**Heuristic functions**

Data Aggregation – choose the edge with the smallest time latency next for *unicast*; otherwise, just pick the edge that was added next in the list

Monitoring – essentially just create all permutations of targets and sensors at every step until we get to a goal state, then backtrack

**Did outcomes make sense?**

Yes. So far, the *iddfs* creates less nodes and the frontier is minimized, *bfs* has the most created nodes and largest frontier size of all three algorithms because it has to travel one depth in the tree at a time (this means we have to exhaust most paths in the monitoring problem if we have to include all 4 sensors for example), and *unicost* has a slightly smaller frontier size than *bfs*. All of these are expected according to the charts discussed in class that displayed the time, space, and completeness of each algorithm. One thing to note is that every algorithm is complete, but in order for each one to be optimal, you have to explore every single possible goal state and then search through all the goal states at the end to find the most optimal one. Since we’re not doing that, the time and space of each algorithm is decreased. With that being said, *unicost* is the only algorithm that seems to produce the optimal solution most of the time.

**Anything surprising?**

Nothing surprising so far. Every traversal has gone to plan and my explored set helps prune cycles.