1. Problem 1

a) non-heuristic search

	BFS	Depth_first_graph	Uniform_cost	Greedy_best_first_graph
Expansion	43	21	55	7
Goal tests	56	22	57	9
New nodes	180	84	224	28
Time_elapsed	0.042 (s)	0.018	0.050	0.006
Plan length	6	20	6	6
Optimality	yes	no	yes	Yes. best

b) heuristic search

	H_ignore_preconditions	H_pg_levelsum
Expansion	41	11
Goal tests	43	13
New nodes	170	50
Time_elapsed	0.039 (s)	0.079
Plan length	6	6
Optimality	Yes.	Yes. Slightly better in expansion.

The optimal actions are:

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

2. Problem 2

a) non-heuristic search

	BFS	Depth_first_graph	Uniform_cost	Greedy_best_first_graph
Expansion	3343	624	4853	998
Goal tests	4609	625	4855	1000
New nodes	30509	5602	44041	8972
Time_elapsed	24.7 (s)	6.10	24.5	4.8
Plan length	9	619	9	15
Optimality	yes	no	yes	no

b) heuristic search

	H_ignore_preconditions	H_pg_levelsum
Expansion	1450	86
Goal tests	1452	88
New nodes	13303	841
Time_elapsed	7.44 (s)	68.69
Plan length	9	9
Optimality	Better in search time	Better in expansion

The optimal actions are:

Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)

2. Problem 3

a) non-heuristic search

	Breadth_first	Depth_first_graph	Uniform_cost
Expansion	14663	408	18223
Goal tests	18098	409	18225
New nodes	129631	3364	159618
Time_elapsed	186 (s)	3.5	119
Plan length	12	392	12
Optimality	yes	no	yes

b) heuristic search

	H_ignore_preconditions	H_pg_levelsum
Expansion	5040	313
Goal tests	5042	315
New nodes	44944	2884
Time_elapsed	34 (s)	325
Plan length	12	12
Optimality	Better in search time	Better in expansion

The optimal actions are:

Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C4, P2, SFO)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Summary:

a) Non-heuristic search has comparable (sometimes even better) performance in problem 1 as heuristic search – the reason is that problem 1 is very simple so brute force method works well. Another thing to note is that depth-first search could not guarantee optimal solution – depth-first search may look for longer path first and is not optimal as explained in the video in lesson 8.23 "search comparison". In more complex problems 2 & 3, heuristic search methods are much better in both optimality and expansion of search.

b) For the two heuristic searches, ignore_preconditions heuristic usually searches with larger expansion of nodes, and still achieves the goal with much less time. The reason is that it does not need to create the planning graph and the heuristic is much cheaper to compute, as discussed in Russel&Norvig 10.2.3. The planning graph with level sum heuristic has the advantage of smaller expansion, but uses longer time due to computation of creating planning graph and level sums. As the problem gets more complicated – with more planes/airports/cargos, the planning graph with level sum will probably be a better choice due to its better expansion performance and the overhead of creating planning graph may be compensated.