Modifications:

RRT* : Initial sample biasing of 13%, increases up to 30% the closer it gets to goal

PRM: Aborts build graph earlier if there is a path found between start and goal every 1000 samples
==> Higher cost but much less time spent and number of nodes
==> The higher the number of samples the more optimal it

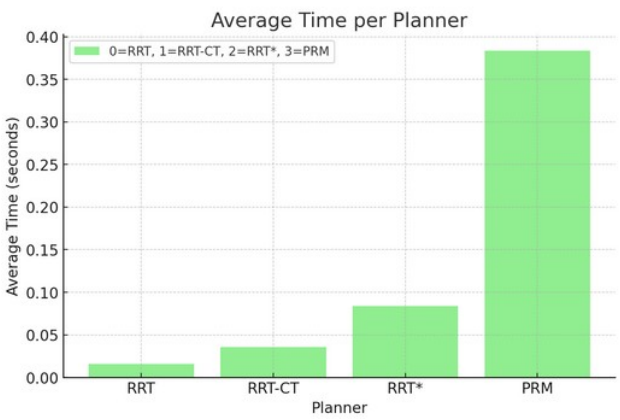
To make it run more traditionally remove

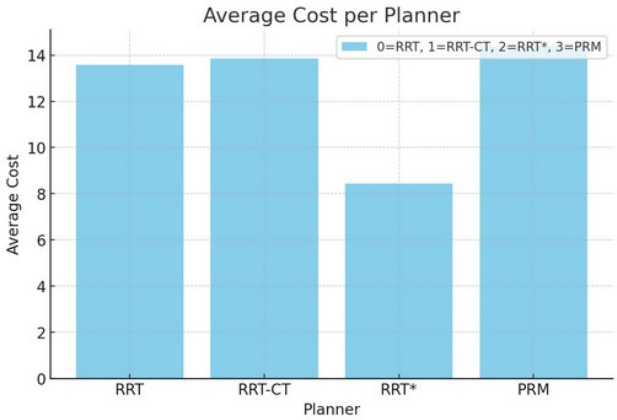
```
        if (k%1000==0){
if(isPathExists(graph, startNode, goalNode)){
printf("path exists\n");
break;
}
```

