

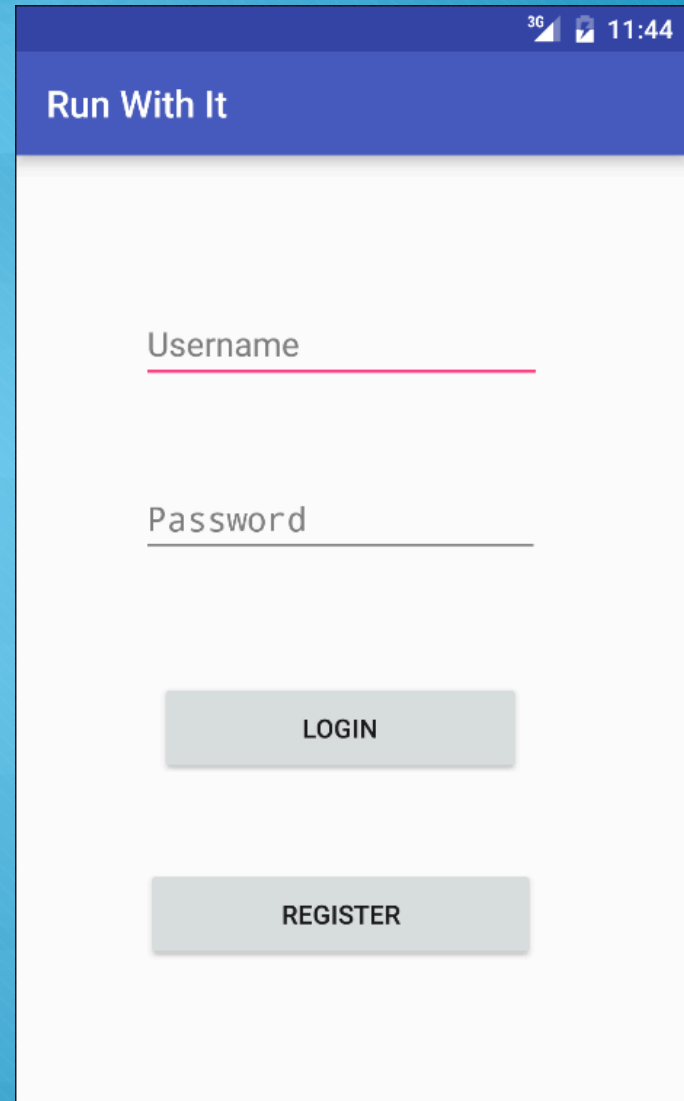


Run With It App Demo

By Gary Zhuo and Jerry Mieles
CS499s1 July 2016

Login Page

- Very simple (as it should be)
- Once users login they'll go straight to the map/route finding page

A mobile application interface for a login page. At the top, a dark blue header bar contains the text "Run With It" in white. Below the header, the background is white. There are two input fields: "Username" with a pink underline and "Password" with a grey underline. Below the "Password" field, there are two grey buttons with black text: "LOGIN" and "REGISTER". In the top right corner of the screen, there is a status bar showing "3G", a battery icon, and the time "11:44".

