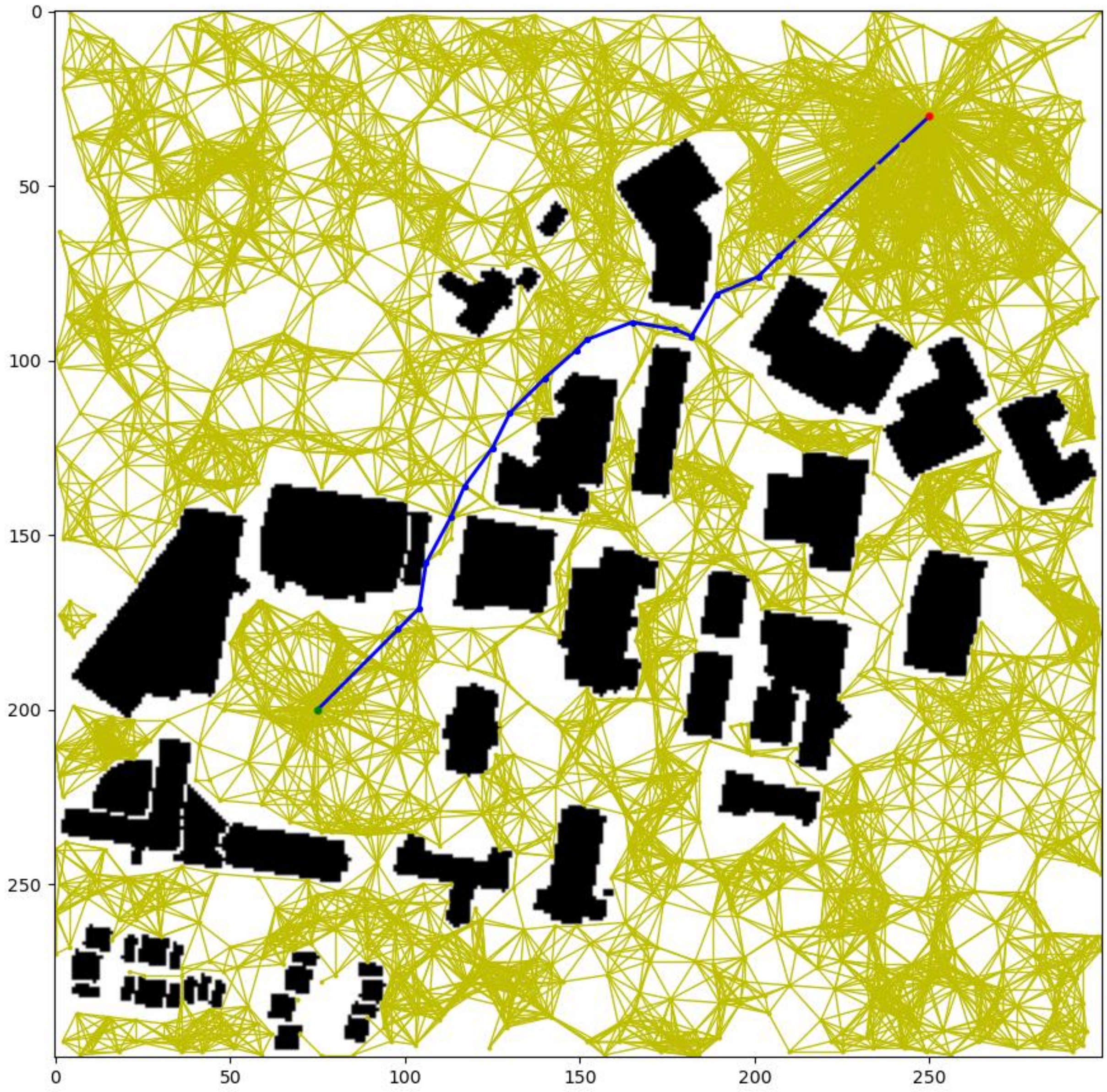


```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion Pl
---------------------------------------------------------------------------
UNIFORM SAMPLING
---------------------------------------------------------------------------
nodes: 815
samples: 961
The constructed graph has 814 nodes and 2684 edges
The path length is 261.61
---------------------------------------------------------------------------
RANDOM SAMPLING
---------------------------------------------------------------------------
nodes: 1655
samples: 2000
The constructed graph has 1651 nodes and 10972 edges
The path length is 318.24
---------------------------------------------------------------------------
GAUSSIAN SAMPLING
---------------------------------------------------------------------------
nodes: 1053
samples: 5000
The constructed graph has 1038 nodes and 6404 edges
The path length is 256.69
---------------------------------------------------------------------------
BRIDGE SAMPLING
---------------------------------------------------------------------------
nodes: 543
samples: 30000
The constructed graph has 536 nodes and 3589 edges
No path found
```

