To do for backend:

* ~~Make sure that the email is fixed to be able to receive a picture~~
* ~~Integrate the server settings into the client email~~
* ~~Add archive home functionality to backend~~
* ~~Add unsubscribe functionality~~
* Debug unsubscribe functionality
* Update routes for react app so that front-end can get the public info for a task with just that task’s UUID
* Create backend email generation:
  + Takes in a contractorUUID and generates a lightweight email to the contractor that contains the (1) time(s) available, (2) client info
* Harden Database:
  + Handle issue when the email is incorrect for the client and an email cannot be sent
  + Handle input validation for data
  + Create a series of helper functions for backend input validation
* Update backend so that we set a field in the task upon the client scheduling an email
* Create an API endpoint for rating of contractors
* ~~Add email retrieval for realtor functionality~~
* ~~Handle client email not defined issue~~

First layer squares:

(I,j)

Second layer squares:

(each connects to all the squares that it could possibly connect with as edge)

Source Node 🡪 Connects to all nodes\_squares with edge of weight 1

Each node has an edge of weight of one that connects to each other node that it can connect to in a second layer

Mark nodes alternating as either a src or a destination such that each source only touches destinations and vice versa

First layer – all possible nodes for sources with an in-edge of exactly one and an out edge to all destination nodes that they touch – enforces that no node can be both a src and a destinaton

The second layer is all the possible destination nodes, each with exactly one out edge and in-edges from all touching source nodes – enforces that each destination can be included at most once

We see exactly one src node per destination node in max flow

Thus, this should work