The analysis report on Cargo Problem.

Below table compares breadth-first, depth-first and depth limit search on Cargo Problem 1, 2 and 3.

Cargo Problem 1:

	Expansions	Goal Tests	New Nodes	Time elapsed (sec)	Number of Steps
Breadth First	43	56	180	0.04	6
Depth First	21	22	84	0.02	20
Depth Limit	101	271	414	0.1	25

Cargo Problem 2:

	Expansions	Goal Tests	New Nodes	Time elapsed (sec)	Number of Steps
Breadth First	3346	4612	30534	17.9	9
Depth First	107	108	959	0.47	37
Depth Limit				Over 10 mins	

Cargo Problem 3:

	Expansions	Goal Tests	New Nodes	Time elapsed (sec)	Number of Steps
Breadth First	14663	18098	129631	145.35	12
Depth First	408	409	3364	2.28	34
Depth Limit				Over 10 mins	

Conclusion:
The running machine is Mac Book Air. Depth limit search can't solve problem 2 and problem 3 in given time(10 mins). Depth first algorithm is the fastest one, in average, it only cost 5% of breadth time to reach the goal. But breadth first algorithm's solution is much better, compared to depth first search. In average, breadth first takes 10.5 step to reach the goal while depth first takes 35 steps.

Below table compares ignore_preconditions and level_sum algorithm on Cargo Problem 1, 2 and 3.

Cargo Problem 1:

	Expansions	Goal Tests	New Nodes	Time elapsed (sec)	Number of Steps
ignore_preconditions	41	43	170	0.04	6
level_sum	11	13	50	0.9	6

Cargo Problem 2:

	Expansions	Goal Tests	New Nodes	Time elapsed (sec)	Number of Steps
ignore_preconditions	1450	1452	13303	4.96	9
level_sum	86	88	841	82	9

Cargo Problem 3:

	Expansions	Goal Tests	New Nodes	Time elapsed (sec)	Number of Steps
ignore_preconditions	5040	5042	44944	21.29	12
level_sum	325	327	3002	127	12

Conclusion:
Both ignore_preconditions and level_sum algorithm can reach the same number of steps to the goal. But ignore_preconditions spend much less time than level_sum, say only 5% of level_sum. Check the data more carefully, the expansions, goal tests and new nodes number on level_sum are great less than those on ignore preconditions. But why spend more time?

Compare between non-heuristic and heuristic algorithm.
If we just want to know if there is a solution or not, depth first search seems the best one.
If we want to know the best steps to reach the goal in a shorted time, seems heuristic algorithm, such as ignore_preconditions are much better.

More depth analysis.
I run level_sum algorithm several times, and the time elapsed quite different: from 127 seconds to 682 seconds. And i analysis the every functions cost time with python -m cProfile. and below give some information:

level_sum algorithm: total time cost 682 seconds.
Top 5 time cost calls:

ncalls	tottime	filename:lineno(function)
126284658/73104582	89.571	utils.py:416(__eq__)

ncalls	tottime	filename:lineno(function)
95261574	53.176	my_planning_graph.py:437(test_mutex)
24054429	52.240	my_planning_graph.py:423(interference_mutex)
118838641	51.985	{built-in method builtins.any}
5675	49.704	my_planning_graph.py:356(update_a_mutex)

ignore_preconditions algorithm, total time cost 30 seconds.
Top 2 calls:

ncalls	Tot time	filename:lineno(function)
25853137/14863845	14.990	utils.py:416(__eq__)
44945	2.714	lp_utils.py:37(encode_state)

Analysis result:
In both algorithm, utils.py:416(__eq__) spends top1 time. Around 50% in ignore_preconditions and 15% of time in level_sum. For level_sum algorithm, not very clear bottle neck in function calls. Maybe it is my laptop (Mac Book Air, 8G memory + 128G disk, and disk space is in short) caused the level_sum algorithm very slow.