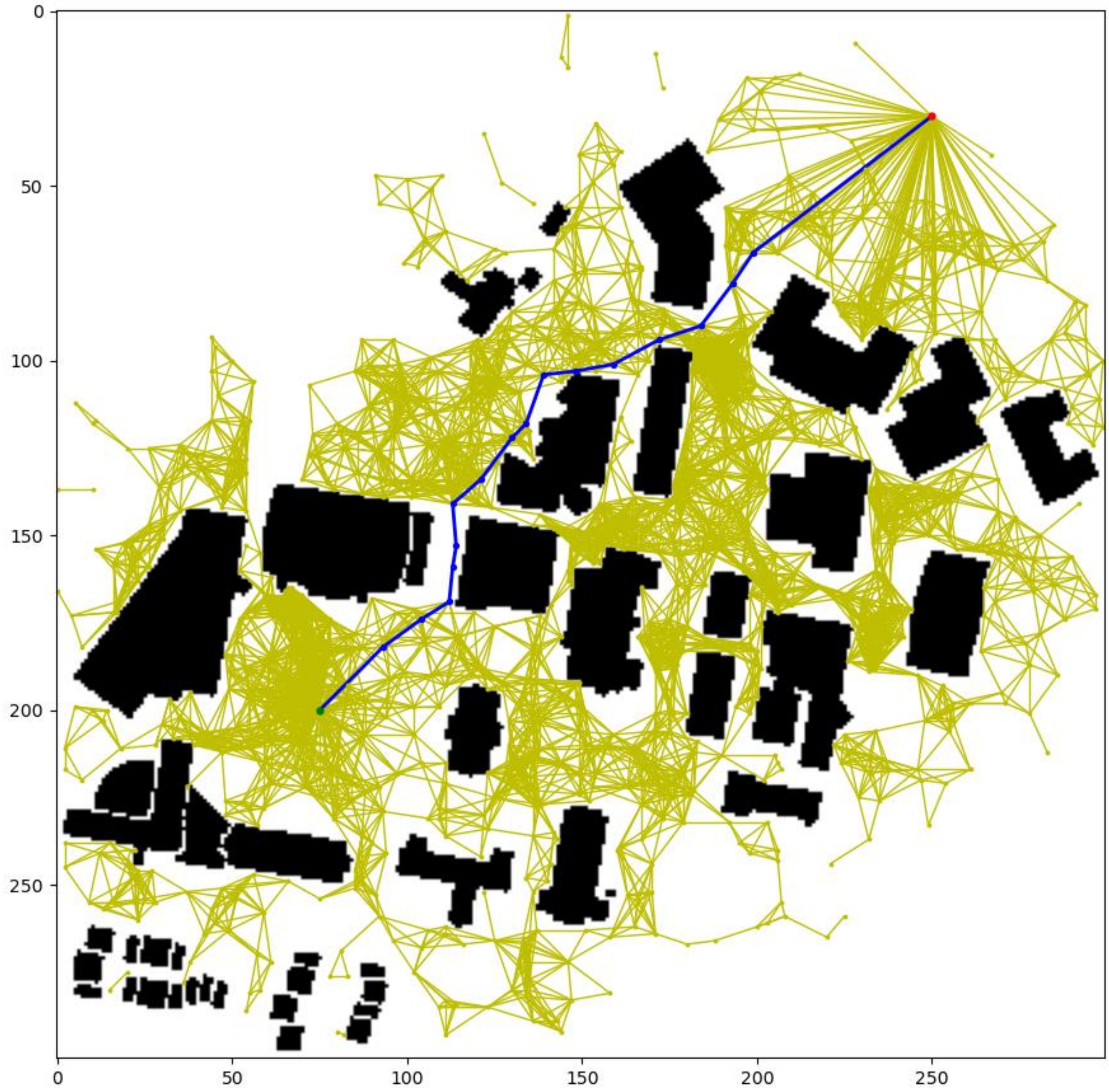
Uniform Sampling –



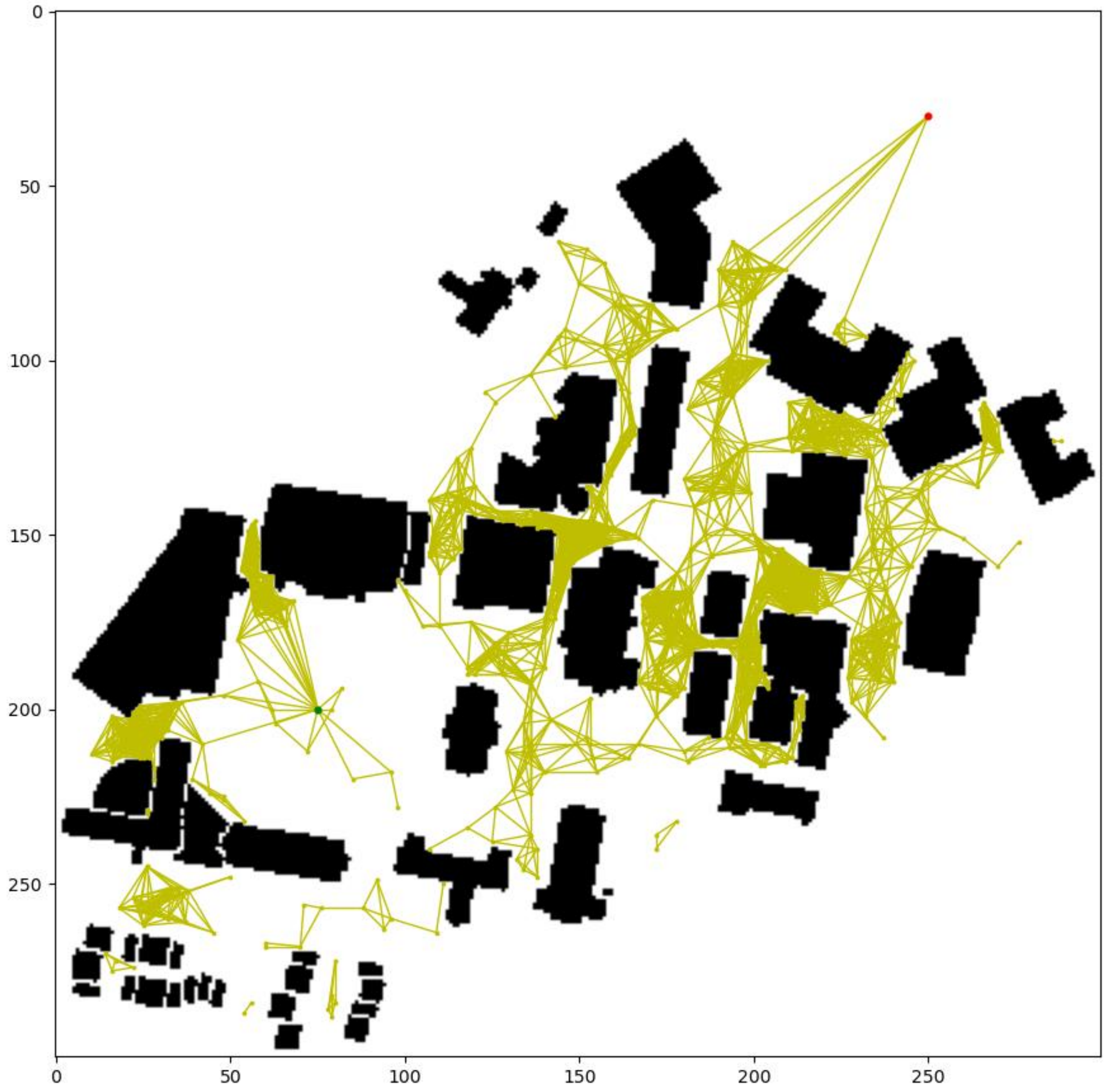
Random Sampling –



Gaussian Sampling –

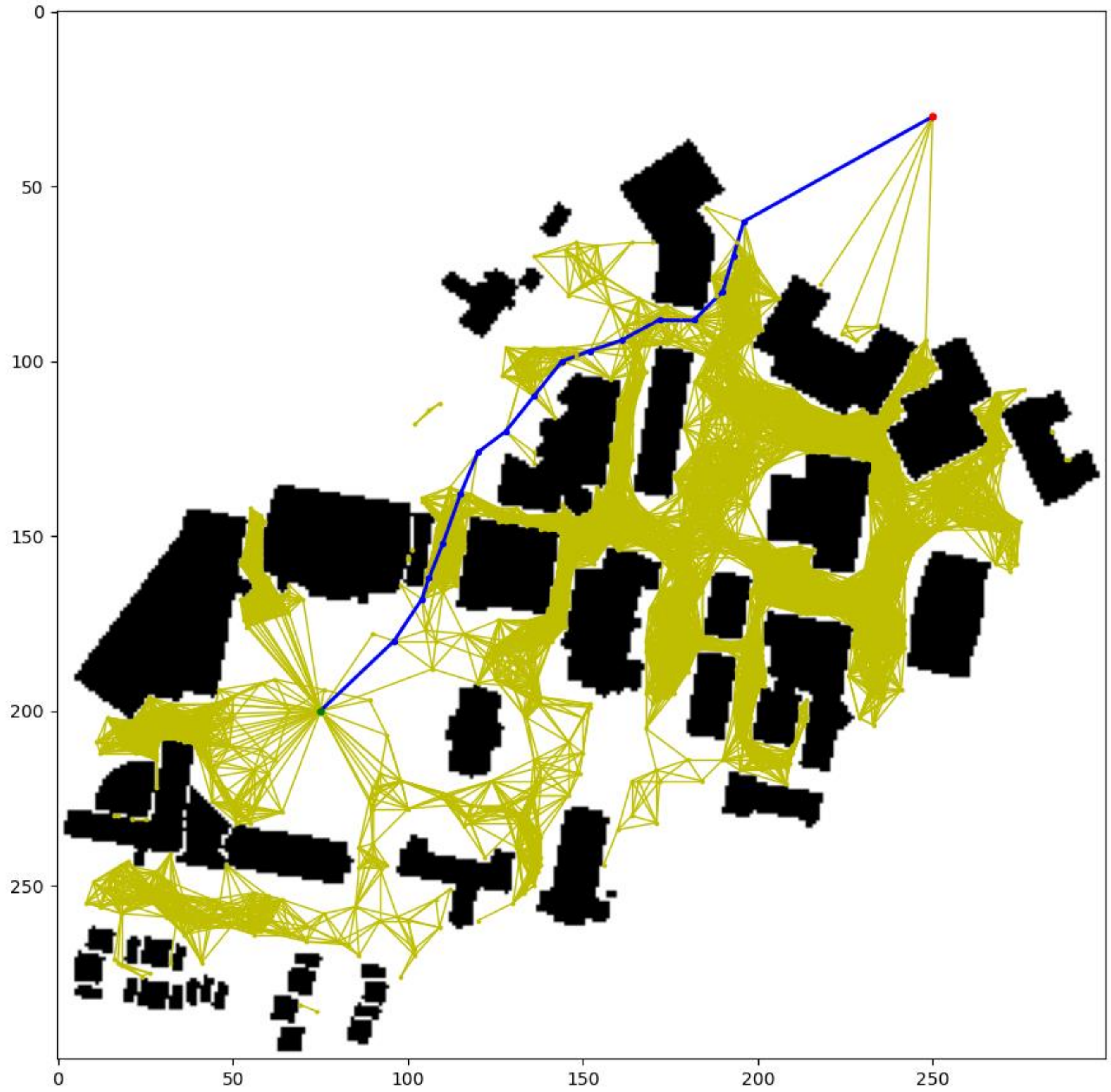


Bridge Sampling –



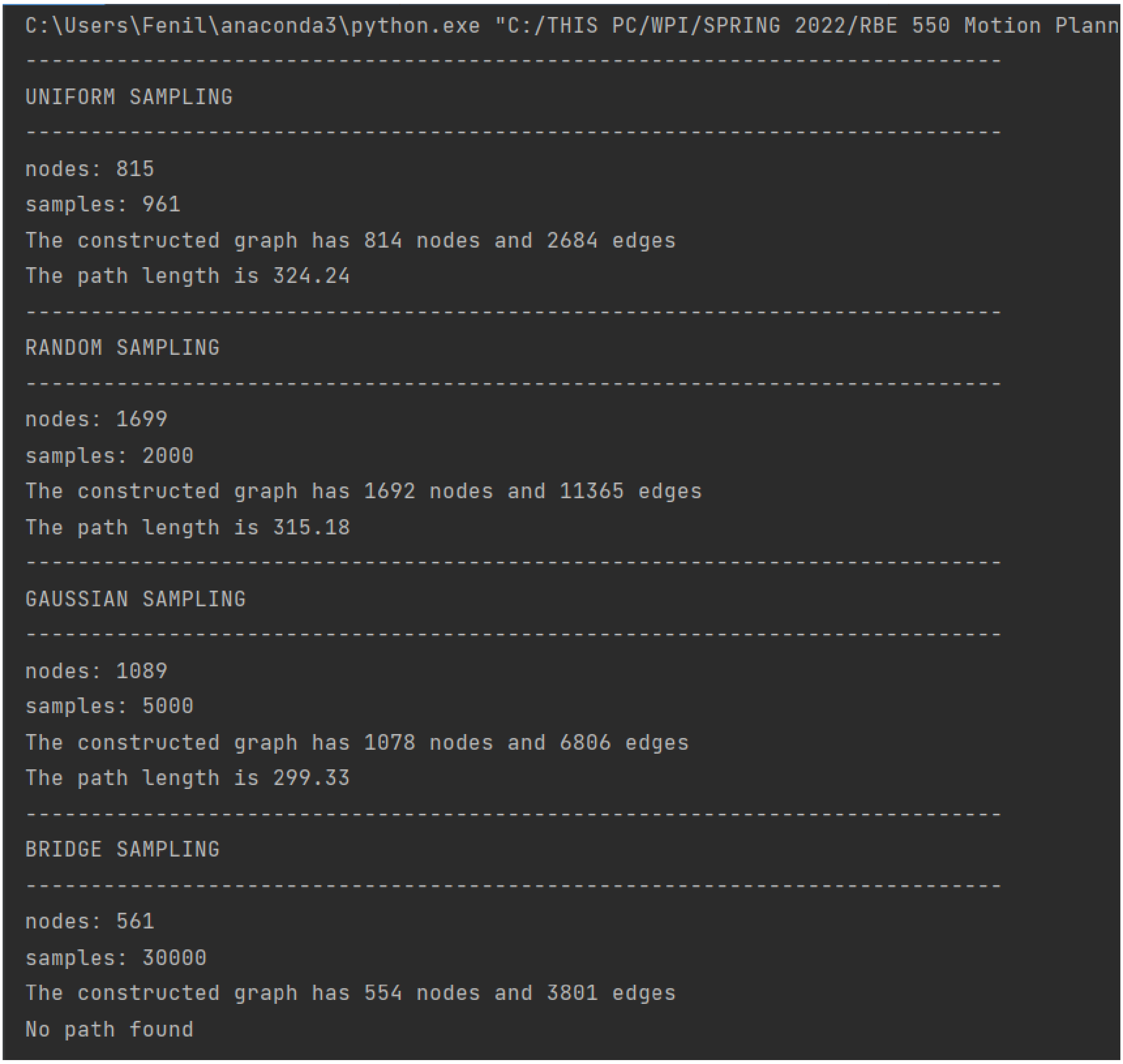
Here in this case, we can observe that simply using the Bridge Sampling may not be able to find the path always. So, we can conclude that using bridge sampler individually may not be able to find path always but if we sample more points, it will be able to find the path in some cases.

Here, I sampled 60000 points instead of 30000 to get a feasible by using only the bridge sampler.



