Analyze the search complexity as a function of domain size, search algorithm, and heuristic

- The following table includes data for all search & heuristic combinations for air cargo problems 1 and 2:

Air Cargo Problem	Actions	Search Function	New Nodes
1	20	1	178
1	20	2	84
1	20	3	240
1	20	4	29
1	20	5	28
1	20	6	24
1	20	7	28
1	20	8	206
1	20	9	122
1	20	10	180
1	20	11	138
2	72	1	30503
2	72	2	5602
2	72	3	46618
2	72	4	170
2	72	5	86
2	72	6	249
2	72	7	84
2	72	8	22522
2	72	9	3426
2	72	10	26594
2	72	11	9605

- The following table includes data for one uniformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4:

Air Cargo Problem	Actions	Search Function	New Nodes
3	88	1	129625
3	88	4	230
3	88	5	126
3	88	8	65711
3	88	9	3403
4	104	1	944130
4	104	4	280

Air Cargo Problem	Actions	Search Function	New Nodes
4	104	5	165
4	104	8	328509
4	104	9	12210

- Regarding the growth trends as the problem size increases, even trivial planning problems become intractable for domain-independent planning. The search space for planning problems grows exponentially with problem size. Refer to the pseudocode, the function chooses a leaf node for expansion according to strategy by calling extend function, which creates and returns a set of new nodes, so the number of New Nodes is bigger than Expansions.

Analyze the search time as a function of domain size, search algorithm, and heuristic

- The following table includes data for all search & heuristic combinations for air cargo problems 1 and 2:

Air Cargo Problem	Actions	Search Function	Search Time
1	20	1	0.005738312
1	20	2	0.003031437
1	20	3	0.008646773
1	20	4	0.001422161
1	20	5	0.430203578
1	20	6	0.123274573
1	20	7	1.154152003
1	20	8	0.008778578
1	20	9	1.096785234
1	20	10	0.446812224
1	20	11	3.002503949
2	72	1	1.793659079
2	72	2	2.714538505
2	72	3	3.067347889
2	72	4	0.017521590
2	72	5	10.322775617
2	72	6	5.927637818
2	72	7	25.886673475
2	72	8	2.055049125
2	72	9	260.652598607
2	72	10	602.945406510
2	72	11	1962.084911764

- The following table includes data for one uniformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4:

Air Cargo Problem	Actions	Search Function	Search Time
3	88	1	9.726209561
3	88	4	0.034026403
3	88	5	23.441392788
3	88	8	7.875705338
3	88	9	426.632465442
4	104	1	88.426900362
4	104	4	0.055148840

Air Cargo Problem	Actions	Search Function	Search Time
4	104	5	42.506554062
4	104	8	51.476212157
4	104	9	2389.225633150

- Regarding the growth trends as the problem size increases, the search time for planning problems grows with problem size. As an informed search, A* algorithm uses problem-specific knowledge to find solutions much more efficiently than uninformed search. Efficiency can be further improved on implementing heuristics by incrementally growing the graph rather than building until it levels off. For example, to calculate the max level heuristic for the planning graph, the graph can be expanded one level at a time until the last goal is met rather than filling the whole graph at the very beginning. This alternate implementation is at least 2x faster than the simple one.

Analyze the optimality of solution as a function of domain size, search algorithm, and heuristic

- The following table includes data for all search & heuristic combinations for air cargo problems 1 and 2:

Air Cargo Problem	Actions	Search Function	Plan Length
1	20	1	6
1	20	2	20
1	20	3	6
1	20	4	6
1	20	5	6
1	20	6	6
1	20	7	6
1	20	8	6
1	20	9	6
1	20	10	6
1	20	11	6
2	72	1	9
2	72	2	619
2	72	3	9
2	72	4	9
2	72	5	9
2	72	6	9
2	72	7	9
2	72	8	9
2	72	9	9
2	72	10	9
2	72	11	9

- The following table includes data for one uniformed search, two heuristics with greedy best first search, and two heuristics with A* on air cargo problems 3 and 4:

Air Cargo Problem	Actions	Search Function	Plan Length
3	88	1	12
3	88	4	15
3	88	5	14
3	88	8	12
3	88	9	12
4	104	1	14
4	104	4	18

Air Cargo Problem	Actions	Search Function	Plan Length
4	104	5	17
4	104	8	14

Answer all required questions

- Greedy Best First Search would be most appropriate for planning in a very restricted domain and needs to operate in real time, because of its minumum number of expansions and new nodes.
- A* algorithm would be most appropriate for planning in a very large domains, for example, planning delivery routes for all UPS drivers in the U.S. on a given day. A* algorithm is faster because it's an informed search.
- Breadth First Search and Uniform Cost Search would be most appropriate for planning problems where it is important to find only optimal plans, because of the minimum length of the plans.