3G 11:44

Run With It

Username

Password

LOGIN

REGISTER



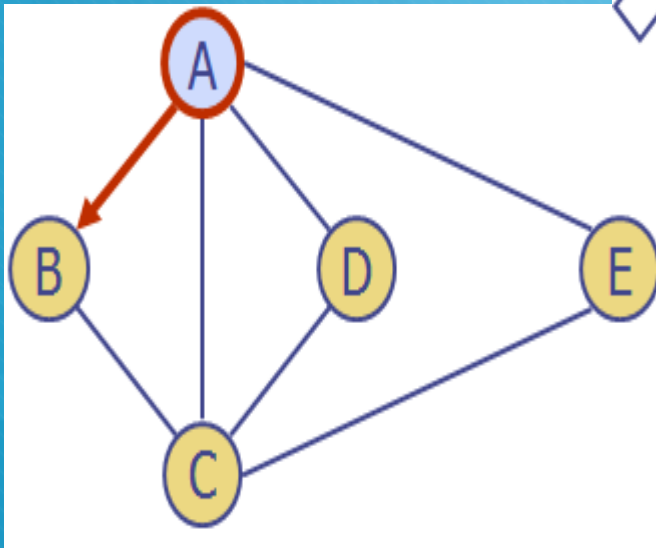
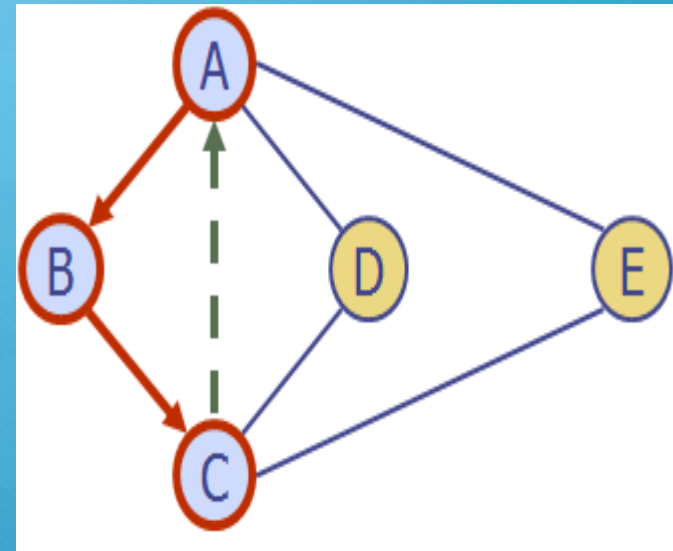
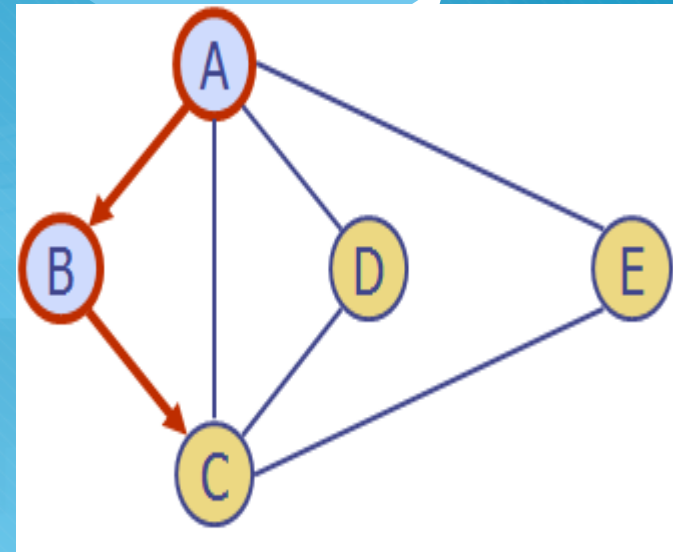
Depth-First Search

