For the AIND-Planning project searches were run against Problems 1, 2 and 3. For some of the search runs an optimal plan was found for each problem. These are:

Optimal Plan for Problem 1	Optimal Plan for Problem 2	Optimal Plan for Problem 3
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)	Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Fly(P3, ATL, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO) Unload(C3, P3, SFO)	Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, ATL) Fly(P2, JFK, ORD) Load(C3, P1, ATL) Load(C4, P2, ORD) Fly(P1, ATL, JFK) Fly(P2, ORD, SFO) Unload(C1, P1, JFK) Unload(C3, P1, JFK) Unload(C2, P2, SFO) Unload(C4, P2, SFO)

In addition the performance results of the searches were recorded and are given here:

	Nodes Expanded	Goal Tests	New Nodes	Time (Seconds)	Optimal
Problem 1					
Breadth-first	43	56	180	0.0504152064	Yes
Depth-first graph	12	13	48	0.01175240149	No
Uniform cost	55	57	224	0.05734031839	Yes
A* w/h_ignore_preconditions	41	43	170	0.06675974181	Yes
A* w/h_pg_levelsum	11	13	50	1.983294874	Yes
Problem 2					
Breadth-first	3343	4609	40960	26.05347398	Yes
Depth-first graph	442	443	5296	3.805696883	No
Uniform cost	4853	4855	58964	89.22195782	Yes
A* w/h_ignore_preconditions	1506	1508	18604	29.05679262	Yes
A* w/h_pg_levelsum	86	88	1166	311.3372816	Yes
Problem 3					
Breadth-first	14663	18098	209742	219.2158862	Yes
Depth-first graph	9479	9480	124029	251.2492998	No
Uniform cost	18236	18238	256119	985.3947493	Yes
A* w/h_ignore_preconditions	5118	5120	74991	229.2944703	Yes
A* w/h_pg_levelsum	404	406	6306	3627.209764	Yes

ANALYSIS

All searches were allowed to run past the 10 minute (i.e., 600 second) maximum recommended time limit. A couple in Problem 3 exceeded that limit substantially bud did complete and so are recorded here.

For uninformed searches breadth-first search consistently outperformed both depth-first graph and uniform cost searches. Per the classes "Search" video quizzes, bread-first and cheapest-cost searches are complete and optimal whereas depth-first search is not. All of the breadth-first searches performed returned optimal plans. Also, one advantage that the breadth-first implementation has it that it performs a goal test check on the next frontier node before expanding its children whereas this is done after expansion for best-first. This difference is not significant for Problem 1 but for Problems 2 and 3 becomes quite apparent in both nodes expanded and time elapsed. The uniform-cost searches were special cases of cheapest-cost searches (using the node's path cost) and always returned optimal plans. The depth-first graph searches produced faster times and fewer nodes and goal tests than other uninformed searches but, as expected, never found an optimal plan.

Ignoring preconditions is a better performing heuristic than (the inadmissible heuristic) level sum for the A* searches. The level sum heuristic also requires the construction of a complete PlanningGraph instance each time and this overhead results in slower runtimes even though the "ignore preconditions" heuristics causes many more expanded nodes and goal tests.

The best performance among all the search choices was close between the breadth-first and A* with the ‘ignore preconditions” heuristic. The first was always about 10% faster than the latter though the A* choice made fewer goal tests and node expansions than breadth-first (especially in Problems 2 and 3). The package delivery problem is not fully decomposable and so the A* search could not take advantage of that. This heuristic is not admissible and, in addition, the relaxed problem of ignoring preconditions generates some states which could be valid plans but are not and so some time is spent handling those cases.