



Intra-Building Maps

Meet The Team



David Koplw
6-9 Sophomore



Raul Alcantara
6-3 Senior



Suzanne Jiang
6-3 Freshman



Owen Dugan
8 Freshman



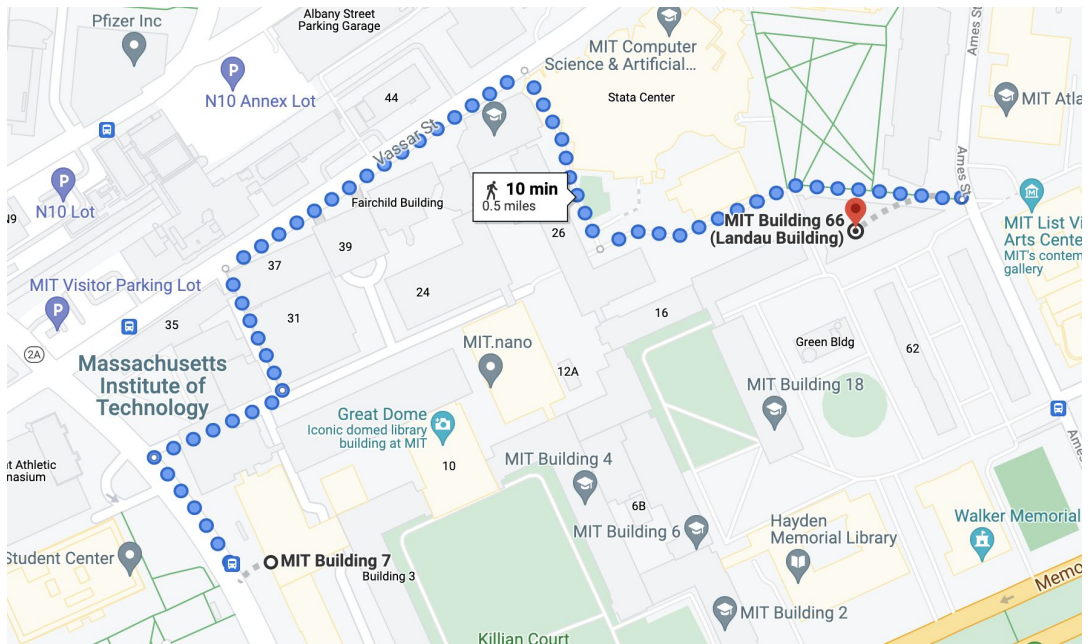
Claire Lu
6-9 Sophomore



Gatlen Culp
6-2 Freshman

Where the **** is 5-233?

