**Explain in your own words, how does D* replan a path by updating the cost?**

D* is a path planning algorithm that is used in situations where the environment is dynamic and changes frequently. To cope with changes in the environment, D* continually updates its path plan by revising the cost of the nodes in its graph.

When D* is first run, it creates a map of the environment and calculates the optimal path from the start node to the goal node. However, as the environment changes, the cost of traversing each node may change as well. For example, a previously clear path may become blocked, or a previously blocked path may become open.

To handle these changes, D* uses a process called "re-planning". Re-planning involves updating the cost of the nodes in the graph and recalculating the path from the start node to the goal node. The cost of a node is determined by the sum of the cost to reach the node from the start node and the estimated cost to reach the goal node from that node.

When the environment changes, D* uses information from its previous path plan to determine which nodes have been affected and updates the cost of those nodes accordingly. It then recalculates the path from the start node to the goal node using the updated costs. This process continues until a stable path is found that takes into account all the changes in the environment.

In summary, D* replans a path by updating the cost of nodes in its graph and recalculating the path from the start node to the goal node using the updated costs. This process is repeated until a stable path is found that takes into account all changes in the environment.

- Why does D* can replan faster than A* or Dijkstra?

D* can replan faster than A* or Dijkstra in certain situations because it only updates the parts of the graph that have been affected by changes in the environment, rather than re-examining the entire graph.

A* and Dijkstra algorithms require the entire graph to be re-evaluated whenever there is a change in the environment, even if only a small portion of the graph is affected. This can be time-consuming and computationally expensive, particularly in large and complex environments.

D*, on the other hand, only needs to update the affected parts of the graph and recalculate the path from the start node to the goal node using the updated costs. This can be much faster than recalculating the entire graph, especially if the changes in the environment are localized and do not affect the overall path.

Additionally, D* makes use of heuristics to estimate the cost of reaching the goal node from each node in the graph, allowing it to quickly determine which parts of the graph need to be updated. This can further reduce the amount of computation required and speed up the replanning process.

In summary, D* can replan faster than A* or Dijkstra because it only updates the parts of the graph that have been affected by changes in the environment, uses heuristics to estimate the cost of reaching the goal node, and avoids re-examining the entire graph.

- What is the key differences between regular RRT* and informed RRT*?

The key difference between regular RRT* and informed RRT* is that informed RRT* incorporates additional information, such as a heuristic or prior knowledge, to bias the search towards the goal region.

Regular RRT* is a probabilistically complete algorithm that constructs a rapidly-exploring random tree (RRT) by randomly sampling points in the space and adding them to the tree. It then attempts to optimize the path by rewiring the tree to reduce the cost of reaching the goal. However, this process can be inefficient if the search area is large or the goal region is difficult to reach.

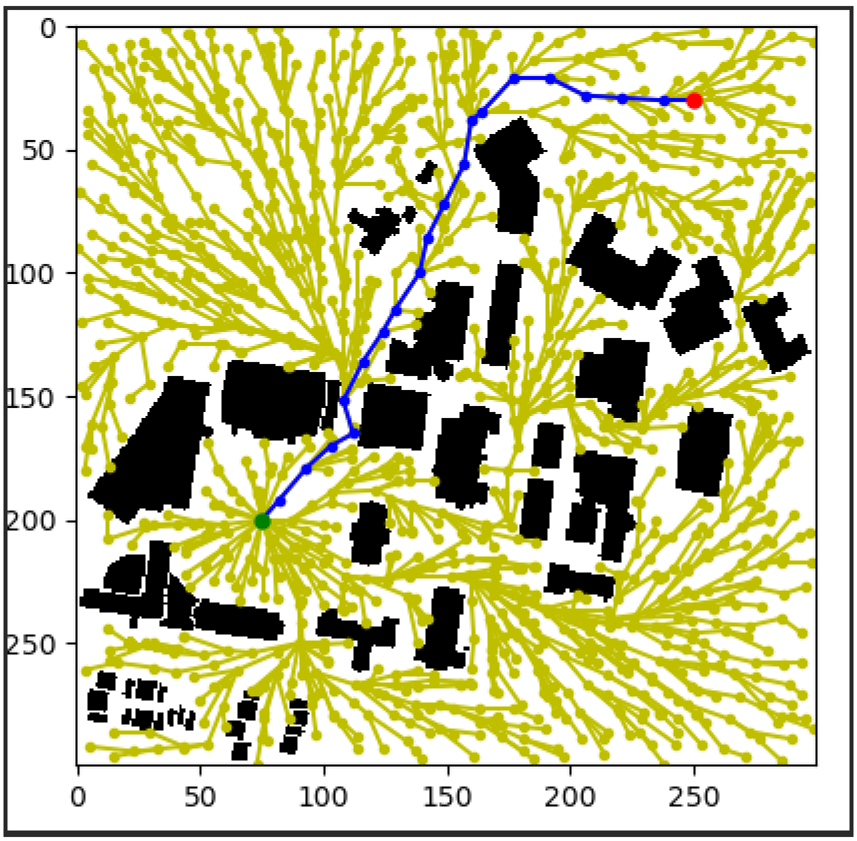
Informed RRT*, on the other hand, incorporates additional information to guide the search towards the goal. This can include a heuristic function that estimates the cost of reaching the goal from a given point, or prior knowledge about the environment or obstacles. This information is used to bias the search towards the goal region, allowing the algorithm to find a solution more efficiently.