We use DFS as a basic concept for our app algorithm

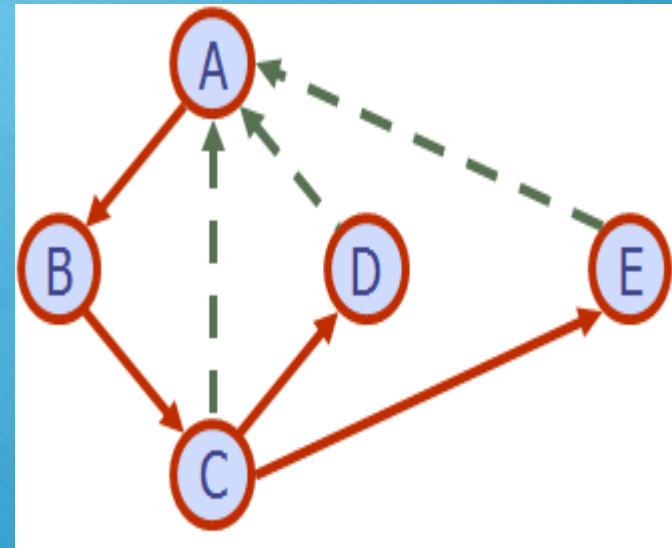
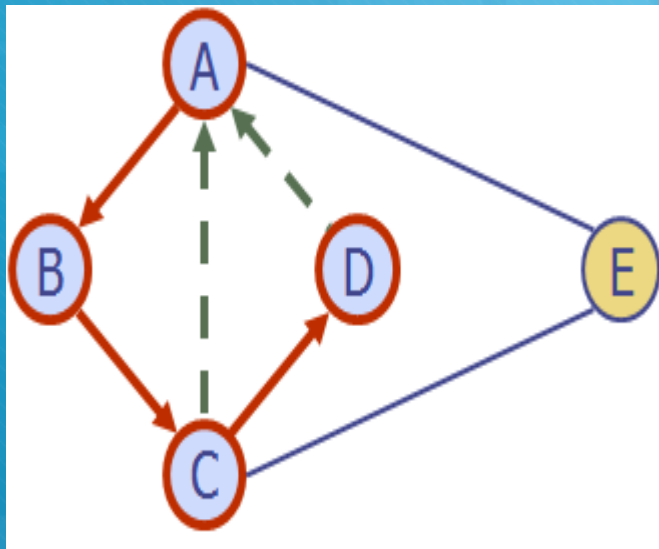
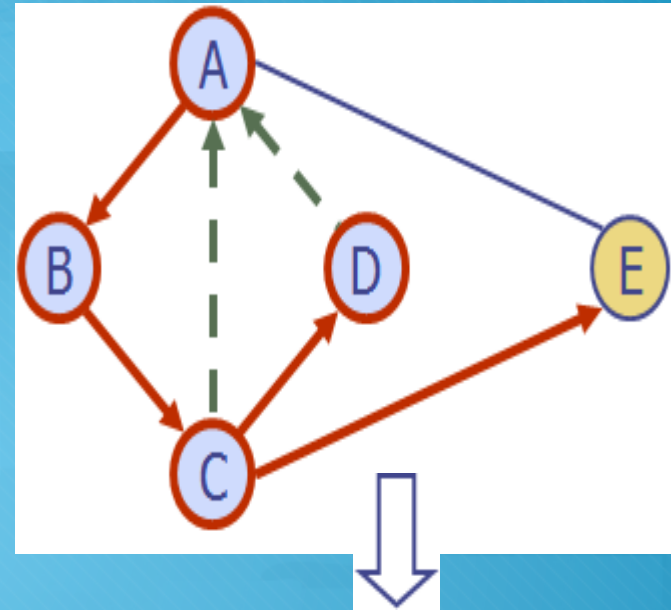
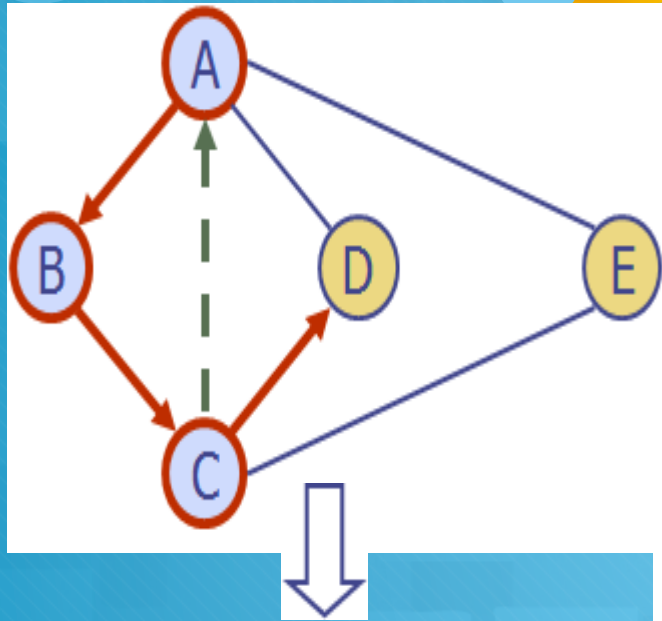
Depth-first search (DFS) is a general technique for traversing a graph

When choosing which vertex to explore next, we favour children over siblings

Example



Example (cont.)



