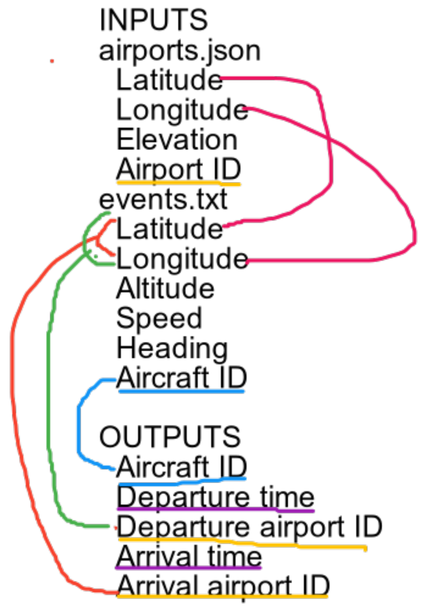

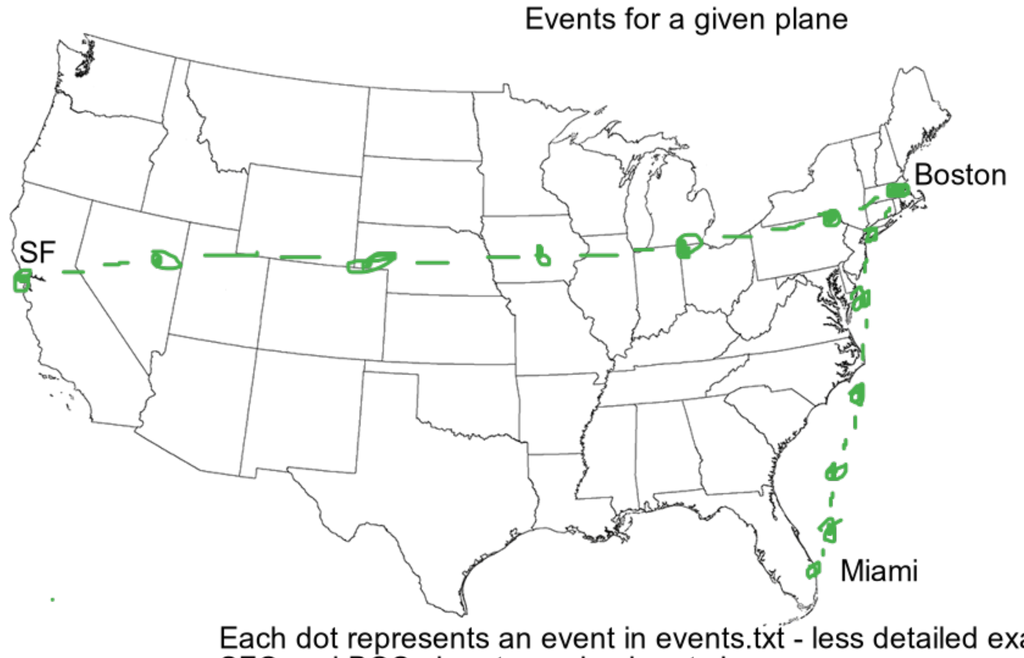

How do we figure out departure time and arrival time we don't have the time of the events but can we use speed?

Each dot represents an event in events.txt - less detailed example than real data
SFO and BOS airports are in airports.json
There is an event at (long,lat) of SFO and an event at (long,lat) of BOS

Timestamp is a field in adbsEvent.java

High Level Idea on What I Need to Do
1. Iterate through the list of airports, create an instance of the Airport object, and then store the Airports in some sort of list or dictionary (whatever collection is most efficient)
2. Iterate through the events, potentially weed out events that don't make sense, and create probable flights by analyzing the events.

Event analysis in detail:
For each event...
take a look at the aircraft ID. If the event's altitude is specified and > 0, then there is no departure or arrival activity
if the plane is at an airport with zero altitude and this is the first event where the plane is at an airport, we are preparing for departure
when the plane leaves the ground during a subsequent event, that is when the plane has left the ground (this may require more finesse using speed/altitude/general physics)
when a departure time has been found, create a new flight with new departure time, departure airport ID, aircraft ID
keep track of the flight and monitor the flight
when the plane has an altitude of 0 (we're not just flying over an airport) and we are at an airport (long/lat), then log the time of that event and mark it as arrival time. get ID of the arrival airport and store that.
Add that flight to the list of probable flights
^Make sure you manage this for the large number of possible concurrent flights on different aircrafts. but an aircraft can only have one flight at a time
When we are done iterating through the events.txt file, return the list of probable flights

Weeding out events that don't make sense in detail:
analyze the speed. is the event consistent against previous events given speed and altitude