<http://web.cs.wpi.edu/~cs2223/d08/Exams/Exam2/>

VARIABLES TO CONSIDER:

* Number of guests in party
* Max occupancy of resturant
* Amount and type of food ordered (are we considering this for every meal on the menu??)
  + Preparation times
  + Order rate
* Reason for visit
  + Dinner
  + Celebration
  + Groups
* When to seat new arrivals vs seating those on waitlist when tables fill up

DATA TO COLLECT:

* Arrival and departure times
* Overall time it takes for a customer to get through queue
* Time it takes for:
  + Seating, ordering, paying
* Party sizes
* # of tables
* # of servers
* # of tables/server
* Preparation times
* Difference in times of dining based on reason for visit
* Approximate time guests stay after finishing meal

TO CONSIDER:

* Large party walk-ins
  + Multiple tables near each other
* Seating two people at four person table when no two person tables available, for example

ALGO:

* Data structures:
  + Menu (maybe???)
  + Servers
    - Number of servers
    - Number of tables per server
  + Parties
    - Walk-ins
    - Reserved
    - Average time calculated based on party size and occasion
    - When they entered the restaurant
    - If they left
    - How long they stayed
  + Tables
    - Number of tables in restaurant
    - Occupancy of each table
    - Whether table is occupied
      * Party size
      * Time seated
      * Which server
  + Waitlist
    - List of parties on waitlist
    - Calculated wait time
* Array for reservations
* Array for walk-ins
* Allotting multiple tables for large party rather than using those tables to seat smaller parties

EXTRA:

* If reserved party is more than 15 minutes late, cancel reservation
* When new party walks in packed restaurant, need to find table closest to finishing their meal and leaving the restaurant
* As waitlist grows, need to match every new walk-in with a table
* Host needs to be able to log when party is seated and when party has left - DESIGN
* Wait times must be constantly updated as parties leave restaurant