Path Finding

the DFS algorithm can be specialized to find all paths between two given vertices.

- It always choose a adjacent vertex as the next one
- Everytime it goes back, it find a possible route

Our algorithm

DFS(depth,v,path):

path.push(v); //add the current vertex to the path

visited.push(v);

if (depth == 0 && v is target):

print path

Return //stop clause for successful branch

if (depth == 0):

return //stop clause for non successful branch return

for each vertex u such that (v,u) is an edge:

DFS(depth-distance(v,u),u,path) //recursively check all paths for
of shorter depth

path.removeLast(); // clean up environment

visited.removeLast();

Our algorithm(cont)

- How to find cycles?

We find all routes ending in the vertices which are adjacent to the start vertex.

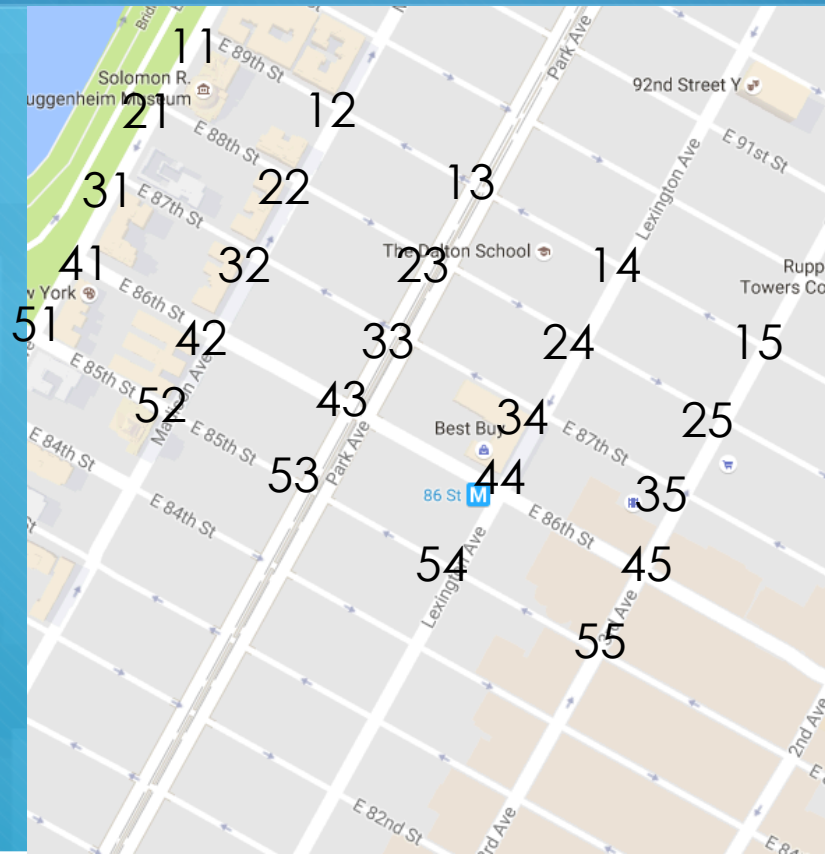
Then add the start vertex in the end to be a cycle.

- Gain all cycles with specific length after DFS
- Difficulty of routes: sum of all absolute values of difference of elevations