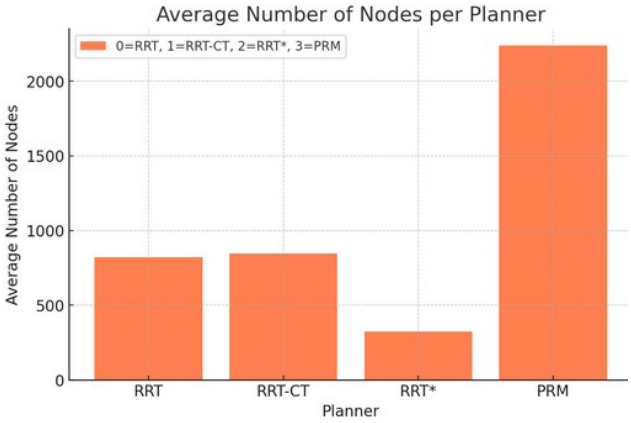
inside of void buildPRM

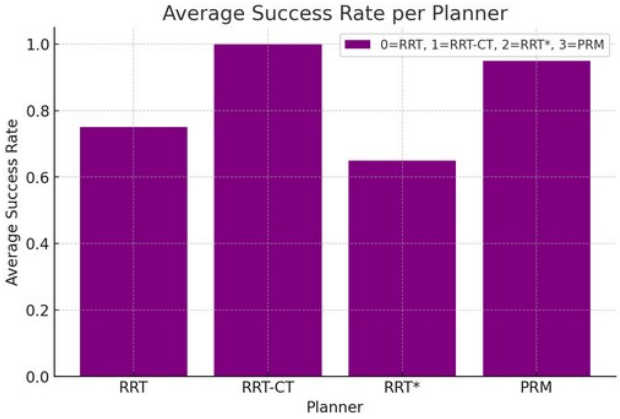
For 4 DoF arm in map2

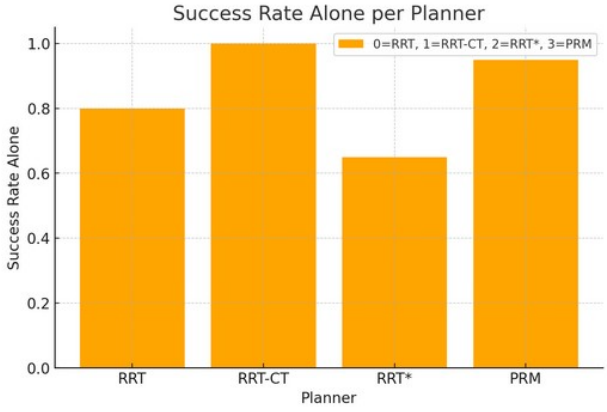
	planner	cost	timespent	Num Nodes	Average Success Rate	Success Rate Alone
1	0	13.580036733333333	0.015668317799645563	821.6	0.75	0.8
2	1	13.861291214	0.03557180190000503	844.8	1.0	1.0
3	2	8.430315538461539	0.08353283376919324	323.46153846153845	0.65	0.65
4	3	14.387903940000001	0.38361067542102417	2238.842105263158	0.95	0.95

