Written Analysis

- **Provide an optimal plan for Problems 1, 2, and 3.**

Problem1
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)

Problem 2
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(C3, P3, SFO)
Unload(C2, P2, SFO)

Problem 3
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)

Table 1

		Breadth First Search	Breadth First Tree Search	Depth First Graph Search	Depth Limited Search	Uniform Cost Search
Problem 1	Expansions	43	1458	21	101	55
	Goal Tests	56	1459	22	271	57
	New Nodes	180	5960	84	414	224
	Plan Length	6	6	20	50	6
	Time	0.0334	0.7812	0.0116	0.0749	0.0306
Problem 2	Expansions	3346	>10min	107	>10min	4853
	Goal Tests	4612		108		4855
	New Nodes	30534		959		44041
	Plan Length	9		105		9
	Time	13.2240		0.2904		11.5000
Problem 3	Expansions	14120	>10min	292	>10min	18223
	Goal Tests	17673		293		18225
	New Nodes	124926		2388		159618
	Plan Length	12		288		12
	Time	92.4039		1.0868		49.5096

- **Compare and contrast non-heuristic search result metrics (optimality, time elapsed, number of node expansions) for Problems 1,2, and 3.**

For Problem 1, both the time elapsed and the number of node expansions of either *Breadth First Tree Search* or *Depth Limited Search* is much larger than the others. Therefore, they are the worst two of them. And *Depth First Graph Search* is the optimality. While others are almost the same.

For Problem 2, the time elapsed of *Depth Limited Search* and *Breadth First Tree Search* is more than 10 minutes. Therefore, they are the worst two of them. And *Depth First Graph Search* is the optimality. While others are almost the same.

For Problem 3, the time elapsed of *Depth Limited Search* and *Breadth First Tree Search* is more than 10 minutes. Therefore, they are the worst two of them. And *Depth First Graph Search* is the optimality.

We can see that both these algorithms are not so stable when facing problems with various scales except *Depth First Graph Search*. When the number of inputs of the problem increases a little, the expansion nodes and time elapsed may increase greatly.

- ## Compare and contrast heuristic search result metrics using A* with the “ignore preconditions” and “level-sum” heuristics for Problems 1, 2, and 3.

For Problem 1, both the time elapsed and the number of node expansions of *Recursive Best First Search with h_1* is much larger than the others. Therefore, it is the worst one of them. And *A* with h_ignore_preconditions* is the optimality. While others are almost the same.

Table 2

		Recursive Best First Search with h_1	Greedy Best First Graph Search with h_1	A* Search with h_1	A* with h_ignore_preconditions	A* with h_pg_levelsum
Problem 1	**Expansions**	4229	7	55	41	11
	Goal Tests	4230	9	57	43	13
	New Nodes	17023	28	224	170	50
	Plan Length	6	6	6	6	6
	Time	2.5325	0.0057	0.0381	0.0273	0.6611
Problem 2	**Expansions**	>10min	998	4853	1450	86
	Goal Tests		1000	4855	1452	88
	New Nodes		8982	44041	13303	841
	Plan Length		21	9	9	9
	Time		2.2745	11.4939	3.5775	65.2941
Problem 3	**Expansions**	>10min	5578	18223	5040	316
	Goal Tests		5580	18225	5042	318
	New Nodes		49150	159618	44944	2912
	Plan Length		22	12	12	12
	Time		14.7901	49.4050	13.8296	294.3089

For Problem 2, the time elapsed of *Greedy Best First Graph Search with h_1* is more than 10 minutes. Therefore, it is the worst one of them. And *A* with h_ignore_preconditions* is the optimality. While others are almost the same.

For Problem 3, the time elapsed of *Recursive Best First Search with h_1* is more than 10 minutes. Therefore, it is the worst one of them. And *A* with h_ignore_preconditions* is the optimality. While others are almost the same.

A with h_pg_levelsum* takes more time but search less nodes. Among 3 A* algorithms, *A* with h_ignore_preconditions* is the most stable one.

• Best heuristic used in these problems

In both three problem, *h_ignore_preconditions* spends least time and expand less nodes than *h_1*, that means it can find the optimal plan easier. So I think it is the best one.

It is not better than non-heuristic search planning methods for all problems for the following reasons.
Compared with the best non-heuristic search planning methods — *Depth First Graph Search*, the time elapsed and the number of expansion nodes is more for all problems.