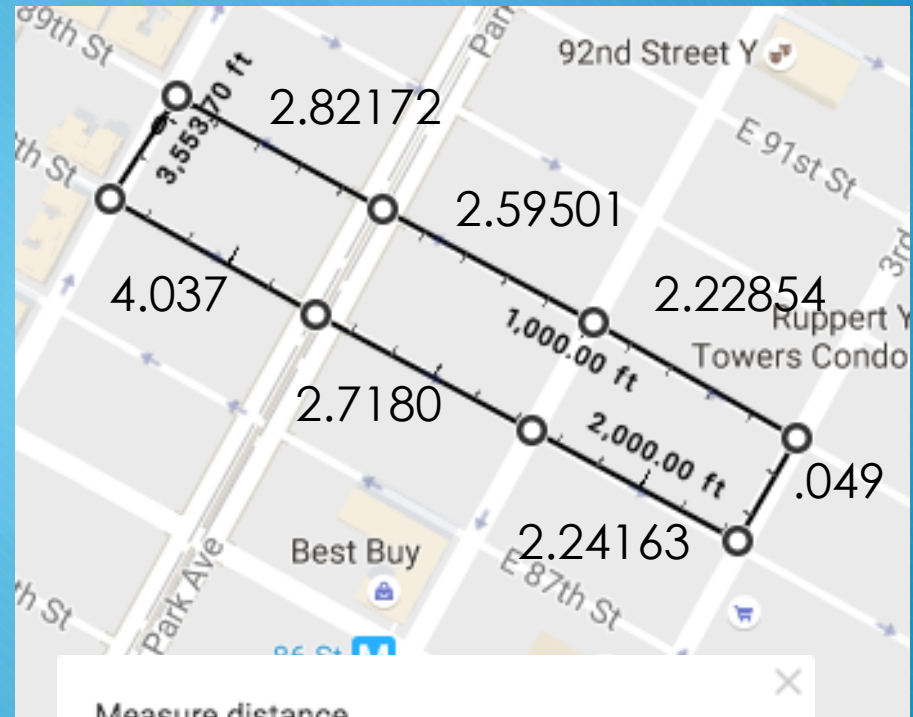
Selected routes match users' preference!