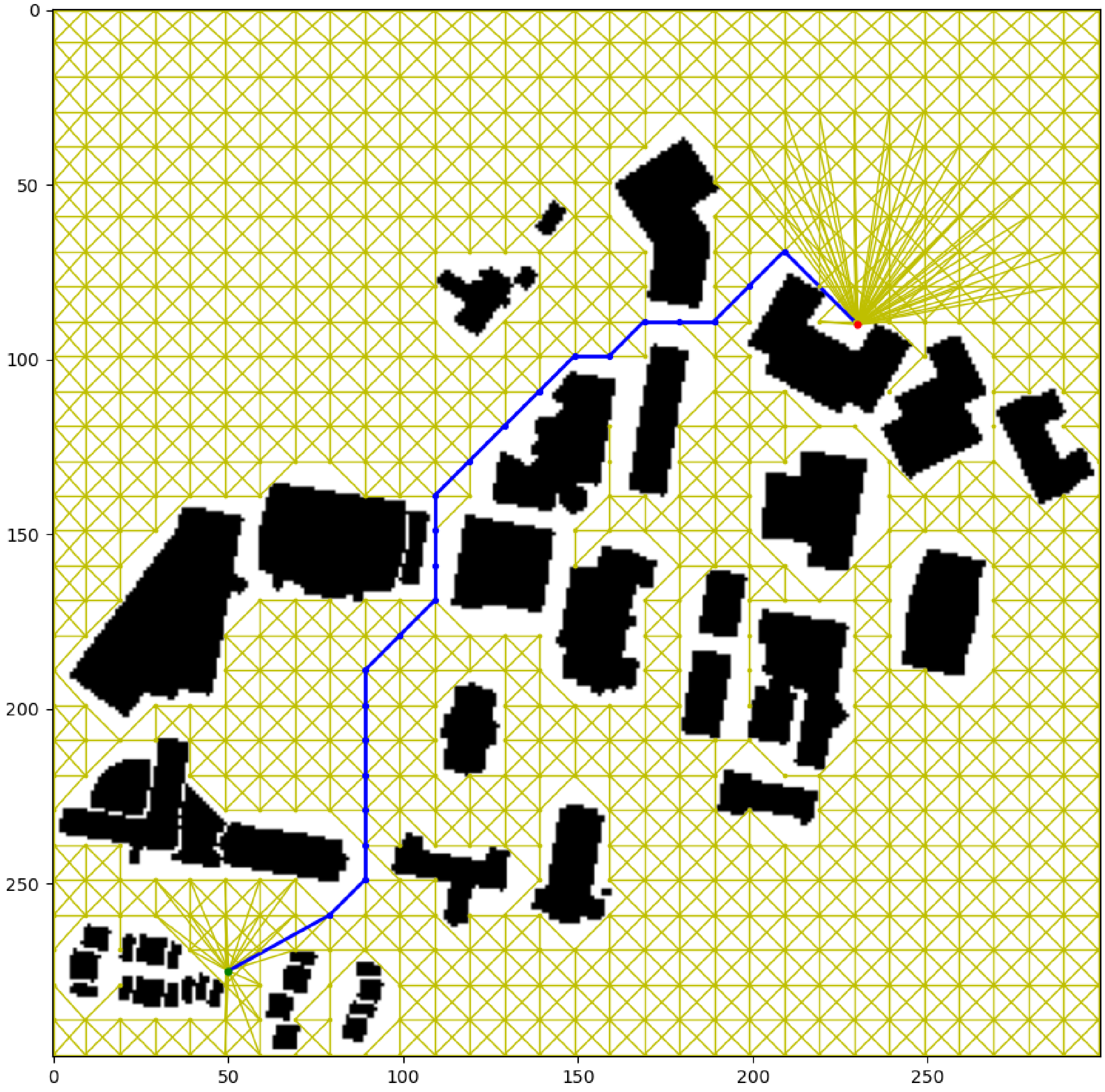
- Experiment 3 –
  This experiment used all the same parameters as the first experiment but here the Start Pose is changed to (275,50) and Goal Pose is changed to (90,230). This combination of start and goal makes it even more to plan the path using PRM method. However, all the sampler methods were able to find a feasible path.

Summary of Experiment 3 –

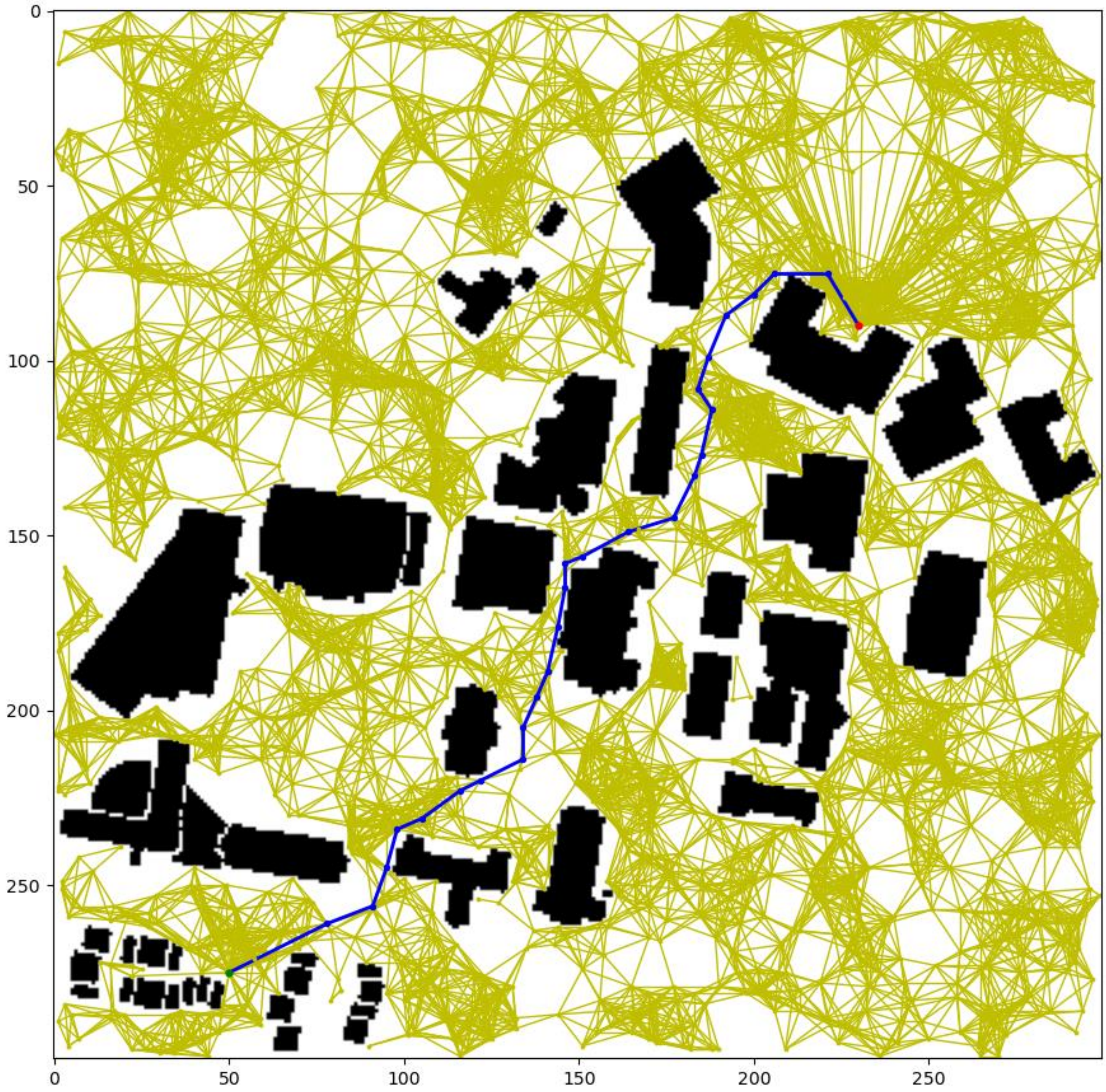


```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion Plann
---------------------------------------------------------------------------
UNIFORM SAMPLING
---------------------------------------------------------------------------
nodes: 815
samples: 961
The constructed graph has 814 nodes and 2684 edges
The path length is 324.24
---------------------------------------------------------------------------
RANDOM SAMPLING
---------------------------------------------------------------------------
nodes: 1699
samples: 2000
The constructed graph has 1692 nodes and 11365 edges
The path length is 315.18
---------------------------------------------------------------------------
GAUSSIAN SAMPLING
---------------------------------------------------------------------------
nodes: 1089
samples: 5000
The constructed graph has 1078 nodes and 6806 edges
The path length is 299.33
---------------------------------------------------------------------------
BRIDGE SAMPLING
---------------------------------------------------------------------------
nodes: 561
samples: 30000
The constructed graph has 554 nodes and 3801 edges
No path found
```

