Planning result analysis

- **Uninformed (non-heuristic) search methods**:

Problem 1

Search Method	Optimality	Plan length	Time elapsed(s)	Number of node expansions
Breadth First Search	Yes	6	0.04	43
Depth first graph search	NO	12	0.01	12
Uniform cost search	YES	6	0.04	55

Problem 2

Search Method	Optimality	Plan length	Time elapsed(s)	Number of node expansions
Breadth First Search	YES	9	11.11	3343
Depth first graph search	NO	1444	16.8	1669
Uniform cost search	YES	9	15.78	4853

Problem 3

Search Method	Optimality	Plan length	Time elapsed(s)	Number of node expansions
Breadth First Search	YES	12	56.15	14663
Depth first graph search	NO	571	3.94	592
Uniform cost search	YES	12	62.25	18164

Algorithm performance can be evaluated in four ways [1]: Completeness, Optimality, Time complexity and Space complexity. The table below from the AIMA book shows the comparison of different Un-informed search methods using the above four categories.

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes^a	Yes^{a,b}	No	No	Yes^a	Yes^{a,d}
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes^c	Yes	No	No	Yes^c	Yes^{c,d}

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: ^{a} complete if b is finite; ^{b} complete if step costs $\geq \epsilon$ for positive ϵ; ^{c} optimal if step costs are all identical; ^{d} if both directions use breadth-first search.

Based on the above table[2], two of the three uninformed search methods used for Problem 1,2,3, Breadth-First and Uniform-Cost, fulfil optimality condition and Depth-First is not. In the heuristic comparison table above, we can see that for all three problems Breadth-First and Uniform-Cost methods arrive at optimal plan of 6, 9, and 12 for Problems 1,2, and 3 respectively. Depth-Frist graph search does not arrive at optimal plan for any of the three problems. Based on just optimality condition, Breadth-First or Uniform-Cost search methods would is well suited in our case. Even though Depth-first method does not arrive at optimal plan, it does work well in-term of space complexity, given by number of node expansions as the depth-first search only requires storage of *O(bm)* nodes as shown in table 3.21. The time complexity for Uniform cost search is higher than Breadth-first search for problem 2 and 3, given by increase in state space complexity and execution time. This is because uniform-cost search can explore large trees of small steps before exploring paths involving large and useful steps[3]. When looking at all four evaluation categories **Breadth-first search** method performs well compared to other Depth-first and uniform-cost search method, hence would be a better choice for all three problems.

- **Informed (Heuristic) search methods:**

Problem 1

Search Method	Optimality	Plan length	Time elapsed(s)	Number of node expansions
A* h_1	YES	6	0.046	55
A* h_ignore_preconditions	YES	6	0.043	41
A* h_pg_levelsum	YES	6	3.74	60

Problem 2

Search Method	Optimality	Plan length	Time elapsed(s)	Number of node expansions
A* h_1	YES	9	15.41	4853
A* h_ignore_preconditions	YES	9	5.77	1450
A* h_pg_levelsum	YES	9	1669.73	5716

Problem 3

Search Method	Optimality	Plan length	Time elapsed(s)	Number of node expansions
A* h_1	YES	12	54.47	18164
A* h_ignore_preconditions	YES	12	17.18	5038
A* h_pg_levelsum	--	--	--	--

The above table gives performance metrics for Informed search methods for three problems. Based on Optimality, Time-complexity and Space-complexity, **A* h_ignore_preconditions** method is best choice among the three A* heuristic search methods. A* h_pg_levelsum metrics were not included for Problem 3 as the method did not return complete solution within 10m.

A* h_ignore_preconditions heuristic method performance is comparable to Breadth-First and Unifrom-Cost search method for Problem 1, but A* h_ignore_preconditions performs better for Problem 2 and 3 across all methods explore in this analysis. A* h_ignore_preconditions is heuristic from a relaxed problem, where we ignore preconditions and have fewer restrictions on the actions. Relaxed problem adds edges to the state space and may arrive at better solutions if the added edges provide shortcuts to the goal[4].

References:

1. "Artificial Intelligence: A Modern Approach" 3rd edition Chapter 3, section 3.3.2, pg 80.
2. "Artificial Intelligence: A Modern Approach" 3rd edition Chapter 3, section 3.4.7, pg.91.
3. "Artificial Intelligence: A Modern Approach" 3rd edition Chapter 3, section 3.4.2, pg.84.
4. "Artificial Intelligence: A Modern Approach" 3rd edition Chapter 3, section 3.6.2, pg.105.