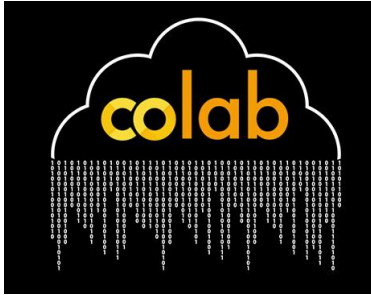
But but...Google Maps? 😞



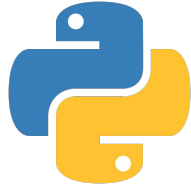
But better

- Go through buildings
- Get to the exact room that you're looking for

Technologies used



Data Processing



Firestore

Data storage



App Development

Workflow: Get the data

Scrape floor plans
from
floorplans.mit.edu/



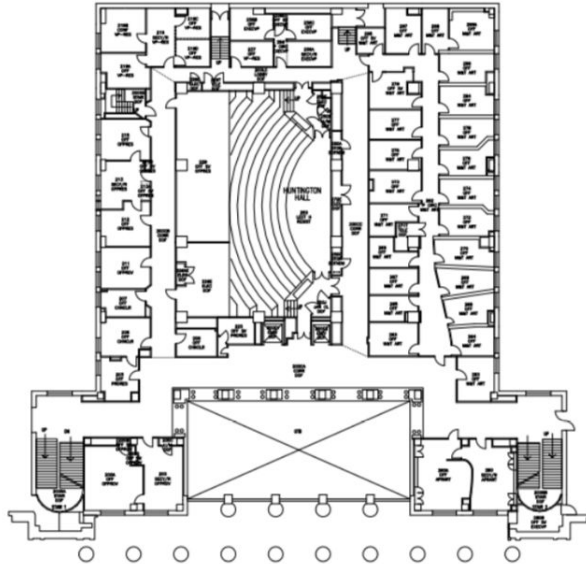
Clean up floor
plan SVGs



Create a graph
from the SVG



Read floor plan
text to identify
rooms



Workflow: Get the data

Scrape floor plans
from
floorplans.mit.edu/



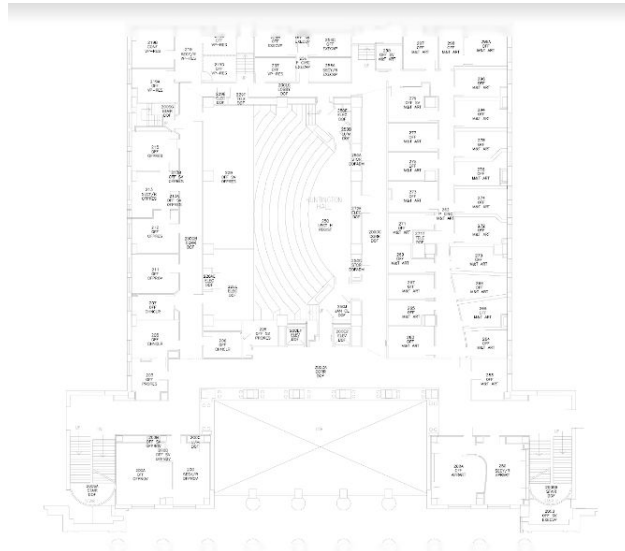
Clean up floor
plan SVGs



Create a graph
from the SVG



Read floor plan
text to identify
rooms



Workflow: Get the data

Scrape floor plans
from
floorplans.mit.edu/



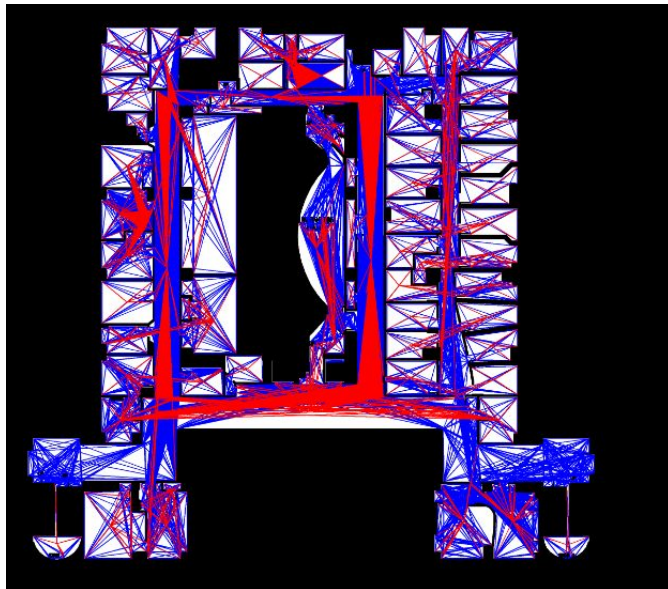
Clean up floor
plan SVGs



Create a graph
from the SVG



Read floor plan
text to identify
rooms



Workflow: We got the data, now what?

Display floorplans
on the app



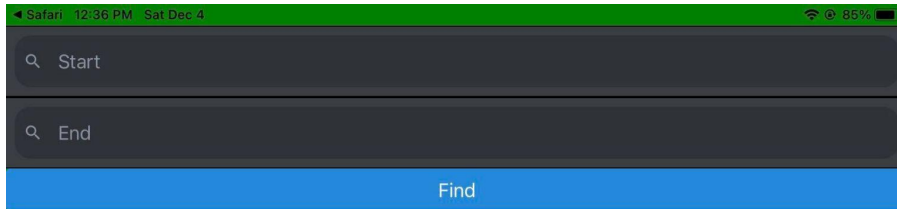
Get start and end
destinations from
the user



Identify the best
path



Draw path on
the screen

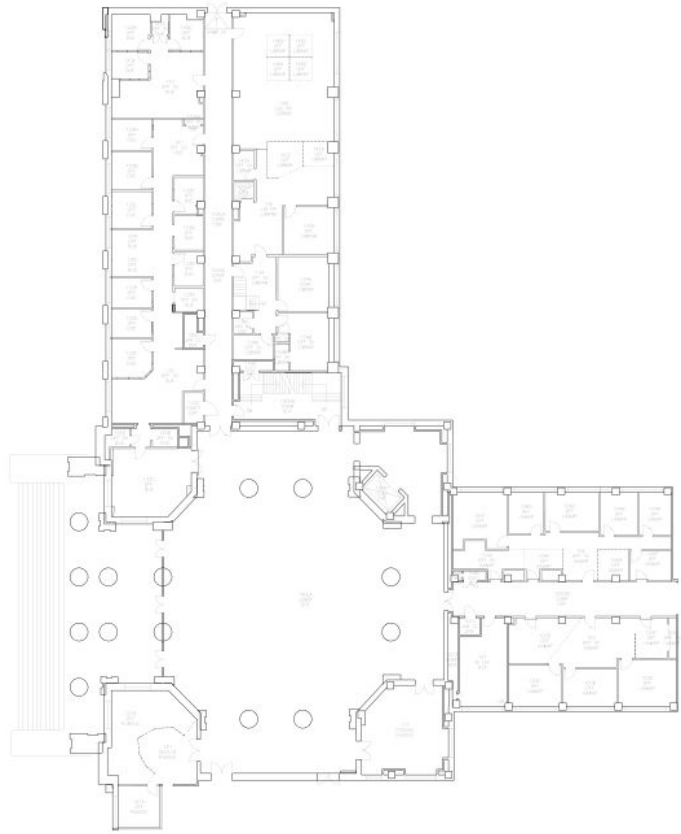


Final
result:

