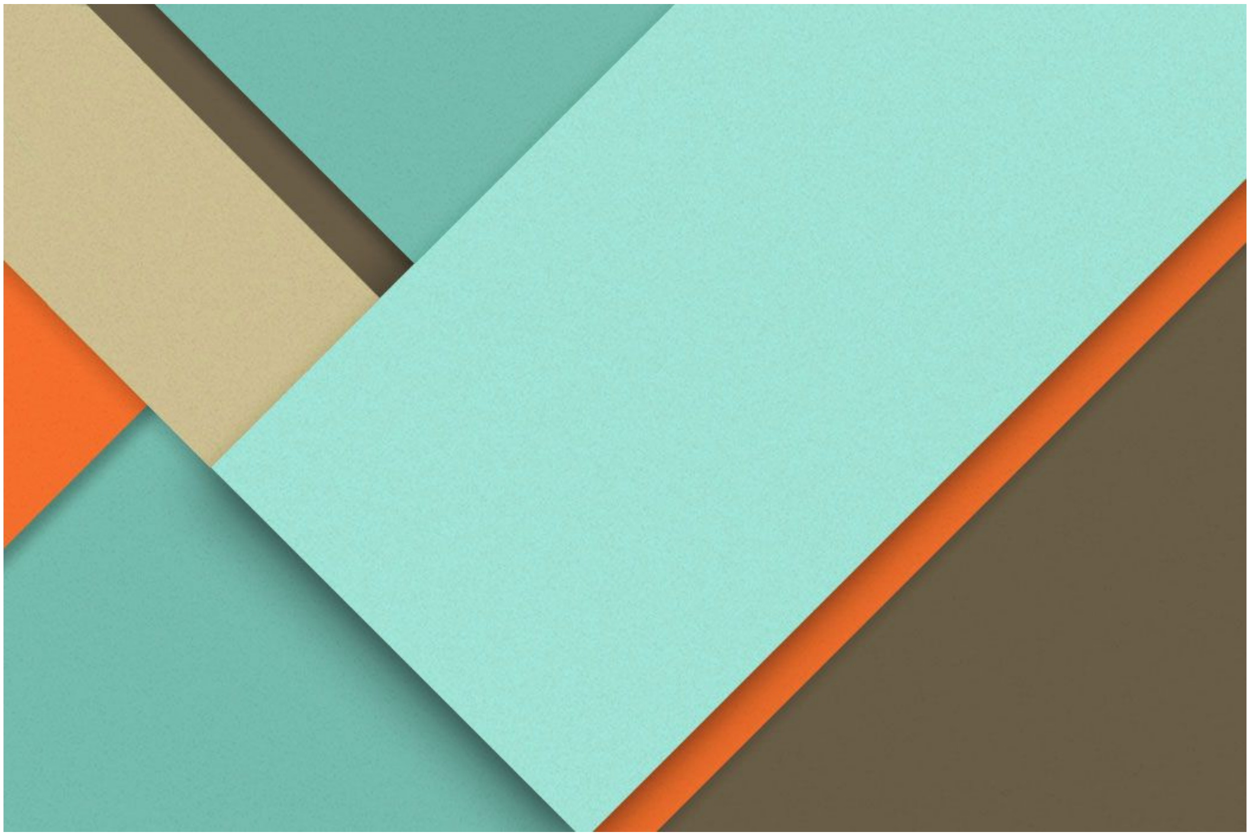
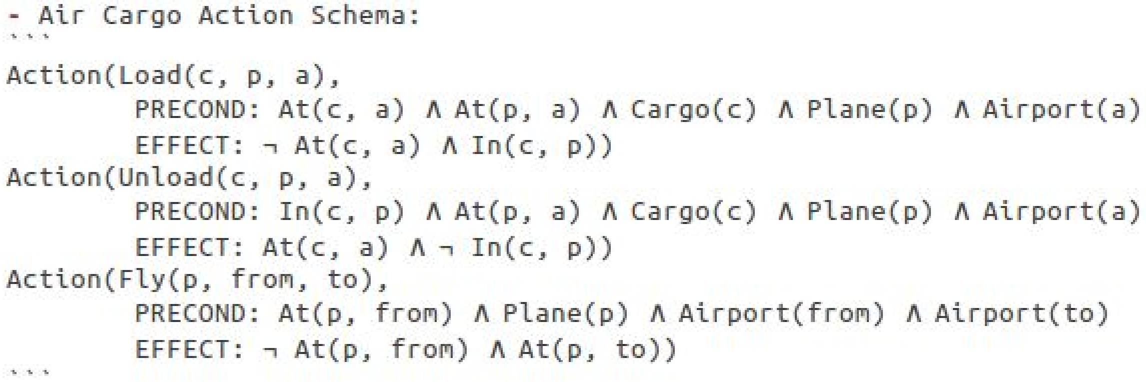