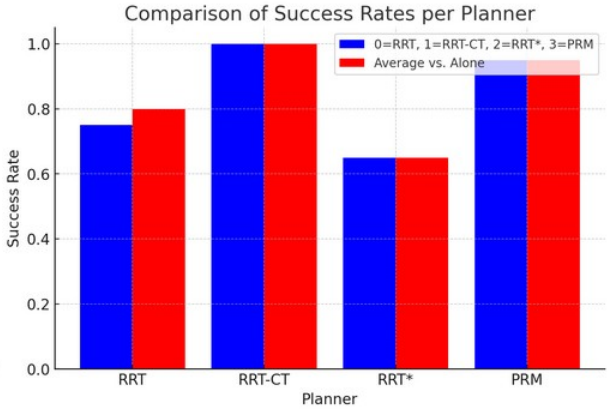








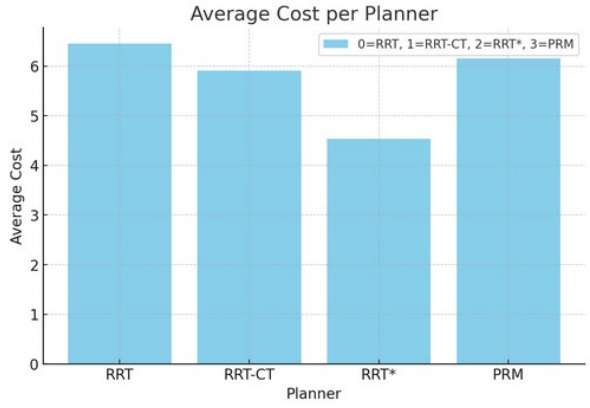


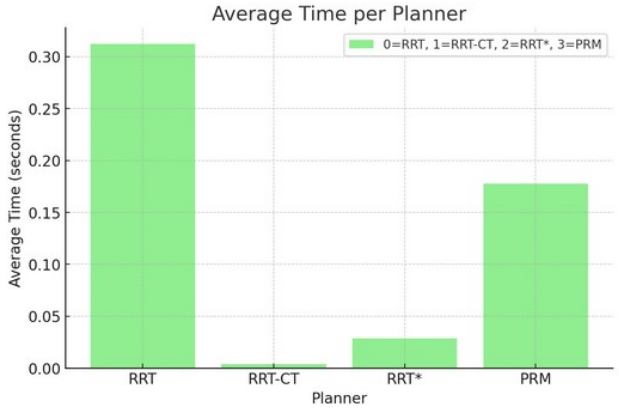


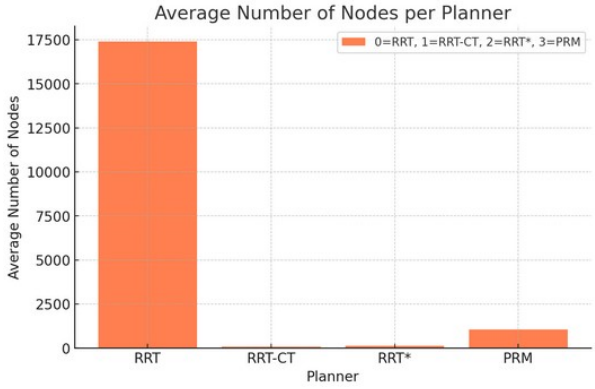
For 3DoF arm in map2

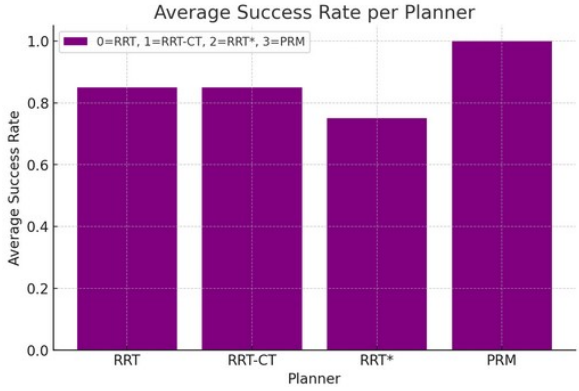
Performs a bit worse since hyperparameters were not changed for comparison purposes

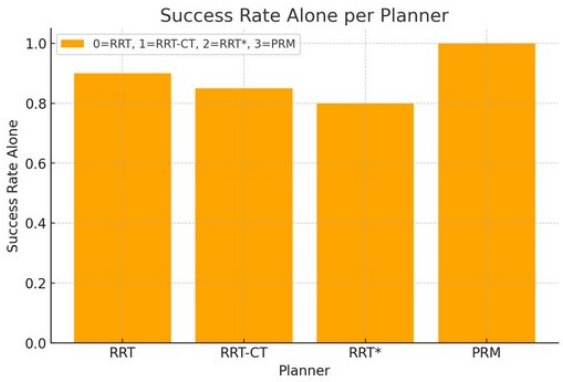
	Algorithm	Average Cost	Average Time	Average Number of No	Average Success Rate	Success Rate Alone
1	0	6.456441251764706	0.3124027161176105 6	17406.647058823528	0.85	0.9
2	1	5.910226623529412	0.0038750522353973534	76.11764705882354	0.85	0.85
3	2	4.533363866666666	0.0287596944668621 2	114.86666666666666	0.75	0.8
4	3	6.153340746	0.1779245219499899	1052.0	1.0	1.0

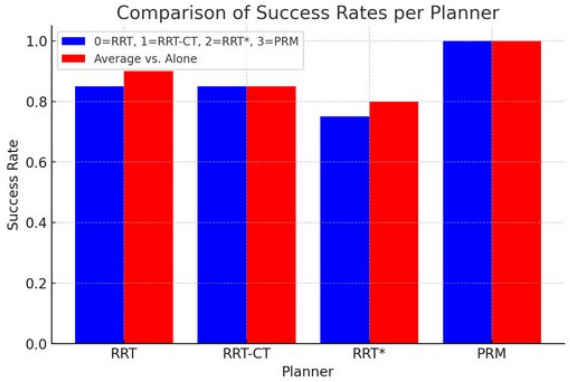












Discussion:
1: The most suitable planner for this environment is RRT connect as it has the fastest runtime while maintaining a high success rate

2: The issue with this planner is that the path cost is suboptimal
Issue RRT: Not very high success rate → requires too many samples to converge
→ sample inefficient
Issue RRT*: High memory overhead and low success rate for higher dimensional problems

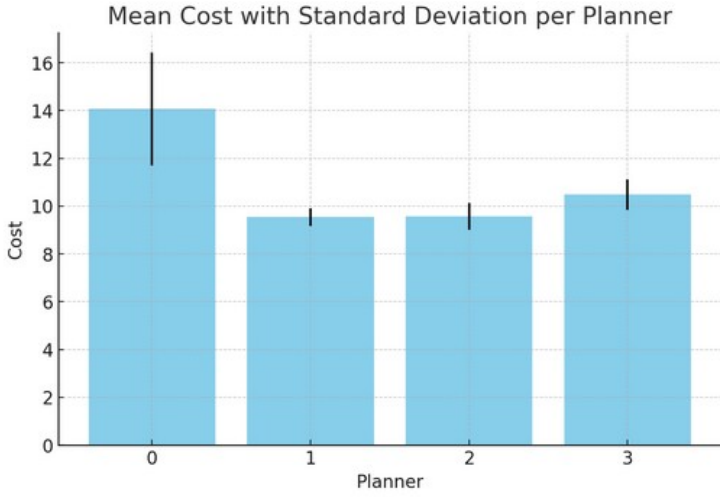
Issue PRM: Slower and uses too many samples