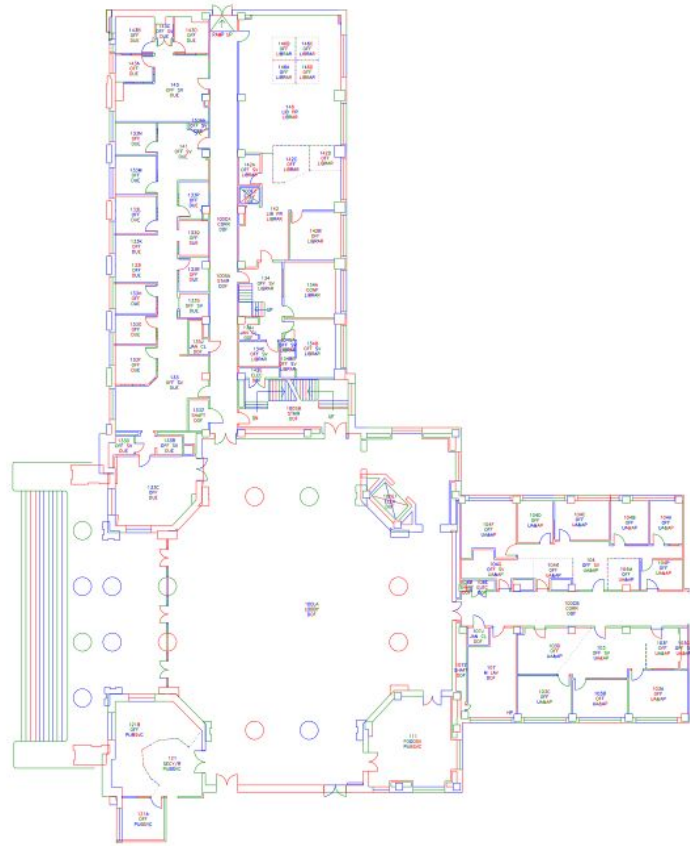




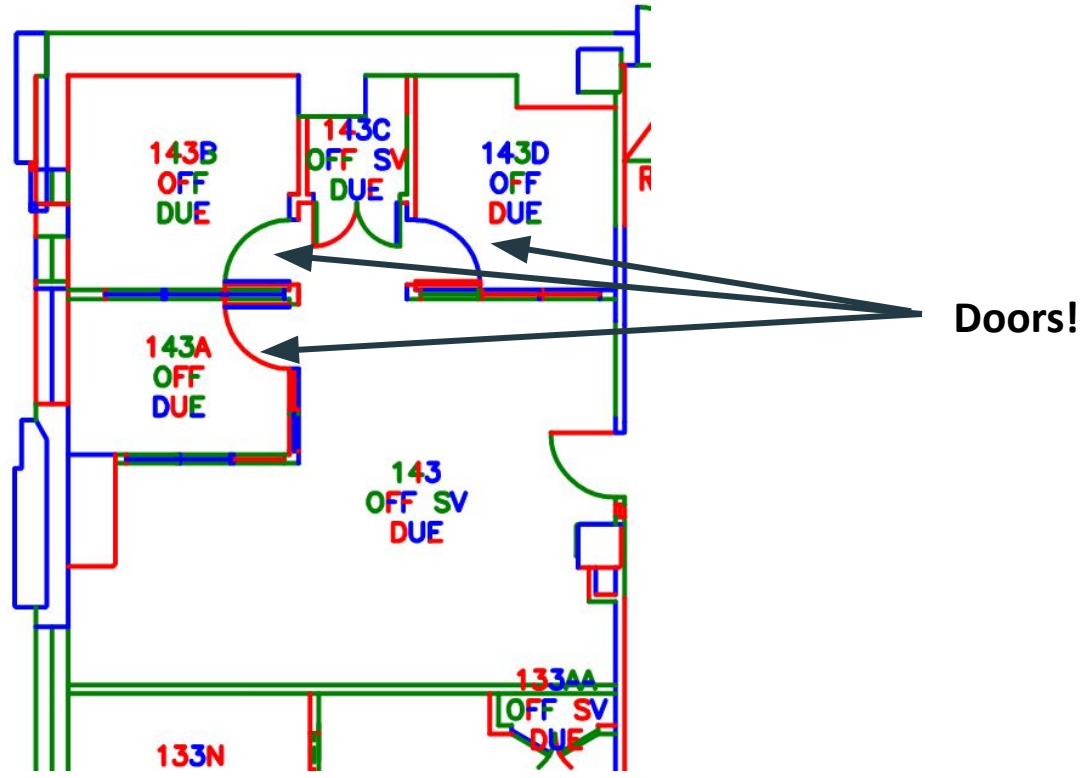
Floor Plan Processing with SVGs



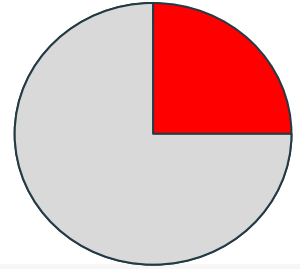
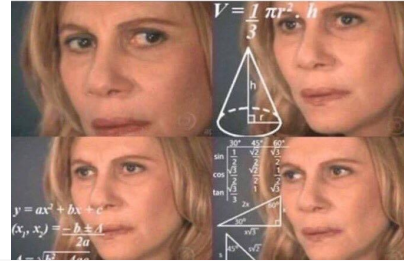
Paths!