One popular approach to informed RRT* is the "RRT* with Informed Sampling" (RRT*-i) algorithm. RRT*-i samples points more heavily in regions that are closer to the goal, as determined by the heuristic function, and less heavily in regions that are farther away. This biases the search towards the goal and can result in faster convergence to a solution.

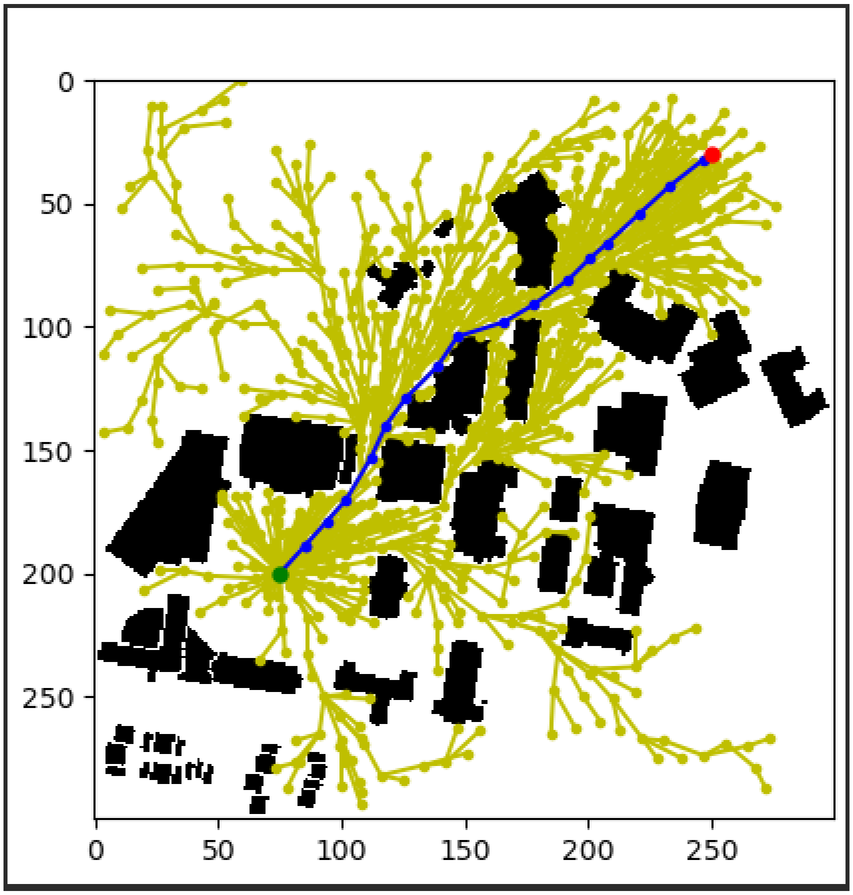
In summary, the key difference between regular RRT* and informed RRT* is that informed RRT* incorporates additional information, such as a heuristic or prior knowledge, to bias the search towards the goal region and find a solution more efficiently.

- By showing and comparing the results of RRT* and informed RRT*, what is the advantages of using the latter?

RRT *




Informed RRT*




From observation we can say that Informed RRT* leads to a shorter or optimal path.

As we can see, informed RRT* has focused its search on the goal region, resulting in a more direct and efficient path to the goal. This is especially useful in situations where the goal region is difficult to reach or the search space is large.

In summary, the advantage of using informed RRT* over regular RRT* is that it can find a solution more efficiently by biasing the search towards the goal region. This can result in a more direct and efficient path to the goal, especially in situations where the goal region is difficult to reach or the search space is large.