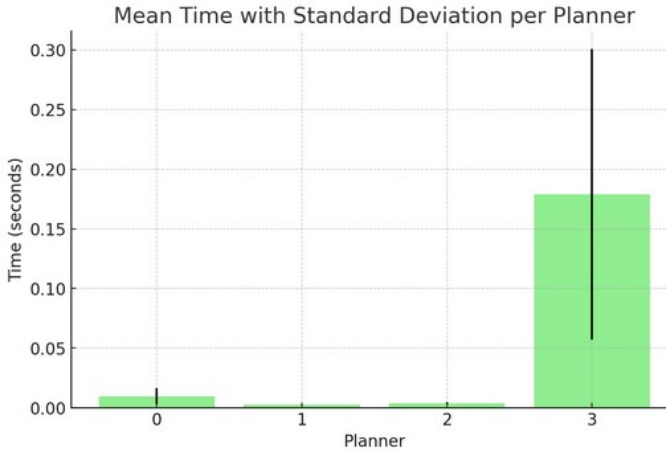
3: RRT connect could be improved by combining concepts of RRT* to enable more rewiring

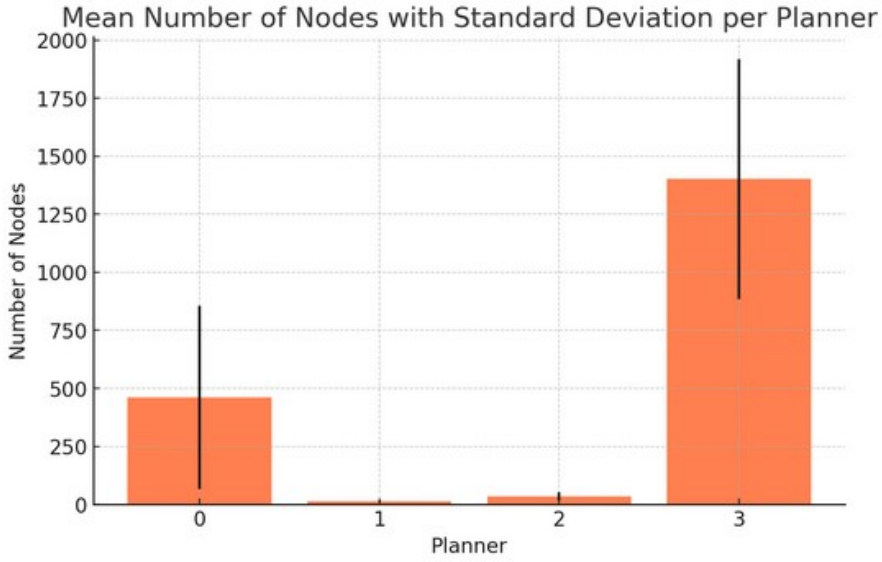
Bonus:
Ran same 4dof sample 10 times in map4 (sample has 100% success rate for all planners)

	Algorithm	Mean Cost	Mean Time	Mean Number of Node
1	0	14.070804	0.00956273289957606	460.5
2	1	9.542855	0.0026362335993325896	12.9
3	2	9.567073	0.0037542732003202897	34.6
4	3	10.4901144	0.1790583691999927	1402.0

	Algorithm	Std Dev Cost	Std Dev Time	Std Dev Number of Nc
1	0	2.3621806561277703	0.006731173182481774	395.1967864241813
2	1	0.3673642521107448	0.00029500389610162607	2.183269719175042
3	2	0.5558543054354352	0.0009618244896000997	18.142644668172156
4	3	0.6381815289649525	0.12180140611823978	516.3977794943223






Mean Number of Nodes with Standard Deviation per Planner
Number of Nodes
2000
1750
1500
1250
1000
750
500
250
0
0
1
2
3
Planner