Heuristic Analysis

Name: Roba Mobtasem Ellamey

Evaluating Different Heuristic Functions:

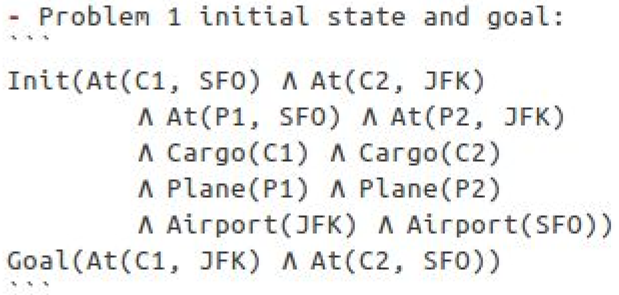
Problem Schema:



~~~~
- Air Cargo Action Schema:
```
Action(Load(c, p, a),
        PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
        EFFECT: ¬ At(c, a) ∧ In(c, p))
Action(Unload(c, p, a),
        PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
        EFFECT: At(c, a) ∧ ¬ In(c, p))
Action(Fly(p, from, to),
        PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
        EFFECT: ¬ At(p, from) ∧ At(p, to))
```
~~~~

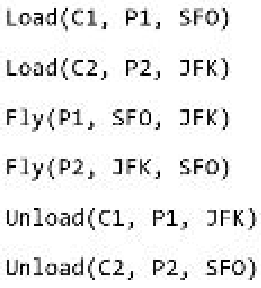
Problem 1:

Below is stated the initial state and goal of the first problem:



~~~~
- Problem 1 initial state and goal:
```
Init(At(C1, SFO) ∧ At(C2, JFK)
        ∧ At(P1, SFO) ∧ At(P2, JFK)
        ∧ Cargo(C1) ∧ Cargo(C2)
        ∧ Plane(P1) ∧ Plane(P2)
        ∧ Airport(JFK) ∧ Airport(SFO))
Goal(At(C1, JFK) ∧ At(C2, SFO))
```
~~~~

Optimum Solution:



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

Final Result Using Different Search Methods:

Search Method	*Plan Length*	*Time*	*Nodes Expanded*	*Optimality*
BFS	6	0.041	43	Yes
DFS	20	0.023	21	No
Greedy Best First Search	6	0.009	7	Yes

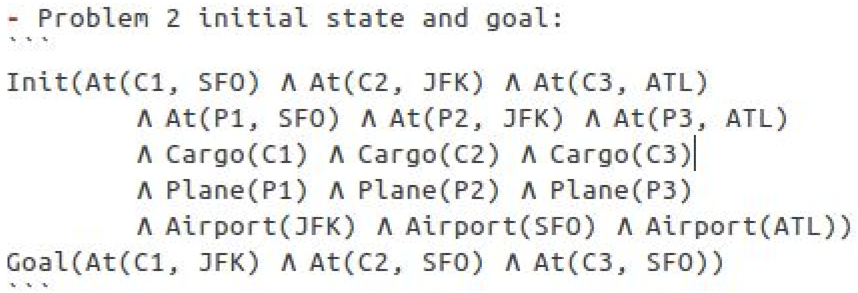
From the above table, it is obvious that **Greedy Best First Search** is the optimum search method where it gives best results in the least time possible with an average number of expanded nodes saving a lot of space in memory. BFS comes second where it takes more time and consumes more memory than Greedy Best First Search. DFS does not get an optimal result where the plan length is 20; however, it consumes less memory and takes less time than BFS.

Search Method	*Plan Length*	*Time*	*Nodes Expanded*	*Optimality*
A* (Ignoring preconditions heuristic)	6	0.054	41	Yes
A* (Using level sum heuristic)	6	0.91	11	Yes

From the above table, it is concluded that heuristics give better results when compared to uninformed search algorithms. The first technique takes less time the second to be done however it consumes a lot more memory than the second one.