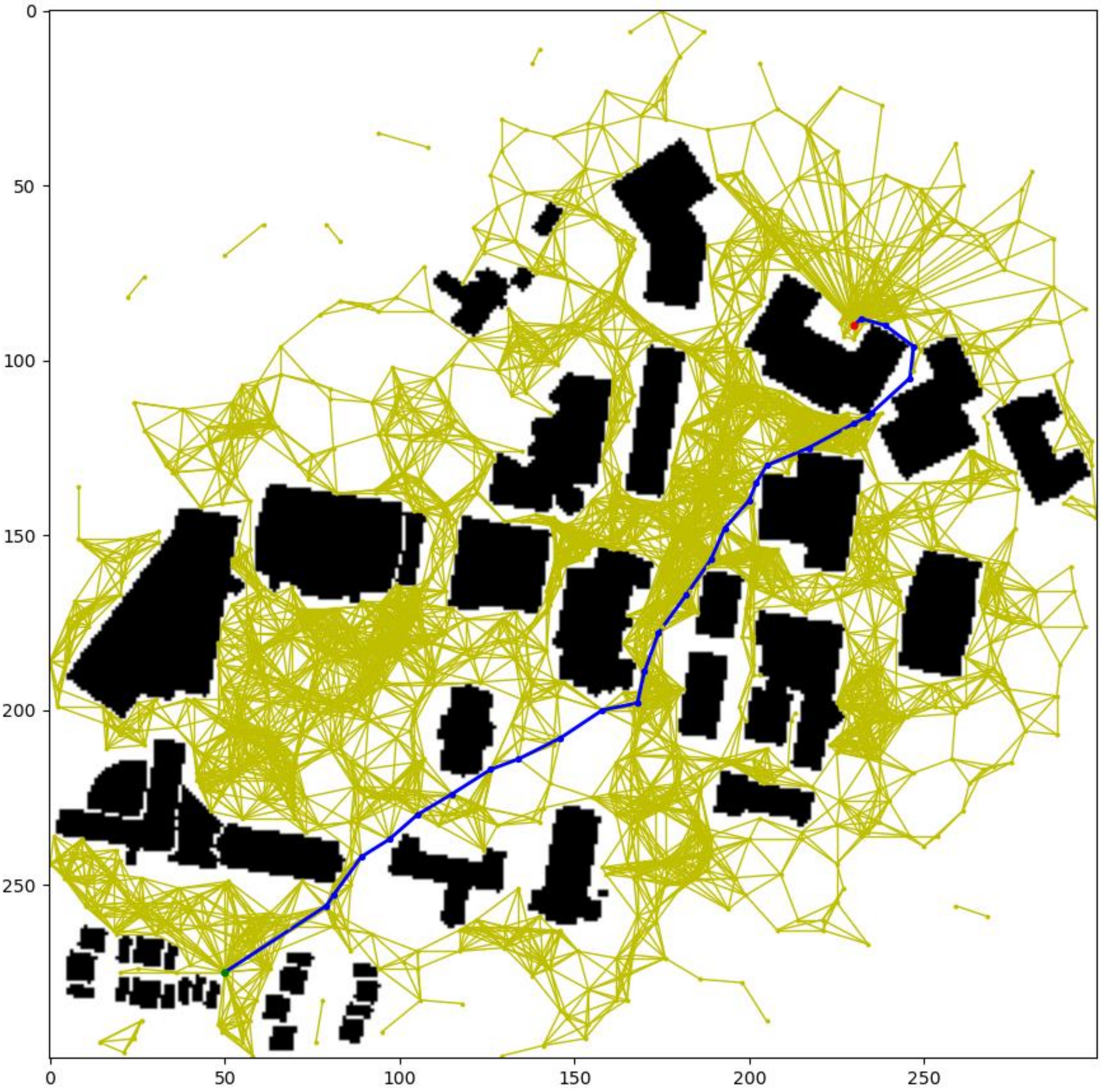
Uniform Sampling –



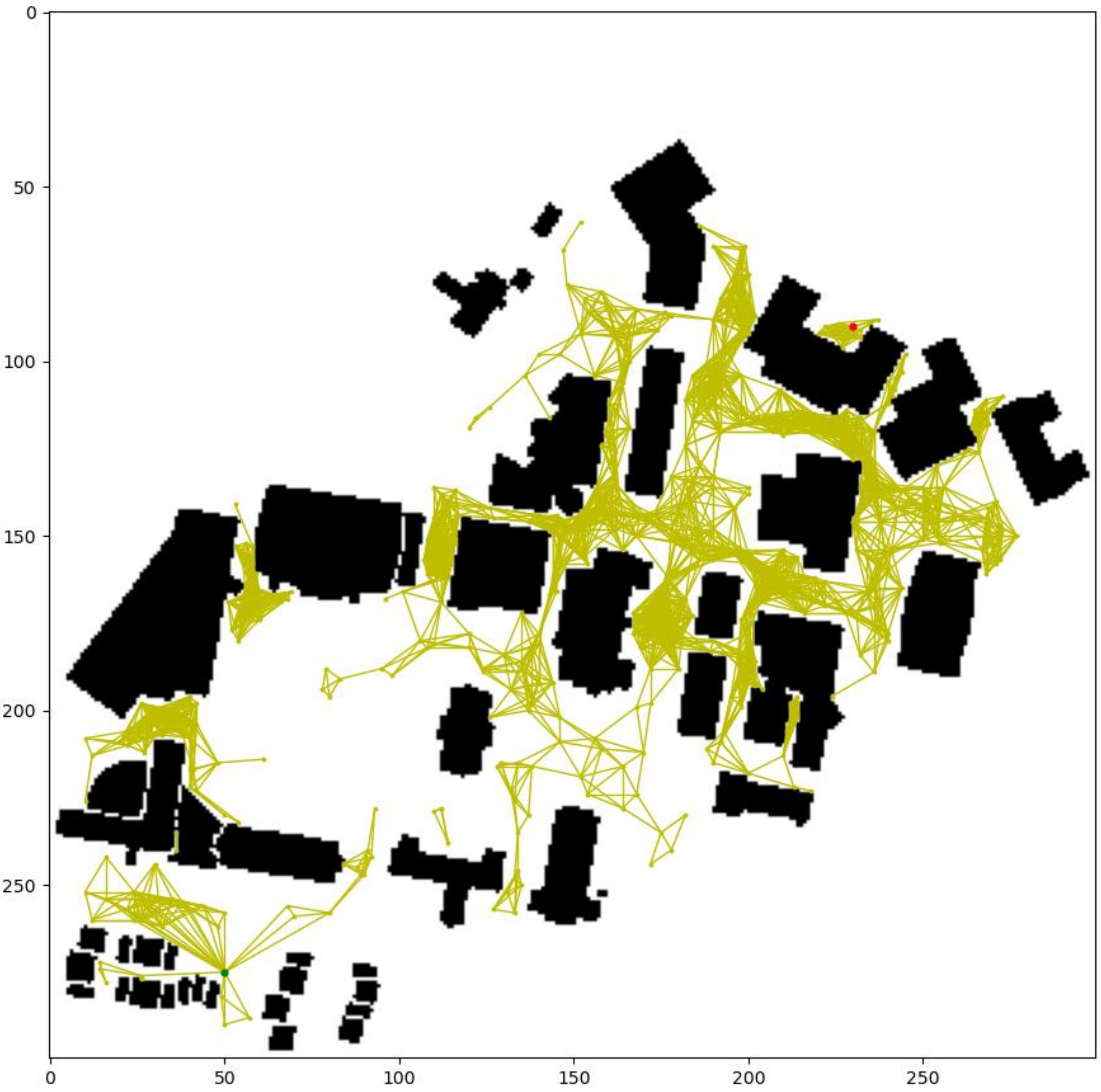
Random Sampling –



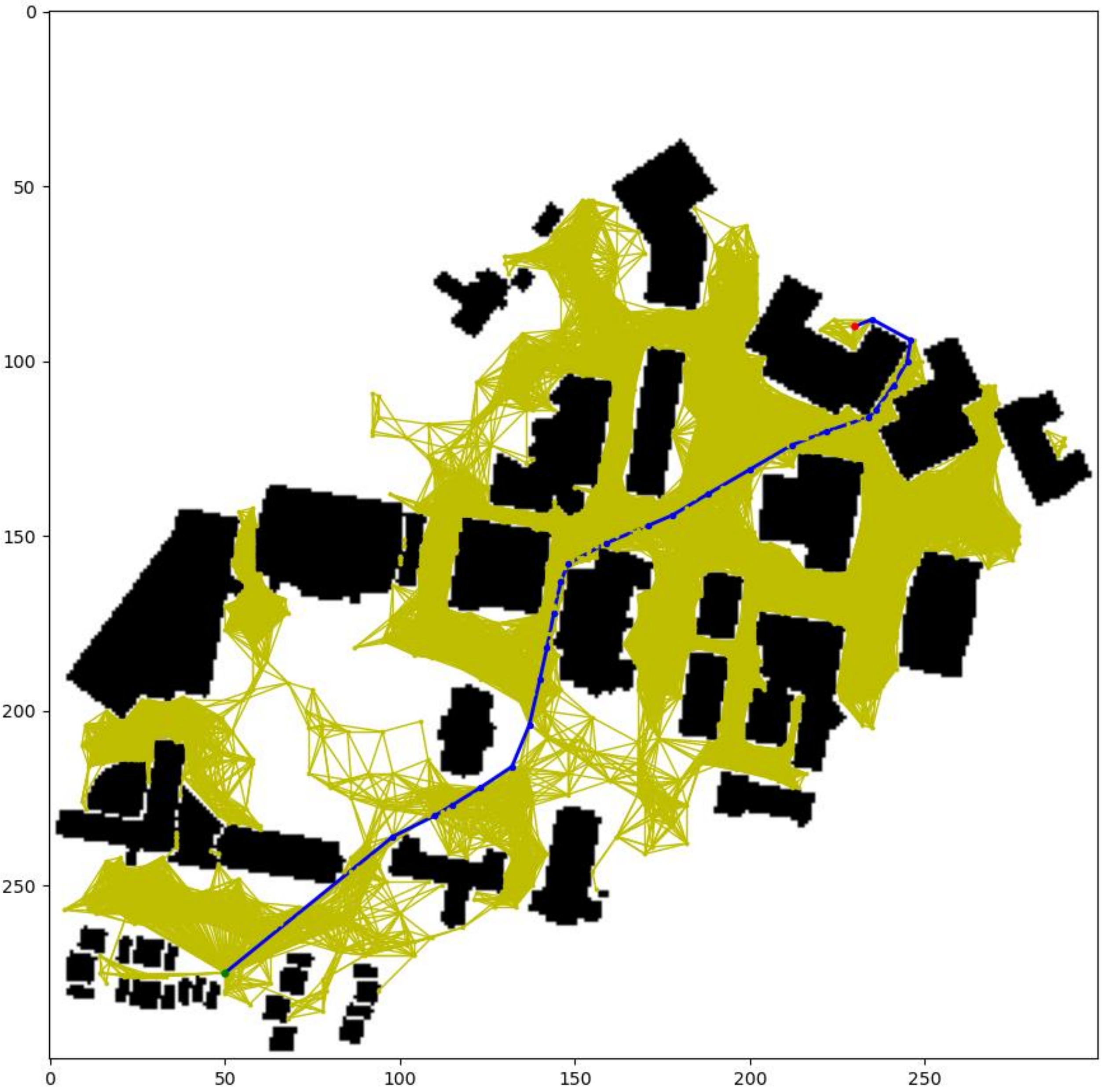
Gaussian Sampling –



Bridge Sampling –



In this case bridge sampler was not able to find the path. Here, I increased the sampling points to 120000 and increased the Start vicinity and Goal vicinity radii to 100 to obtain the result shown below.


0
50
100
150
200
250
0
50
100
150
200
250

2. RRT and RRT*:

   Four experiments were conducted for this implementation.