Doors

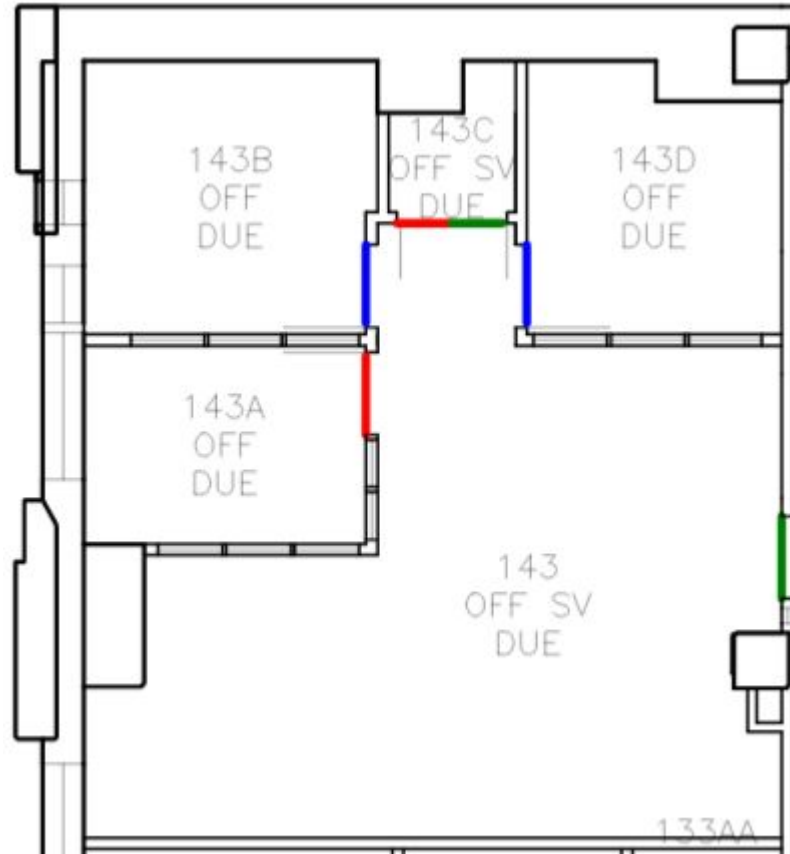


Detecting Doors

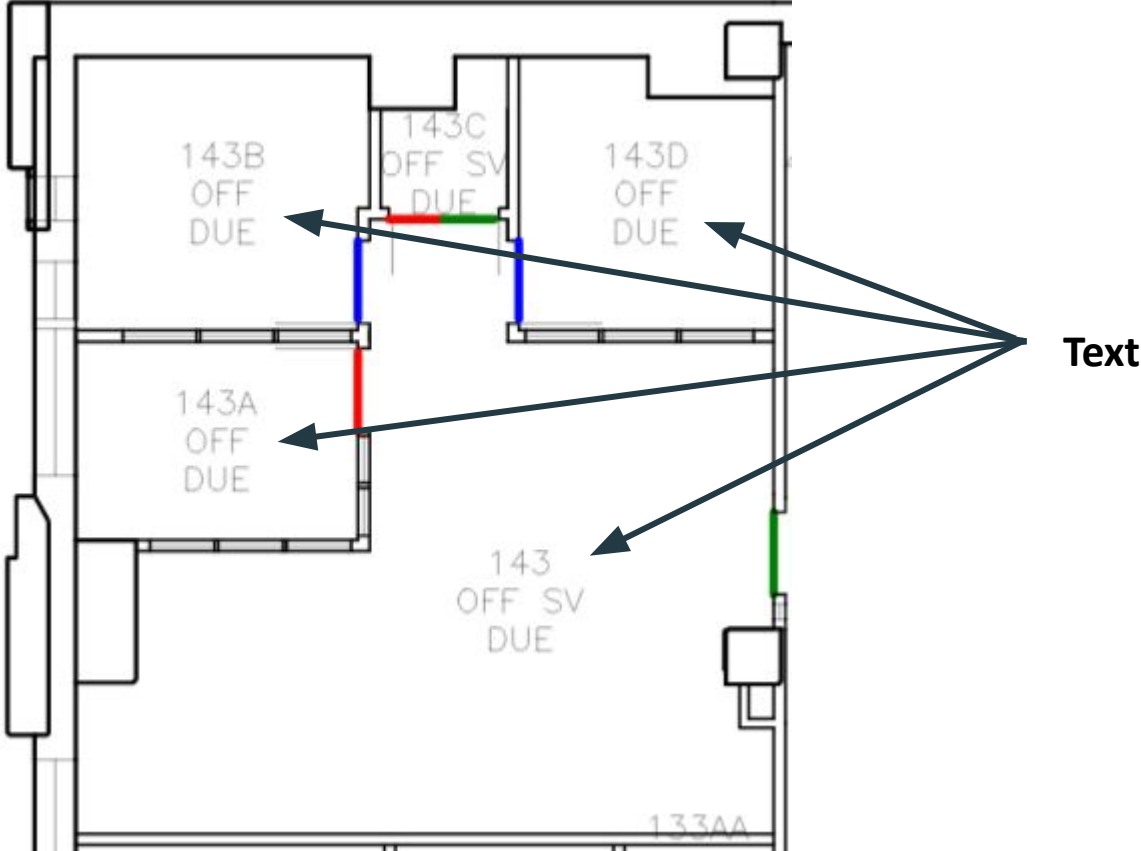


```
def is_door(path, attribute):
    # Making sure the path is closed, continuous and curved
    if path.isclosed() or not path.iscontinuous() or "C" not in attribute['d']:
        return False
    start, end = path.start, path.end
    diff = start-end
    segment_length = math.sqrt((diff.real)**2 + (diff.imag)**2)
    curve_length = path.length()
    q = curve_length / segment_length #should be pi / 2sqrt(2) coz math
    good_q = math.pi / (2 * math.sqrt(2))
    return abs(q/good_q - 1) < 1e-2
```

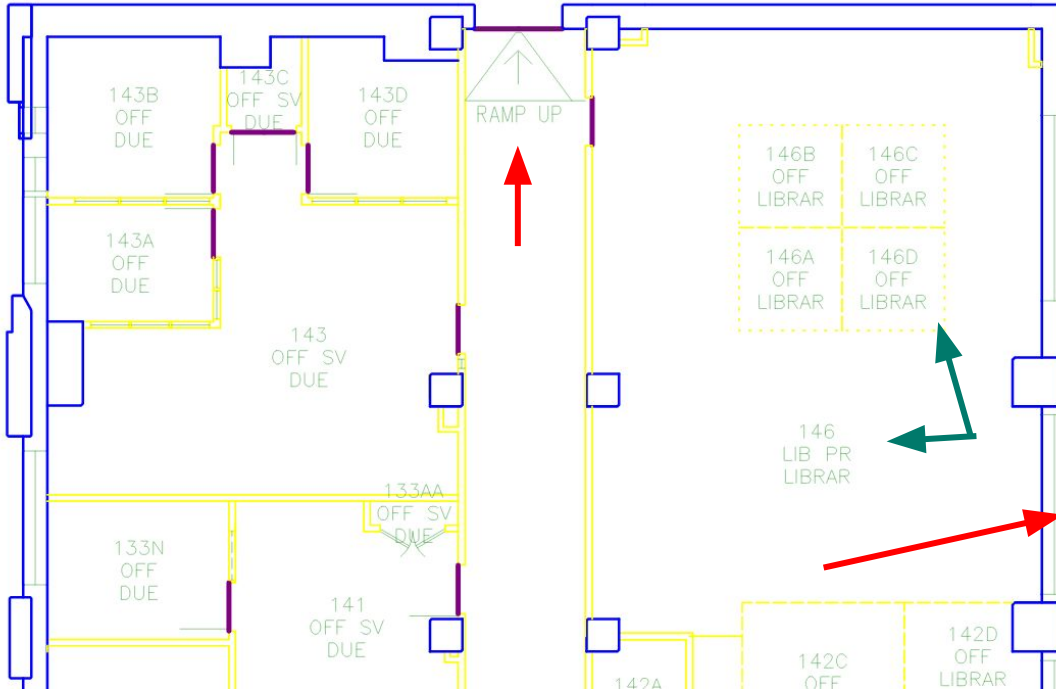
Result:



Detecting Text



Idea: separate everything by its stroke width



Good:

- All the text seems to be of the same width!

Bad:

- Other non-text stuff seems to be caught
- Widths of text is not consistent over different floor plans

Conclusion:

(or at least we thought so)



Solution

1. Stroke-Width: Keep the paths with the width of a text
2. Letters are not too long, so only keep the ones under a given threshold
3. Apply OCR to get the text

Step 1: Divide by width



Step 1: Result



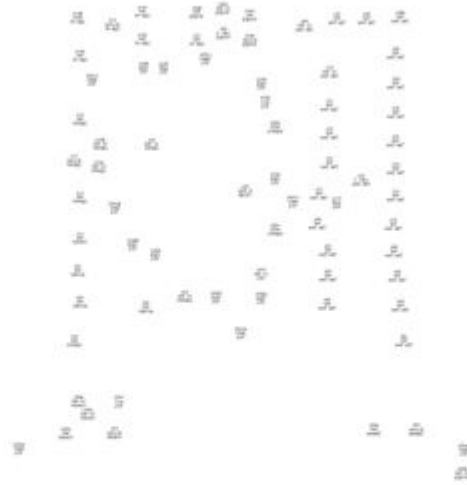
Copyright 2009-2011 Massachusetts Institute of Technology. All rights reserved. Please see the MIT open licensing policies for details and use of the design icon.