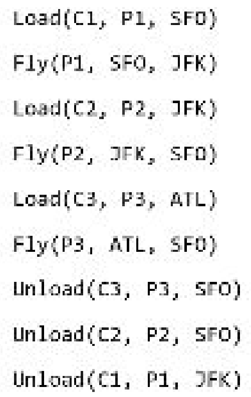
Problem 2:

Below is stated the initial state and goal of the second problem:



~~~
- Problem 2 initial state and goal:
```
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL)
        ∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL)
        ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3)
        ∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3)
        ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL))
Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))
```
~~~

Optimum Solution:



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

Final Result Using Different Search Methods:

Search Method	*Plan Length*	*Time*	*Nodes Expanded*	*Optimality*
BFS	9	17.3	3343	Yes
DFS	619	4.6	619	No
Greedy Best First Search	21	2.75	998	No

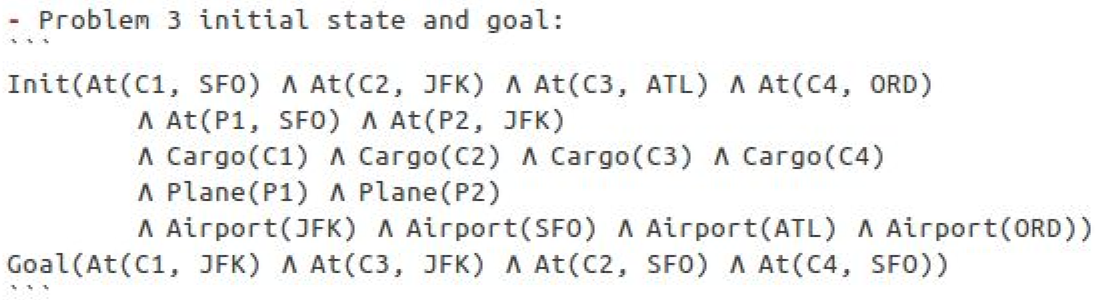
From the above table, it is obvious that **BFS** is the optimum search method though it consumes more memory and more time compared to the other two methods. However, it has the least plan length which makes it more suitable in this problem, but it has the maximum number of nodes expanded which is relatively very large compared to the other two methods.

Search Method	***Plan Length***	***Time***	***Nodes Expanded***	***Optimality***
A* (Ignoring preconditions heuristic)	9	5.8	1450	Yes
A* (Using level sum heuristic)	9	72	86	Yes

From the above table, it is observed that both heuristics are optimum at certain points. However, the time consumed by the second heuristic is way longer than the first one but it consumes a lot less memory that the other method.

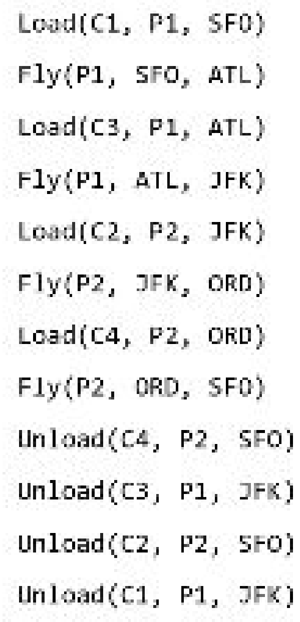
Problem 3:

Below is stated the initial state and goal of the third problem:



~~~
- Problem 3 initial state and goal:
```
Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧ At(C4, ORD)
        ∧ At(P1, SFO) ∧ At(P2, JFK)
        ∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧ Cargo(C4)
        ∧ Plane(P1) ∧ Plane(P2)
        ∧ Airport(JFK) ∧ Airport(SFO) ∧ Airport(ATL) ∧ Airport(ORD))
Goal(At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧ At(C4, SFO))
```
~~~

Optimum Solution:



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

Final Result Using Different Search Methods:

Search Method	*Plan Length*	*Time*	*Nodes Expanded*	*Optimality*
BFS	12	122.361	14663	Yes
DFS	392	1.956	408	No
Greedy Best First Search	16	18.469	5580	No

From the above table, for the third time the **BFS** is considered the best solution regardless of the memory loss and the large time consumed. DFS converged away from the optimum solution where the length of its plan was 392 while that of BFS is 12. Finally, the Greedy Best First Search is very close to the optimum solution while using less memory and consuming less time.

Search Method	***Plan Length***	***Time***	***Nodes Expanded***	***Optimality***
A* (Ignoring preconditions heuristic)	12	18.942	5040	Yes
A* (Using level sum heuristic)	12	363.24	315	Yes

Again like the previous two problems, both heuristics are optimum as they have a simple plan length however the second heuristic runs a lot more slower than the first one but consuming less memory.

Conclusion:

From the results of the above three problems, it is concluded that the **A* with Ignore Preconditions Heuristic** is considered the best search method from both sides; time and optimum plan length. However, it consumes a lot more memory than **A* with Level Sum Heuristic**, so it is better used if there is enough memory available.

In order to guarantee optimality when working hard problems, it is best to use A* search techniques. However, if the problems are simple, then it is prefered to use BFS as it performs better, but if speed matters in the problem then Greedy Best First Search is used.