   The important parameters for the implementation of RRT and RRT* algorithms are as follows:

   - Step size for node extension = 10

   - cost = Euclidean Distance

   - Standard Deviation for Gaussian and Bridge sampling = 30 in both row and column directions

   - Radius for goal neighborhood (Goal Vicinity) = 15

   - Radius to find the neighbor list for a new node = 20

   - Start Position = (200,75)

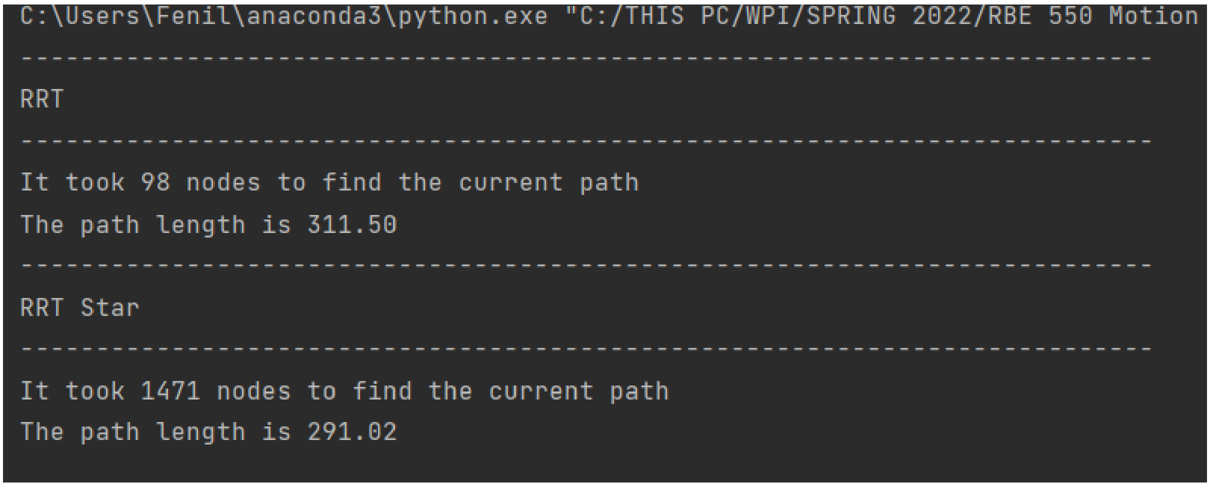
   - Goal Position = (30,250)

   - Experiment 1 –

     Number of sampling points for RRT = 1000

     Number of sampling points for RRT* = 2000

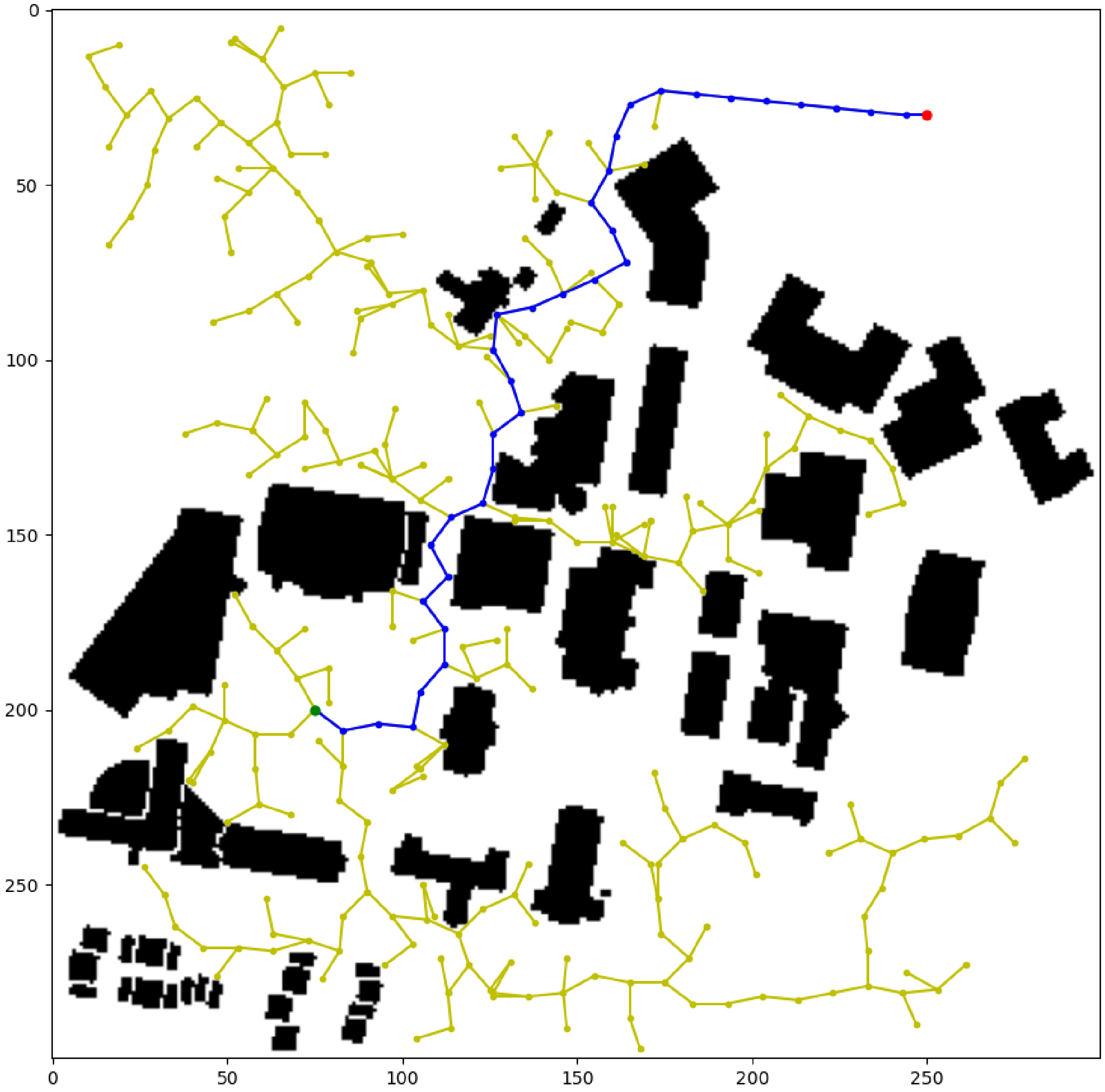
Summary of Experiment 1 –



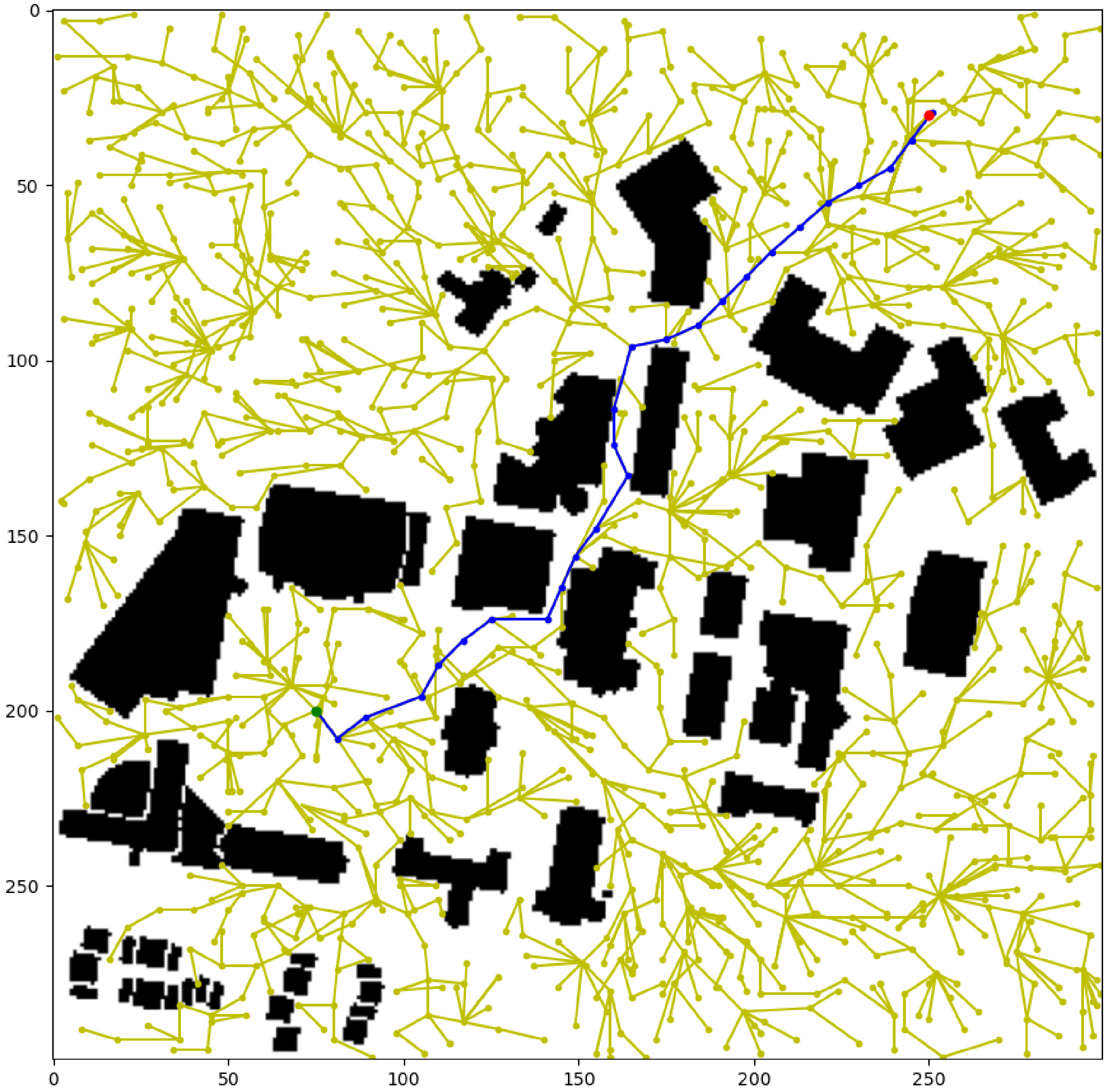
```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion
---------------------------------------------------------------------------
RRT
---------------------------------------------------------------------------
It took 98 nodes to find the current path
The path length is 311.50
---------------------------------------------------------------------------
RRT Star
---------------------------------------------------------------------------
It took 1471 nodes to find the current path
The path length is 291.02
```

Here, we can observe that RRT* produces shorter path as compared to RRT in this case.