Step 2 Divide by length



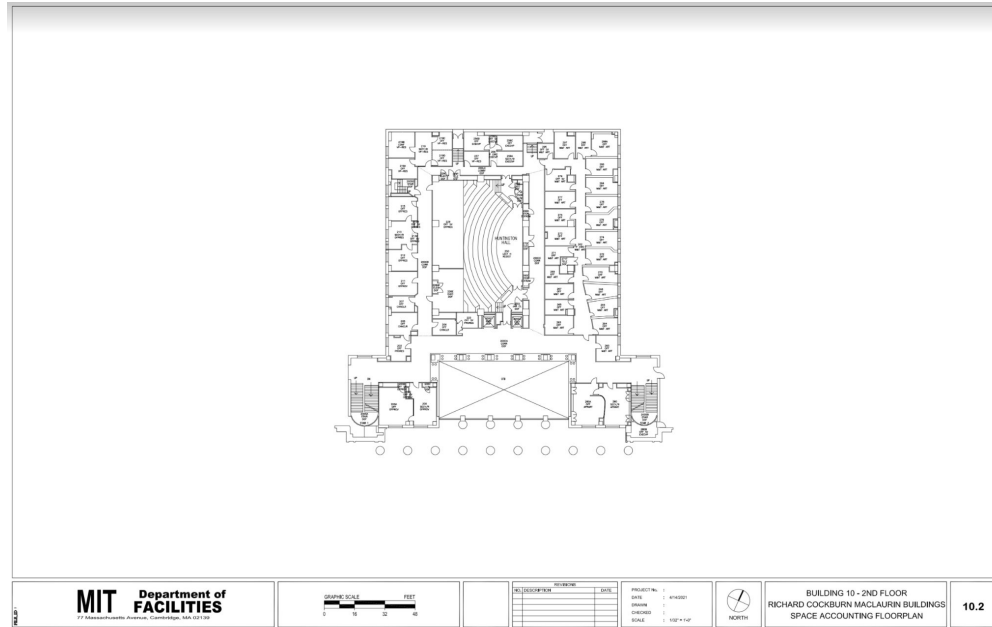
Step 2: Result



Step 3: OCR

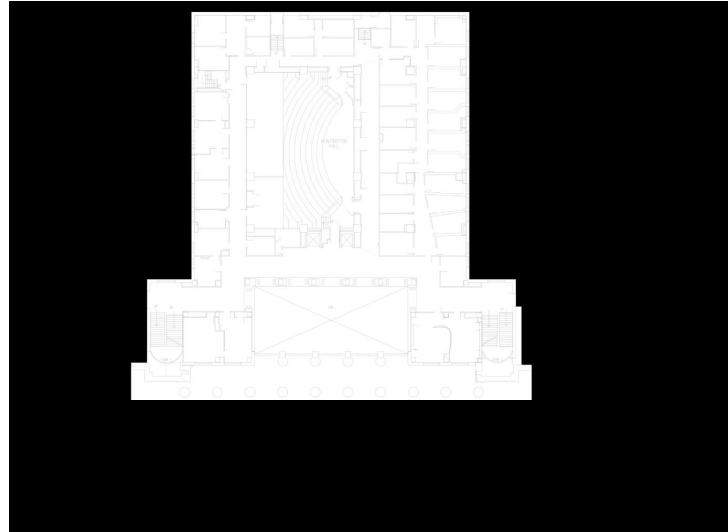
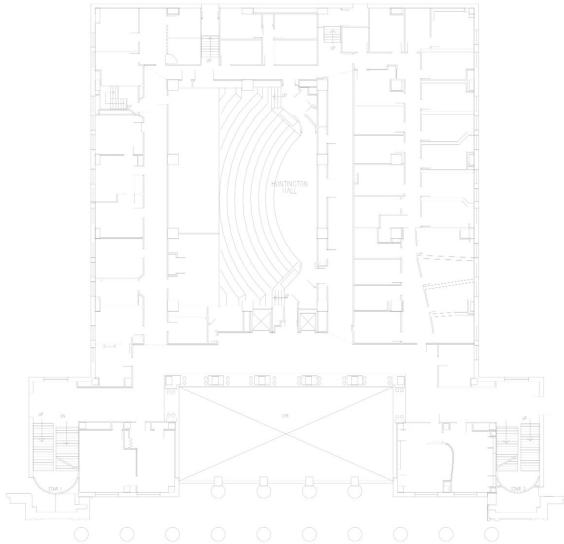
219B CONE VP??RES
219 SECY/R VP??RES
219C VERE
219D OFF VP??RES
219A OFF VP??RES
200356 STAIR DOF
215 OFF OFFRES
256B OFF EXECVP
256D post
256C OFF EXECVP
296 P
296 OFF M&T ART
200LC LOBBY Pon.
296 OFF M&T ART
296 OFF SV M&T ART
294 OFF M&T ART
250B U/M DOF
277 OFF M&T ART
278 OFF M&T ART
250A STOR DQADM
298 J GFF F M&T ART
295 OFF SV M&T ART
298

