1 Summary of Approach

1.1 Implementation Details

My code generates the permutations of symbols for different sizes, using these permutations generates the grounded action list to get the preconditions, effects, and map arguments for each condition. The planner function is where the main A* search is implemented and uses heuristics to guide the search, the traceback function is outside of the loop and traces the path back and returns a list of actions once the goal is reached. Similar to the first homework the key point for implementing the A* search algorithm is in the data structure for the closed and open list. I implemented the open list as a priority queue so that the states with the minimum f-states values are pushed to the top of the queue and the states are picked from the queue in that order. The closed list is initialized as an unordered map, to minimize how the list is searched through, instead of searching through the entire list it can look at the specific cell where the information is stored to correctly find the state that is being searched for.

1.2 Discussion

In Table 1 and Table 2, the results of my planner can be seen below running both with and without heuristics. It is evident in these results that the heuristics impacted the search in the case of the BlocksTriangle much more than it did the simple Blocks environment and the more complicated Fire Extinguisher environment. This is likely due to the heuristics that were used. I attempted to implement two different heuristics once being a simple heuristic that calculates the number of unsatisfied conditions in the goal state. In the blocks environment the impact of the heuristics is not heavily weighted, as the environment itself is relatively simple. This means that the heuristics would not have a significant impact on reducing the number of states expanded and guiding the search. In the Fire Extinguisher Environment, one potential reason it fails to help the search is that the heuristic is not able to accurately consider the complexity of the environment.

The second heuristic tried to use a more forward approach by implementing an admissible A* heuristic, and multiplying the h-value by a factor of 10 (modeling the weighted A* approach). However, this heuristic did not make much of an improvement to the performance of the planner, and frequently gave segmentation faults/bad memory allocation. This suggest that the weighted A* heuristic if implemented correctly could potentially provide better guidance in certain environments, however with the faults faced there is either an issue with the implementation itself or the approach might not be suited for the environments it is used to test upon. I was not able to solve these issues so I decided to keep the simple heuristic, however a more informed heuristic would have helped to guide the search even further and could have improved the results of the BlocksTriangle and FireExtinguisher environment such that the number of states expanded could be reduced and the time taken could also be reduced. That being said, the improvement I was expecting to see with the change in heuristic was that the number of states expanded would decrease for the Fire extinguisher environment as well as the BlocksTriangle environment.

Table 1: Results with Heuristics

Environment	Elapsed Time (s)	# of Expanded States
Blocks	0.00266361	5
BlocksTriangle	0.743856	104
FireExtinguisher	4.7566	1246

Table 2: Results without Heuristics

Environment	Elapsed Time (s)	# of Expanded States
Blocks	0.00476181	7
BlocksTriangle	2.6513	500
FireExtinguisher	4.83911	1246

2 Compiling Code

To compile the code simply run the commands provided in the homework prompt:

1. g++ planner.cpp -o planner.out
2. ./planner.out <path to environment description file>

By default the planner runs with heuristics, to run the planner without heuristics:

1. Uncomment the following lines:
    a. 836, 837, 838, 868, 869, and 870
2. Comment out the following lines:
    a. 826, 829, 830, 860, 861, and 862

Results will print at the end of each run, with the elapsed time and numbers of states expanded printing first, and then the set of actions printing afterward.