RRT –

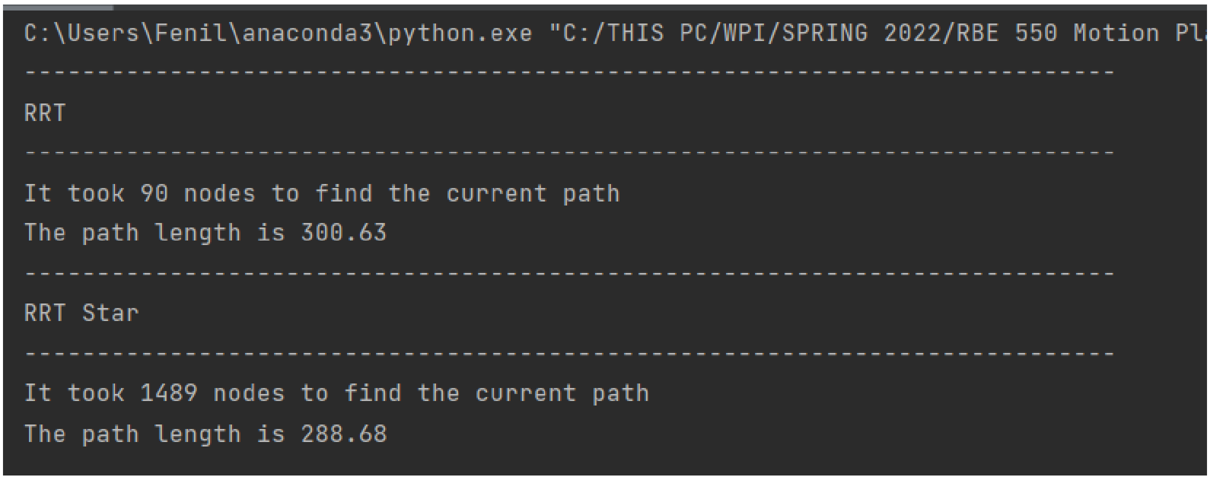


RRT* –



- Experiment 2 –
  In this experiment we try to sample more point in case of both RRT and RRT* and observe what happens.
  Number of sampling points for RRT = 2000
  Number of sampling points for RRT* = 4000

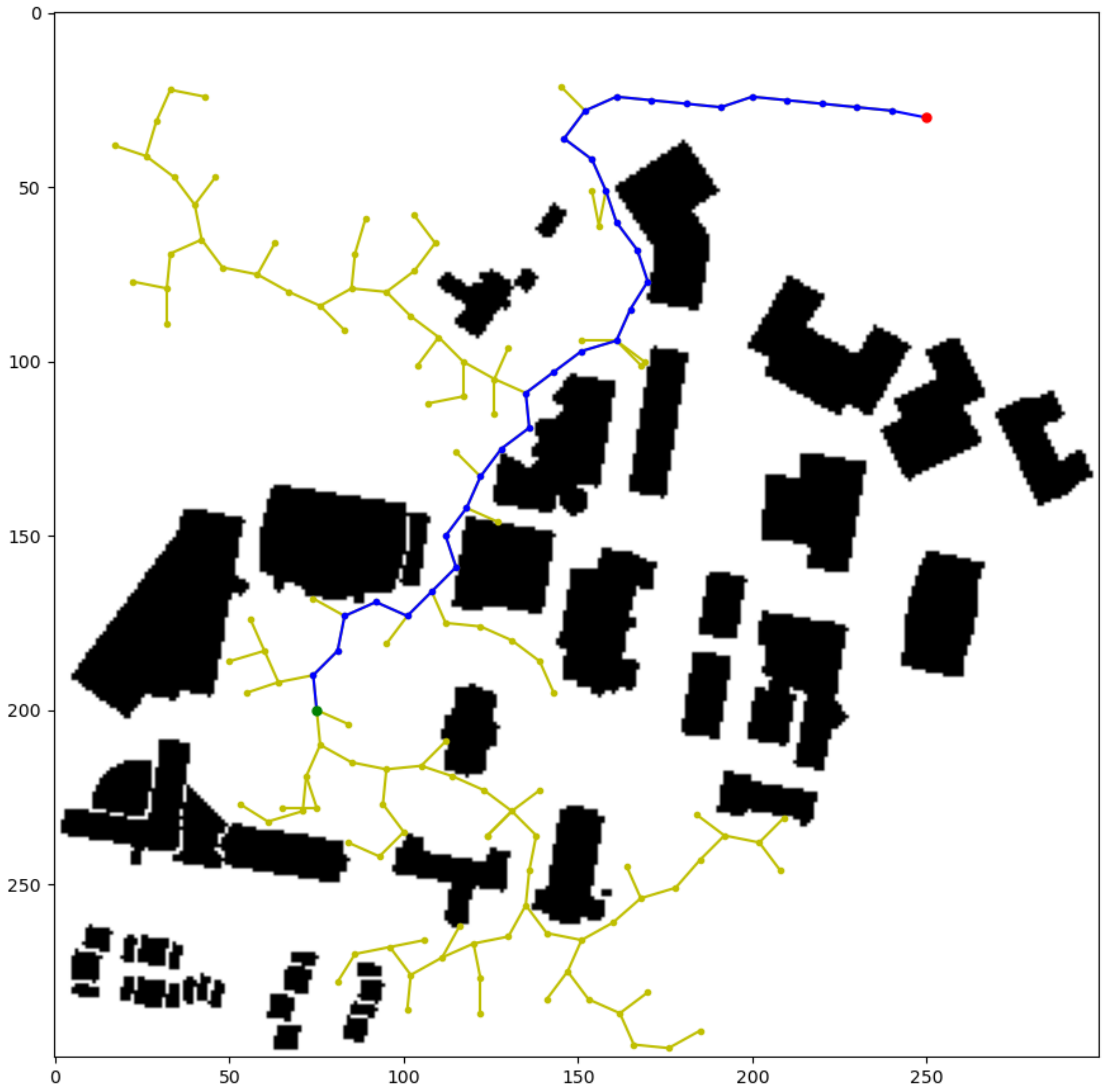
Summary of Experiment 2 –



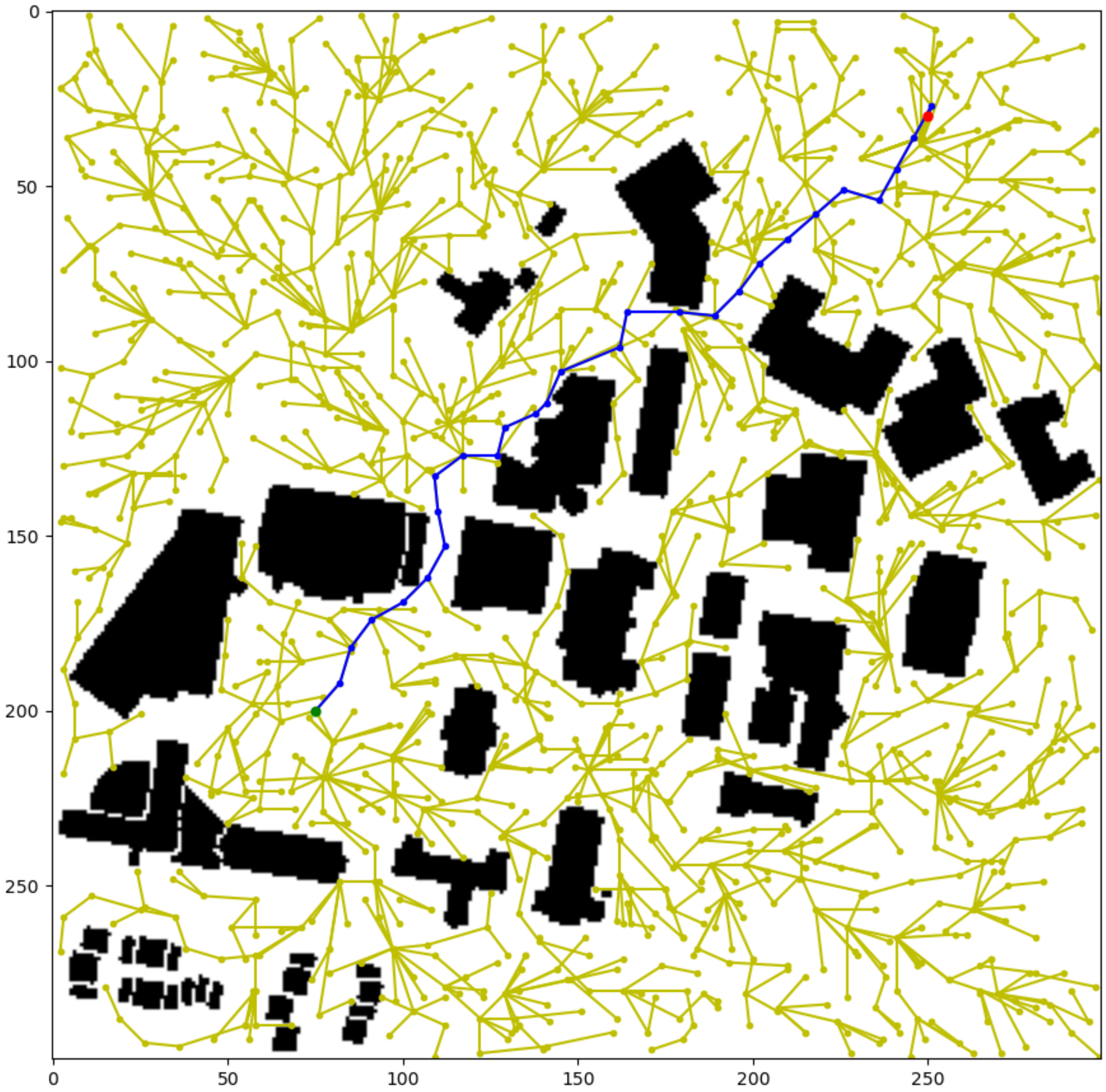
```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion Pl
----------------------------------------------------------------------------
RRT
----------------------------------------------------------------------------
It took 90 nodes to find the current path
The path length is 300.63
----------------------------------------------------------------------------
RRT Star
----------------------------------------------------------------------------
It took 1489 nodes to find the current path
The path length is 288.68
```

We can observe from the Summary of Experiment 1 and Experiment 2 that RRT* tends to produce more and more better path as we increase the sampling point. However, this will always not be the case since the points are sampled randomly, but the probability of getting more optimal path will increase with increase in number of sampling points.