Generating Graph



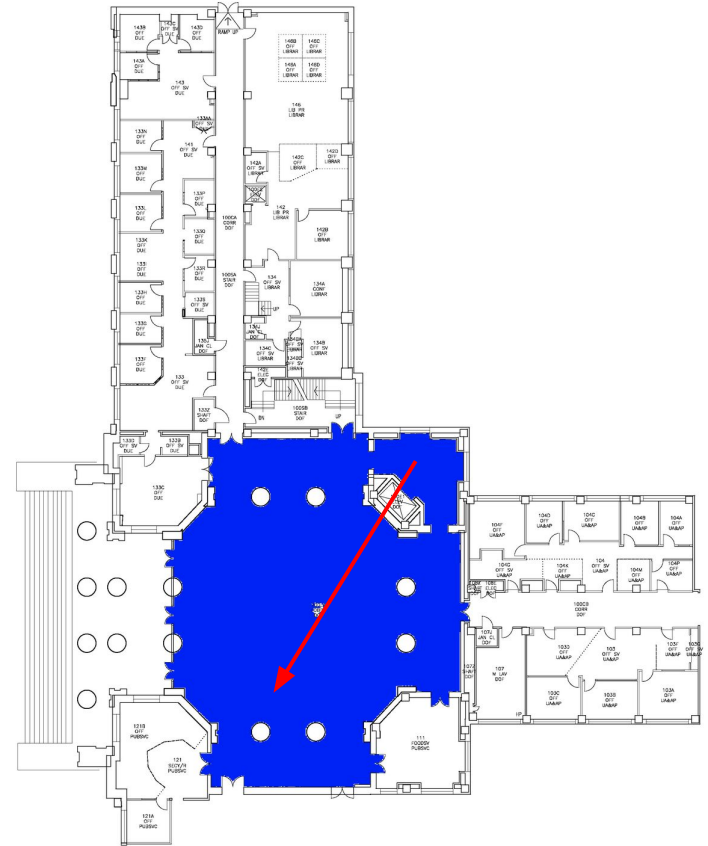
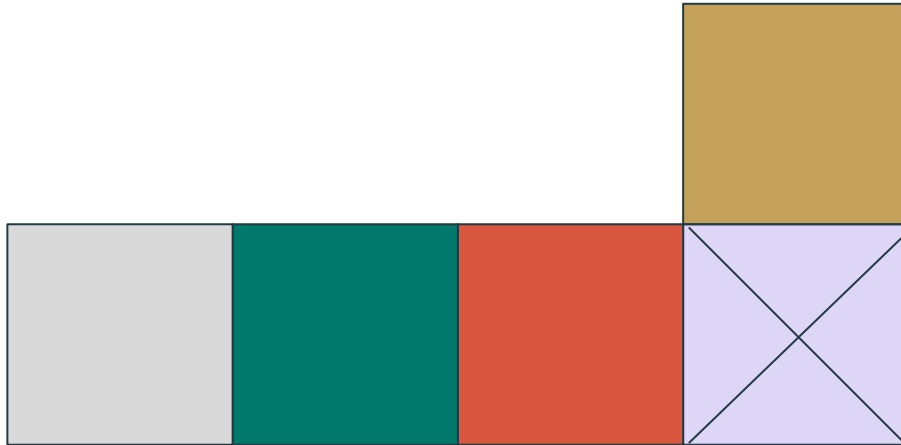
MIT Department of FACILITIES 77 Massachusetts Avenue, Cambridge, MA 02139	GRAPHIC SCALE 0 10 20 30 40 FEET	REVISIONS NO. DESCRIPTION DATE	PROJECT NO. 1	NORTH	BUILDING 10 - 2ND FLOOR RICHARD COCKBURN MACLAURIN BUILDINGS SPACE ACCOUNTING FLOORPLAN	10.2
			DATE 4/16/2011			
<small>Copyright 2000-2011 Massachusetts Institute of Technology. All rights reserved. Plans are for MF space accounting purposes only and are not for design use.</small>						

Getting Graph

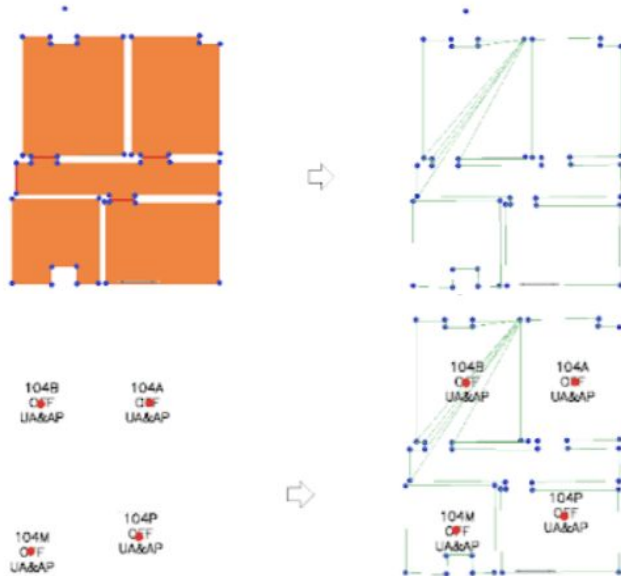


Room Filling

1. Examine squares around a point to determine if a point is in a room
2. Spread outward from starting point

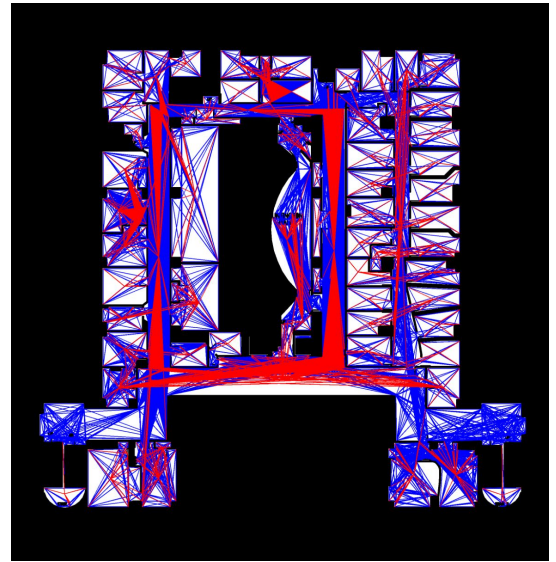
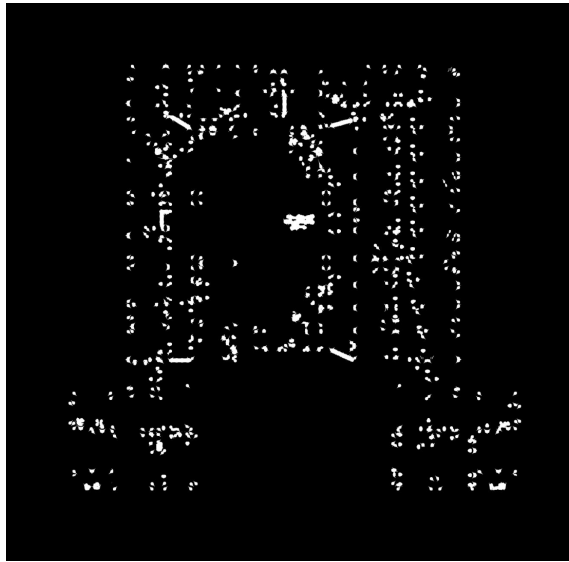
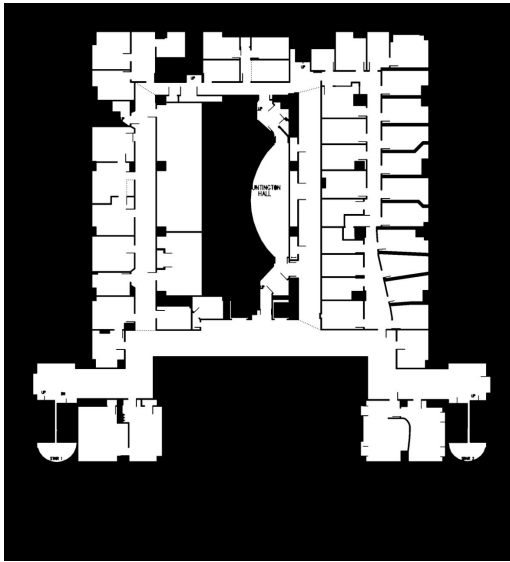


