Structures

# Reader

// Data structure holding the travel times between planets.

Travel\_Times\* constraints; // FIRST STRUCTURE

// Planet name to planet object

std::map<std::string, Planet\*> planets;

// Planet-name pair to edge object

std::map<const Planet\*, std::map<const Planet\*, Edge\*> > edges;

// Ship name to id.

std::map<std::string, Ship\_ID> ships;

# Fleet

std::vector<std::string> names;

# Itinerary

std::vector<Planet\*> destinations;

std::vector<Leg> legs;

# Edge

std::vector<Leg> departures;

# Planet

// edges shows the connections between this planet and it's

// neighbors. See class Edge.

std::vector<Edge\*> edges;

# Galaxy

std::vector<Planet\*> planets;

# Priorityqueue

std::vector<Thing\*> data\_;