RRT -

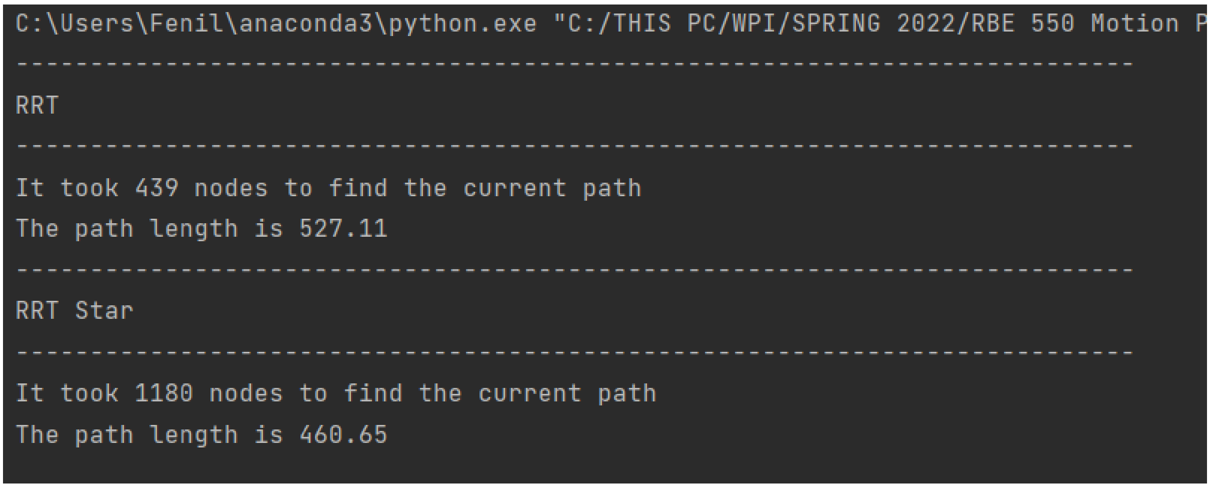


RRT* -



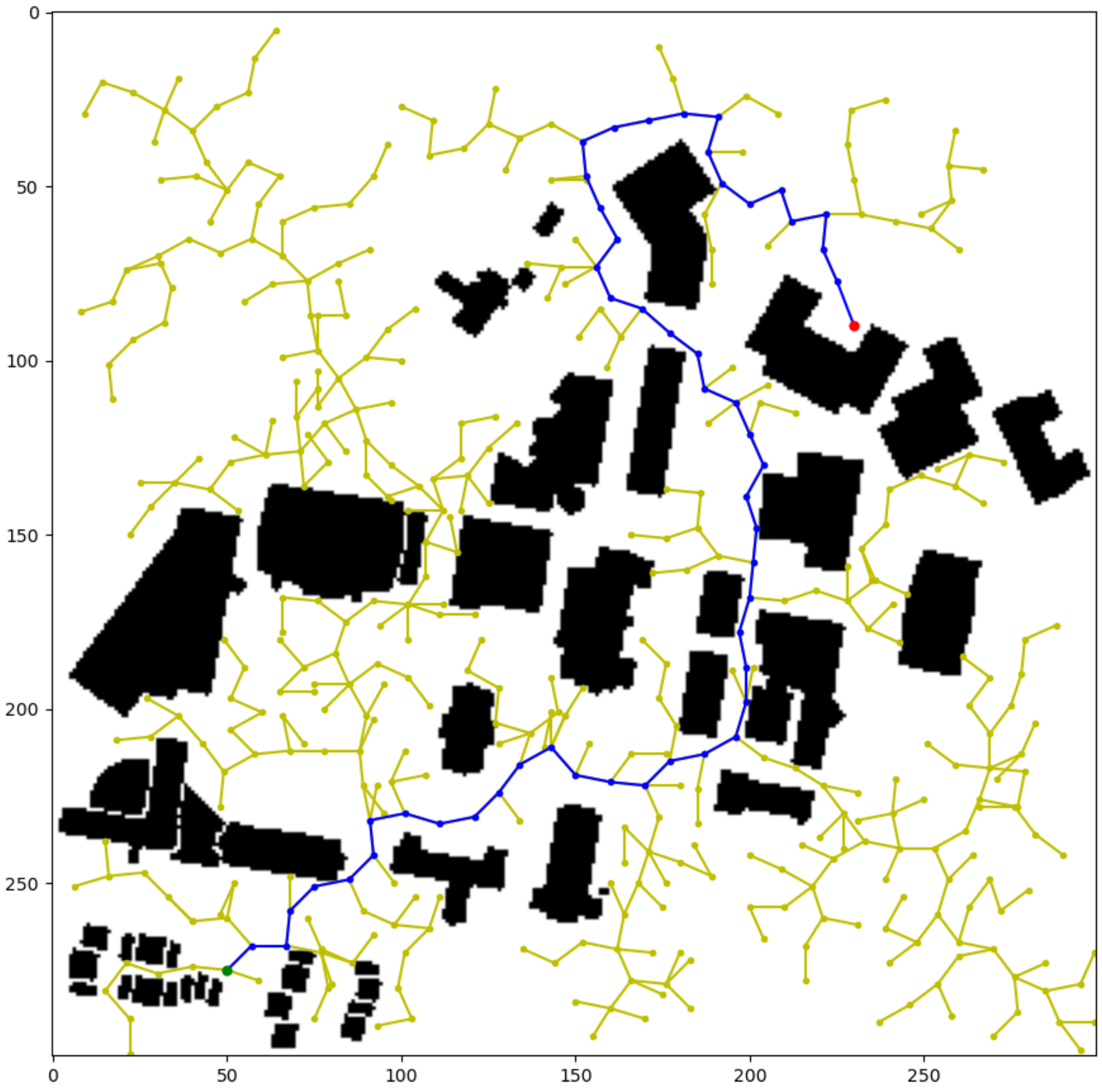
- Experiment 3 –
  I carried out experiment after changing the Goal pose to (275,50) and the Goal Pose to (90,230). Under such critical conditions RRT and RRT* algorithm performs well. Still the RRT* algorithm performs shorter paths as compered to RRT.

Summary of Experiment 3 –

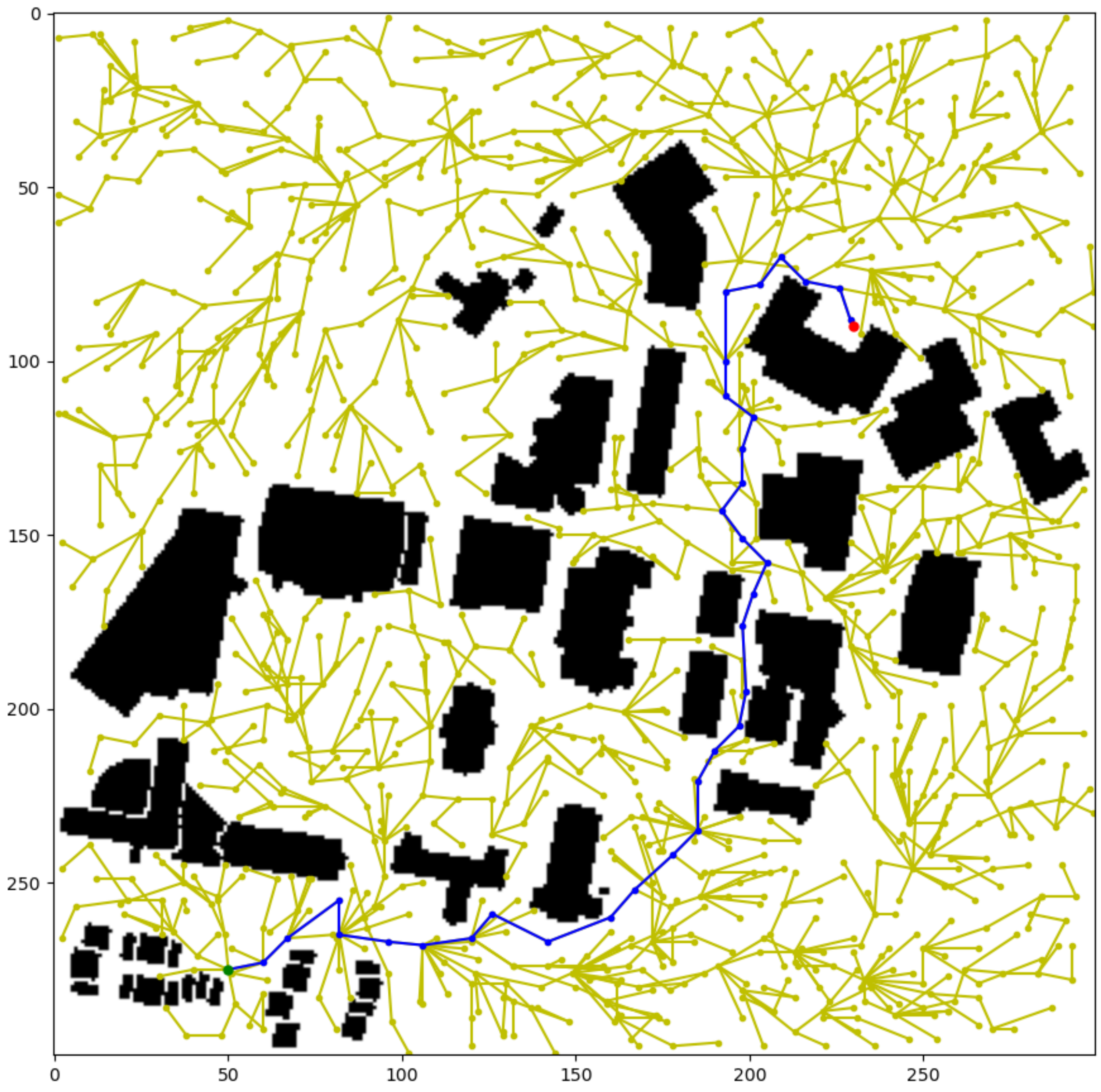


```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion P
---------------------------------------------------------------------------
RRT
---------------------------------------------------------------------------
It took 439 nodes to find the current path
The path length is 527.11
---------------------------------------------------------------------------
RRT Star
---------------------------------------------------------------------------
It took 1180 nodes to find the current path
The path length is 460.65
```

