

OPTIONAL TOPIC
(not on exam or assignments)

February 27,
2017
Monday ①

Finding shortest routes
on a map.

> 1 sec

Dijkstra's algorithm
 A^*

s t

US road network
24 M nodes

58 M edges

5 μ s - 63 μ s

query response
times

$O(m + n \lg n)$

↓ ↓
number number
of of
nodes nodes
edges

how to reduce runtime

- reduce m
and n

- parallelize the
search

- store some results

$$d(s, t)$$

(2)

$$\min \{ d(s, u) + d(u, v) + d(v, t) \}$$

↓
s's access
node

↓
t's
access
node

all distances for all pairs of transit nodes are precomputed and stored.

local queries: Dijkstra's, A^*

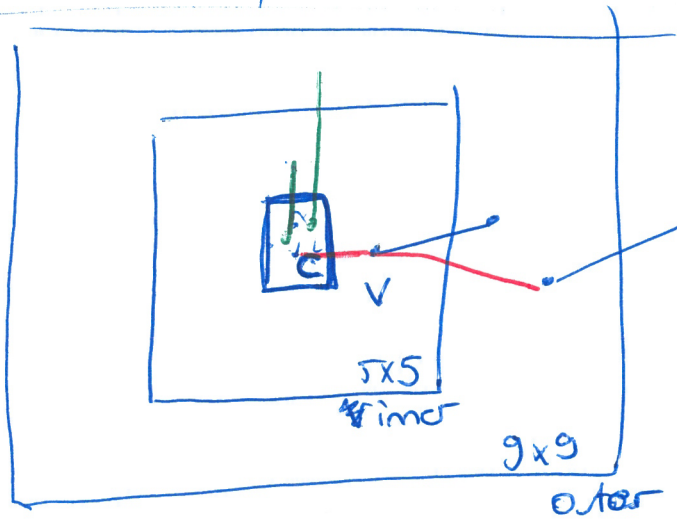
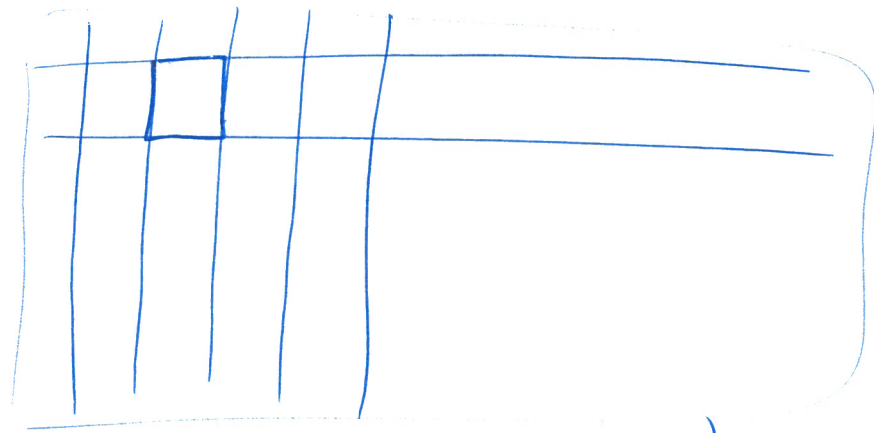
next problem

- distance from s to u
- what will be the transit nodes?
- nodes with high degree are candidates
- space set evenly

node 



3



V_C : all the nodes of these crossing edges that are in C

V_{inner} :

V_{outer}

To identify the transit nodes:
 Get the set of nodes V from V_{inner} such that there exists a shortest path from some node in V_C to some node in V_{outer} which passes through v .

tradeoff: time vs memory