Database grid



Route 4

Goes through points 12, 13, 14,
15, 25, 24, 23, 22



Measure distance

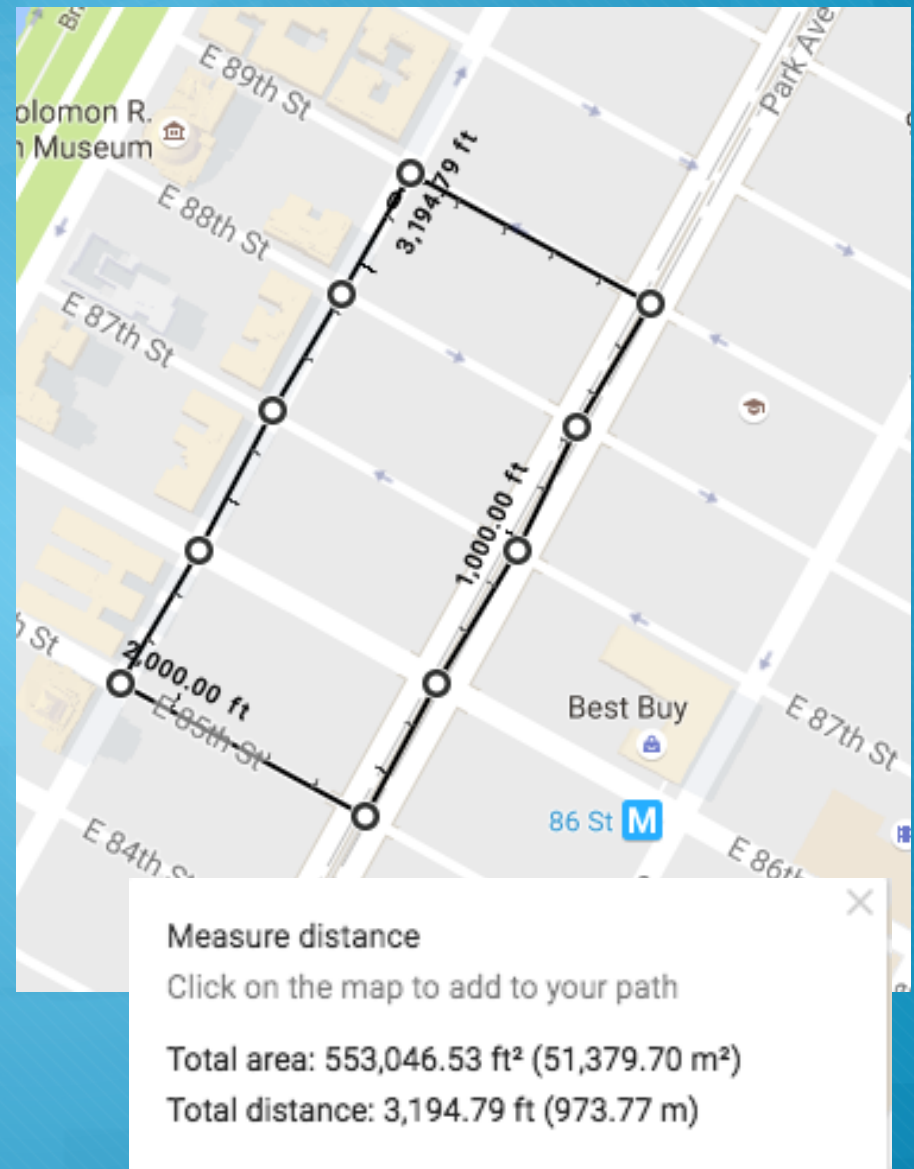
Click on the map to add to your path

Total area: 399,617.61 ft² (37,125.69 m²)

Total distance: 3,553.70 ft (1.08 km)

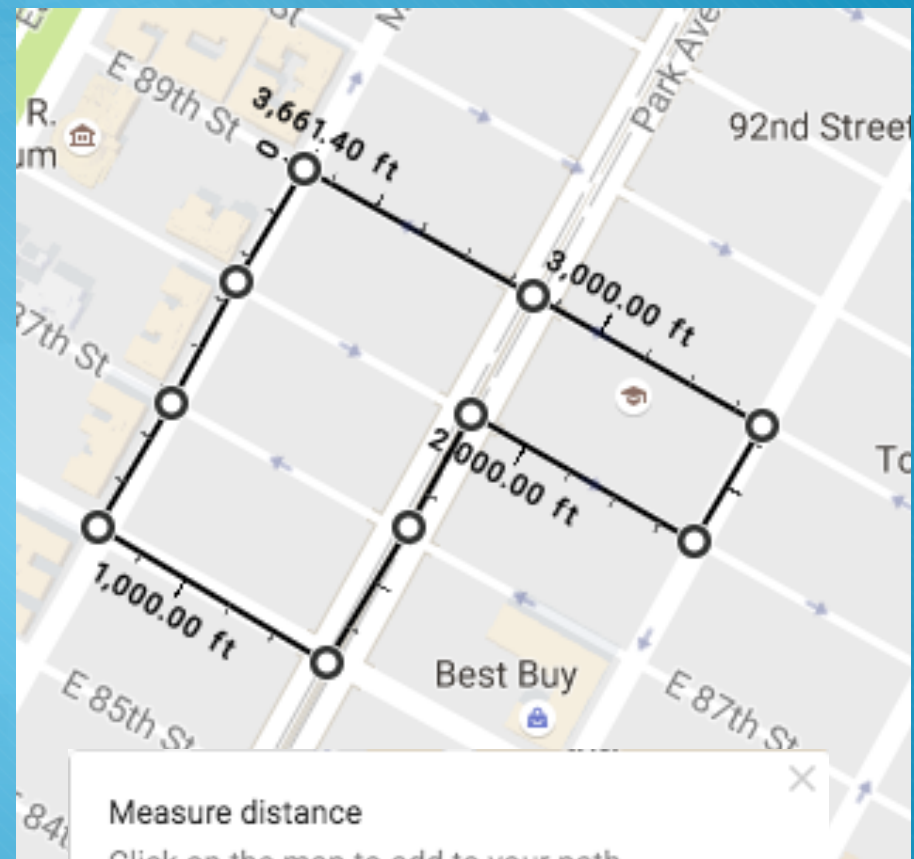
Route 6

Goes through points 12, 13, 23,
33, 43, 53, 52, 42, 32, 22, 21



Route 10

Goes through points 12, 22, 32,
42, 43, 33, 23, 24, 14, 13



Measure distance

Click on the map to add to your path

Total area: 553,114.06 ft² (51,385.98 m²)

Total distance: 3,661.40 ft (1.12 km)