Creating the Graph

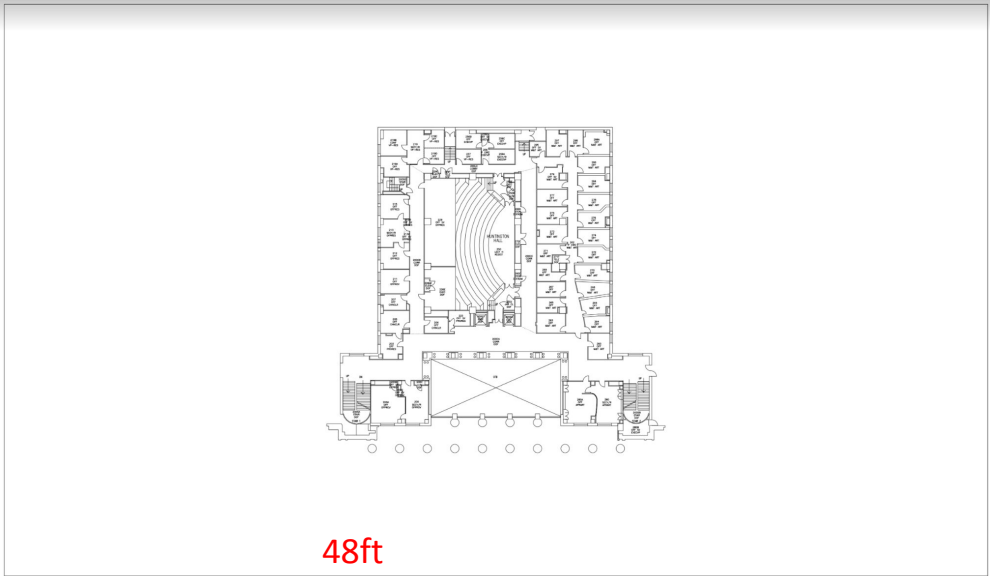


Use room corners and text as nodes, connecting nodes by line-of-sight




Getting Graph



Scale



48ft

 MIT Department of FACILITIES <small>77 Massachusetts Avenue, Cambridge, MA 02139</small>		<table border="1"><thead><tr><th>NO.</th><th>DESCRIPTION</th><th>SCALE</th></tr></thead><tbody><tr><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td></tr></tbody></table>	NO.	DESCRIPTION	SCALE													<table border="1"><tr><td>PROJECT No. 1</td><td>4/2007</td></tr><tr><td>DATE</td><td>1/10/07</td></tr><tr><td>DRAWN</td><td> </td></tr><tr><td>CHECKED</td><td> </td></tr><tr><td>SCALE</td><td>1/8" = 1'-0"</td></tr></table>	PROJECT No. 1	4/2007	DATE	1/10/07	DRAWN		CHECKED		SCALE	1/8" = 1'-0"		BUILDING 10 - 2ND FLOOR RICHARD COCKBURN MACLAURIN BUILDINGS SPACE ACCOUNTING FLOORPLAN	10.2
	NO.	DESCRIPTION	SCALE																												
PROJECT No. 1	4/2007																														
DATE	1/10/07																														
DRAWN																															
CHECKED																															
SCALE	1/8" = 1'-0"																														
<small>Copyright 2000-2011 Massachusetts Institute of Technology. All rights reserved. Plans are for MF space accounting purposes only and are not for design use.</small>																															



The App

Frontend

- Searchbars
- Button
- Background Image

Q start

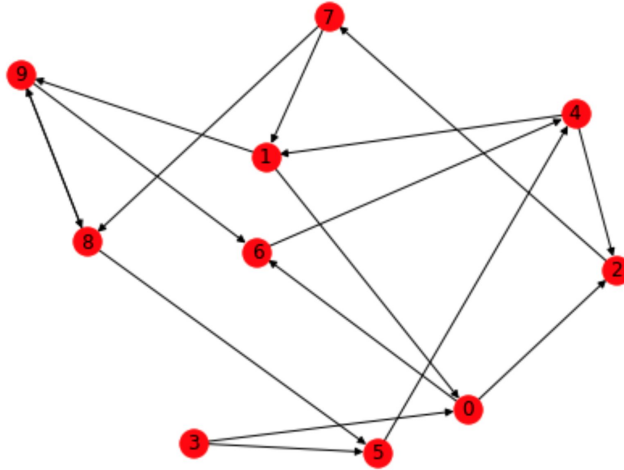
Q end

Find



How to Get to a Room?