RRT –

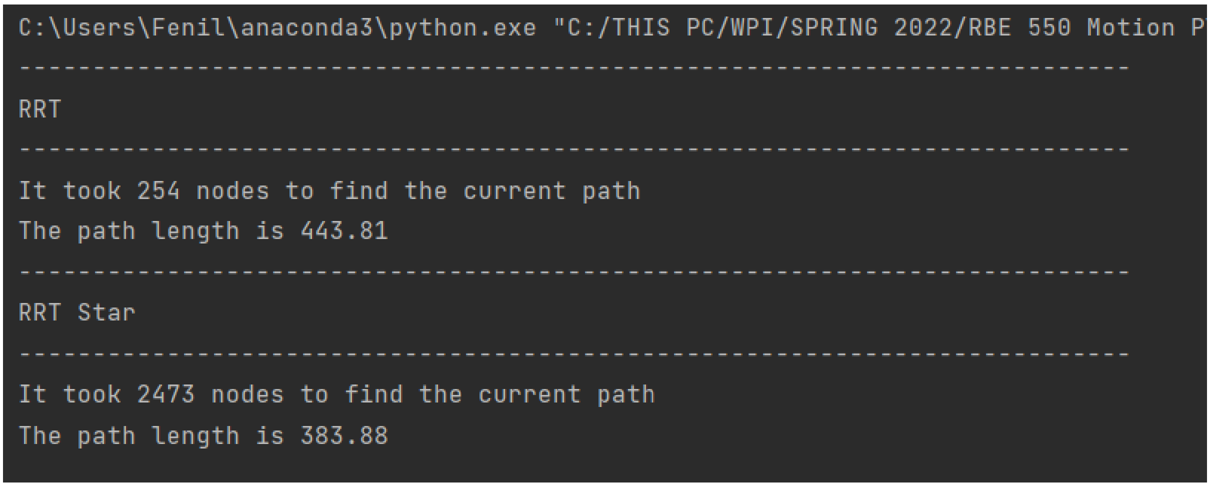


RRT* -



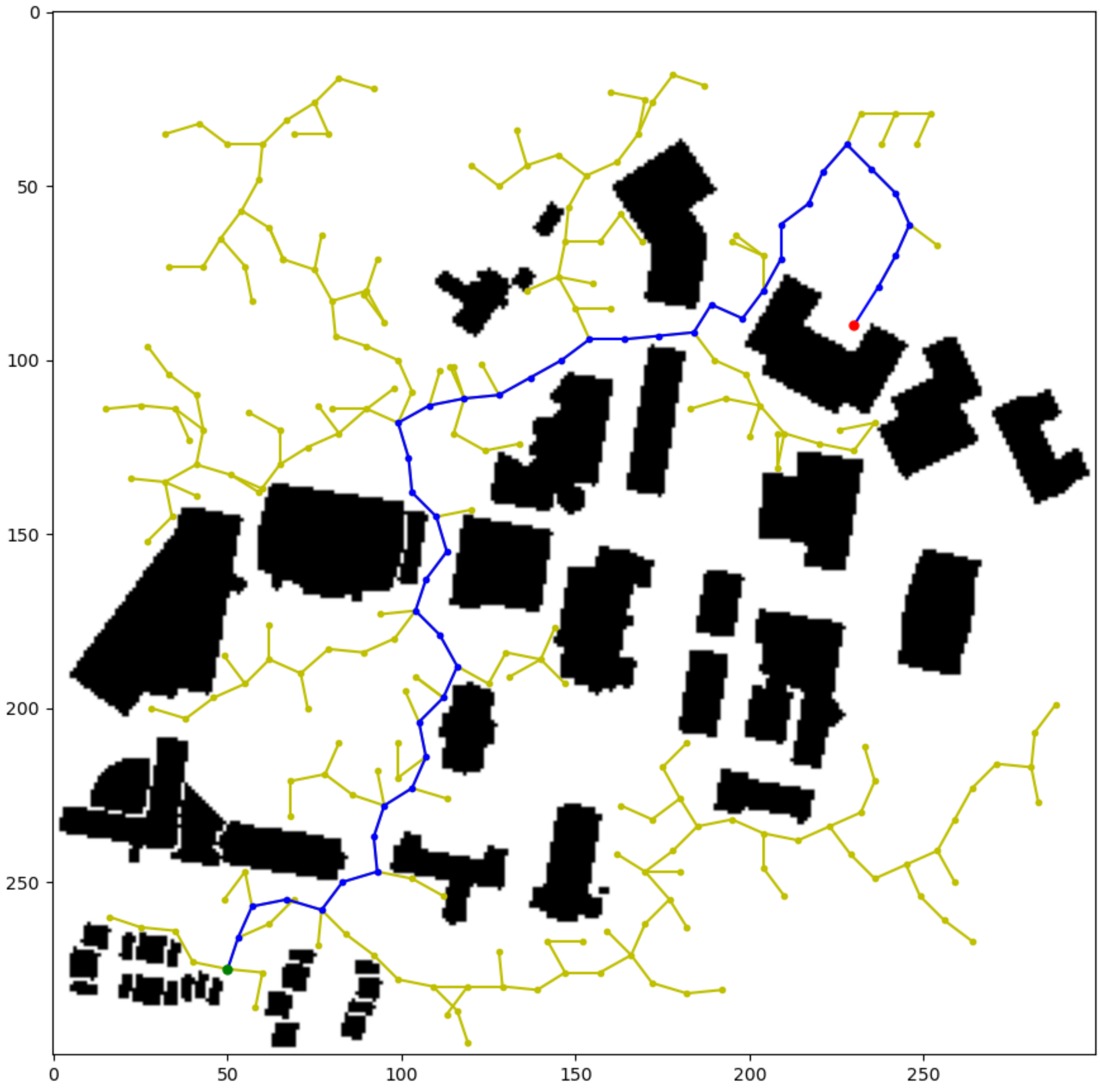
- Experiment 4 –
  I tried the simulation with same start and goal pose as experiment 3 but sampled more points for both RRT and RRT*.
  Number of sampling points for RRT = 2000
  Number of sampling points for RRT* = 4000

Summary of Experiment 4 –

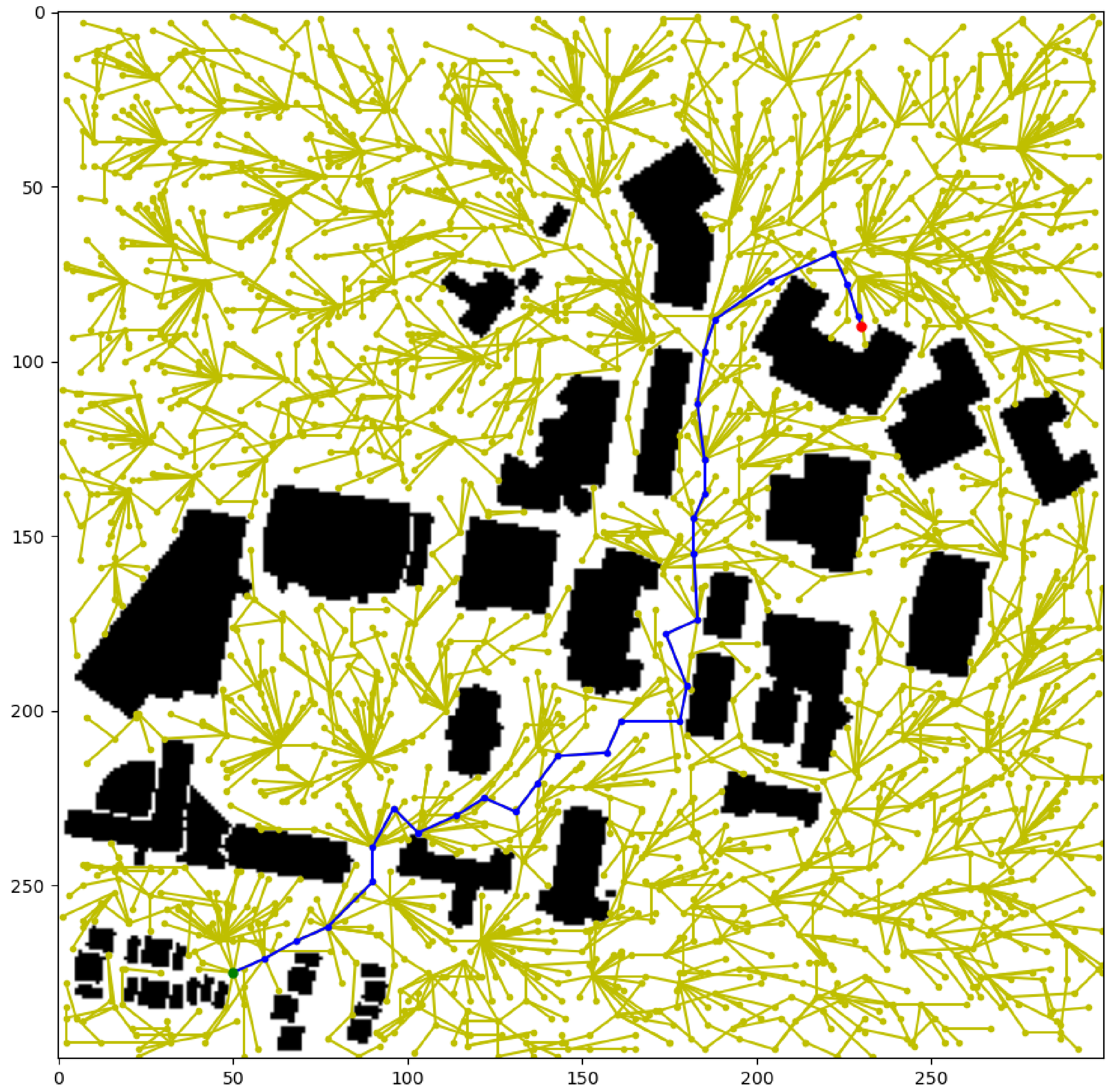


```
C:\Users\Fenil\anaconda3\python.exe "C:/THIS PC/WPI/SPRING 2022/RBE 550 Motion P
--------------------------------------------------------------------------
RRT
--------------------------------------------------------------------------
It took 254 nodes to find the current path
The path length is 443.81
--------------------------------------------------------------------------
RRT Star
--------------------------------------------------------------------------
It took 2473 nodes to find the current path
The path length is 383.88
```

RRT –



RRT* -



We can observe from the Summary of Experiment-3 and Experiment-4 that RRT* tends to produce more and more better path as we increase the sampling point. However, this will always not be the case since the points are sampled randomly, but the probability of getting more optimal path will increase with increase in number of sampling points.

References and Resources:

1. Sampling Based Planning Lecture material by Prof. Zhi Jane Li (Worcester Polytechnic Institute)
2. Template Code for the Programing Assignment 2

END OF REPORT