- Graph
- A* algorithm



Firestore

- Database that stores the graph
- 2 models: nodes and edges

Nodes

```
name: "i"  
x: 100  
y: 50  
z: 1
```

Edges

```
end_id: "rjyErvv0bMeLmXT8jVso"  
end_name: "g"  
start_id: "wsahAwWBNRh7DyMdvh3Q"  
start_name: "a"
```

The link between UI and Firebase: `getGraph`

Function that returns the nodes and the connections from the graph

```
const getGraph = async () => {
  let [nodes, positions] = await getNodes();
  let edges = await getEdges();
  let connections = [];
  for (let i = 0; i < nodes.length; i++) {
    connections.push([]);
  }
  for (let edge of edges) {
    let start_id = edge["start_id"];
    let start_index = positions[start_id];
    let end_id = edge["end_id"];
    let end_index = positions[end_id];
    connections[start_index].push(end_index);
    connections[end_index].push(start_index);
  }

  return [nodes, connections];
};
```



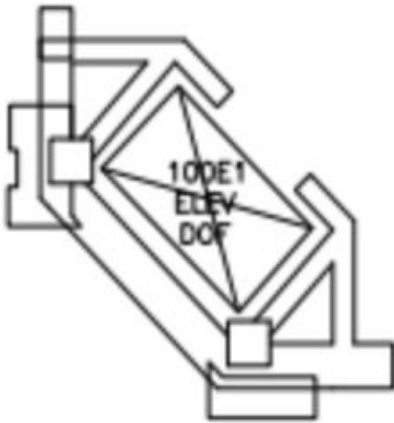

Demo!



Successes :) and Failures :(

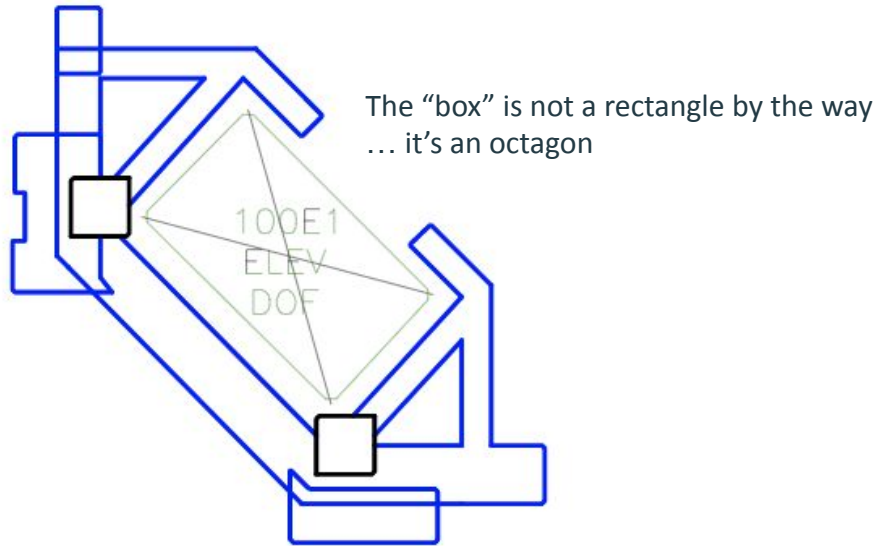


Identifying Elevators



- Elevators and stairs are necessary to move between different floors
- Elevators on floor plans are a bit more distinct

Unexpectedly difficult



- Inconsistent
- Always picked up weird stuff
- Made some progress by splitting up every path into smaller paths

Connecting floor plans



campus map

Search Reset Map

Print Email Share

Welcome to the MIT Campus Map. You can pan the map by dragging it with your mouse. **Enlarge** (zoom) the map with the + and - controls.

Other things you can do:

- **Search** the map, e.g.,
 - visitor center
 - food
 - hotels
 - krege auditorium
 - building 46
- **Click** anywhere on the campus to learn more about that location. **Right-click** on the resulting marker to remove it.
- **Share** (or bookmark) and **print** maps using the links in the upper-right corner.
- **Show** additional landmarks:
 - Emergency Phones
 - Covid Pass Access Points
 - Accessible Entrances & Ramps
 - Bike Racks

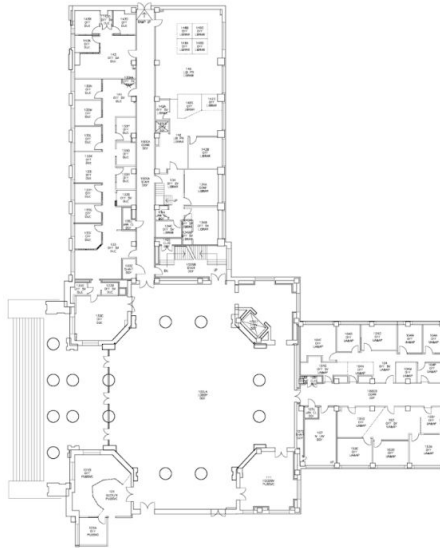
Need more help? Read the [frequently asked questions](#).



Map data © MIT, ESRI | Keyboard shortcuts | Image may be subject to copyright | Terms of Use

← Expand Show: Emergency Phones Covid Pass Access Points Accessible Entrances Bike Racks

Matching North



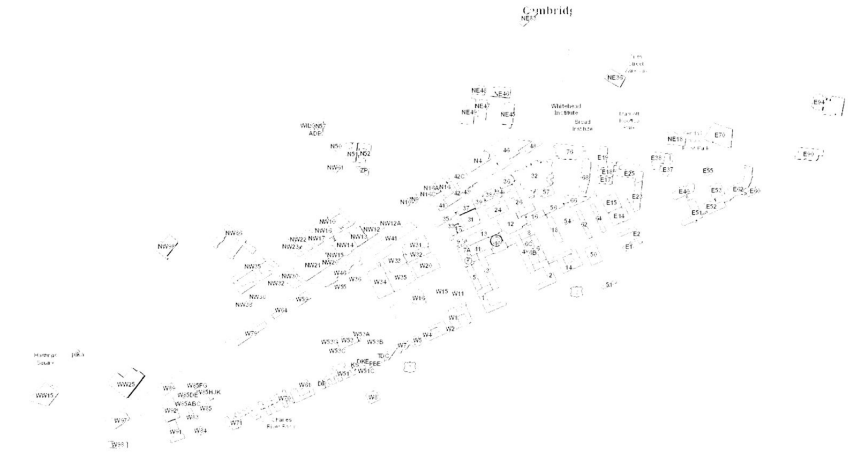
Matching North



Masking



North



Matching North



What's next?

- Multiple floors
- Multiple buildings
- ~~Campus domination~~