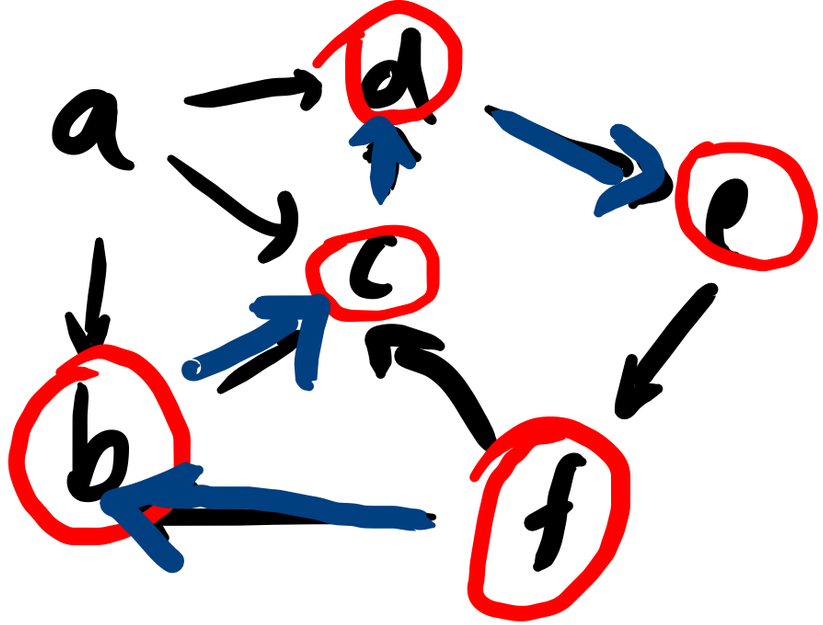


DFS starting from f

Obs. If there is a path from s to t in G, then DFS on G starting from s will find a path from s to t.

DFS

- Bottom-up analysis of graphs
  ex. computing optimal strategies for games.
- Path-finding in graphs.

dfs(G,s)

Visited[v] ← false for all v ∈ V(G)

Stack ← {s}

while Stack ≠ ϕ

v ← Pop Stack } ≤ m times

if visited[v] is false

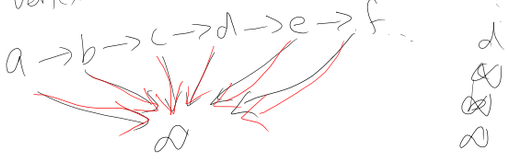
visited[v] ← true

for each v→u in G

{ if visited[u] is false (≤ m times)

Push u into stack

This loop is executed ≤ 1 times for each vertex v



claim The number of times a vertex v is pushed into the stack is at most the number of incoming edges into v.

$G = (V, E)$

$n = |V|$, $m = |E|$

Size of a graph := $n+m$

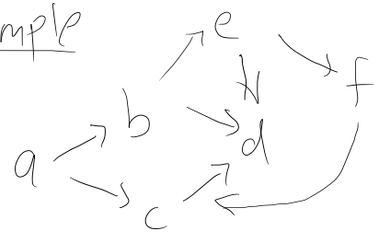
The best-case we can hope for, in general, to visit all vertices is $(n+m)$ steps

claim: DFS takes $O(n+m)$ time

Assumption: the loop for each v→u can be executed in $O(\text{outdeg}(v))$ time.

1 → 2 → 3 → 4 ... → 100

Example



dfs(G,a)

Push a

visit a

Push c

Push b

visit b

Push e

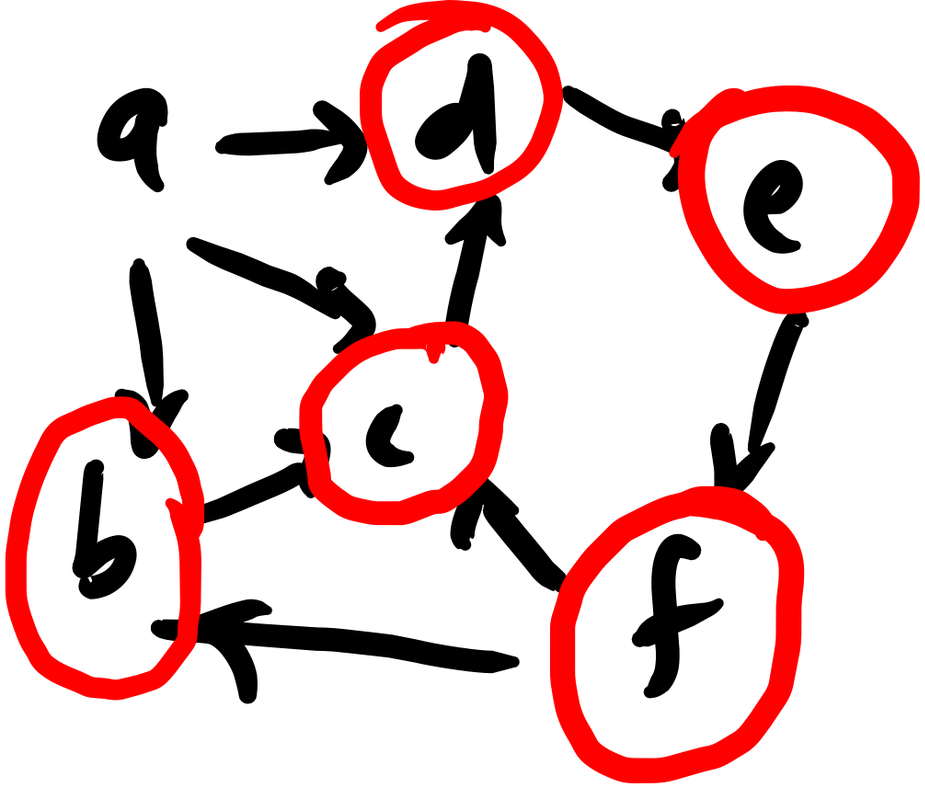
Push d

Visit d

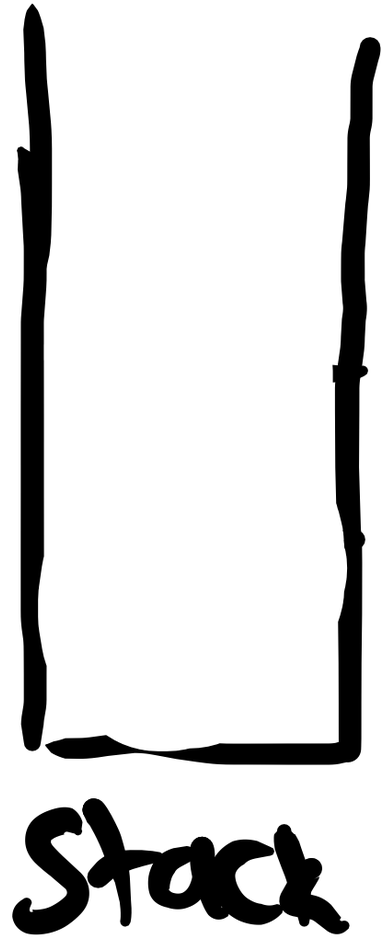
Visit e

Push f

Visit f



dfs (G, t)



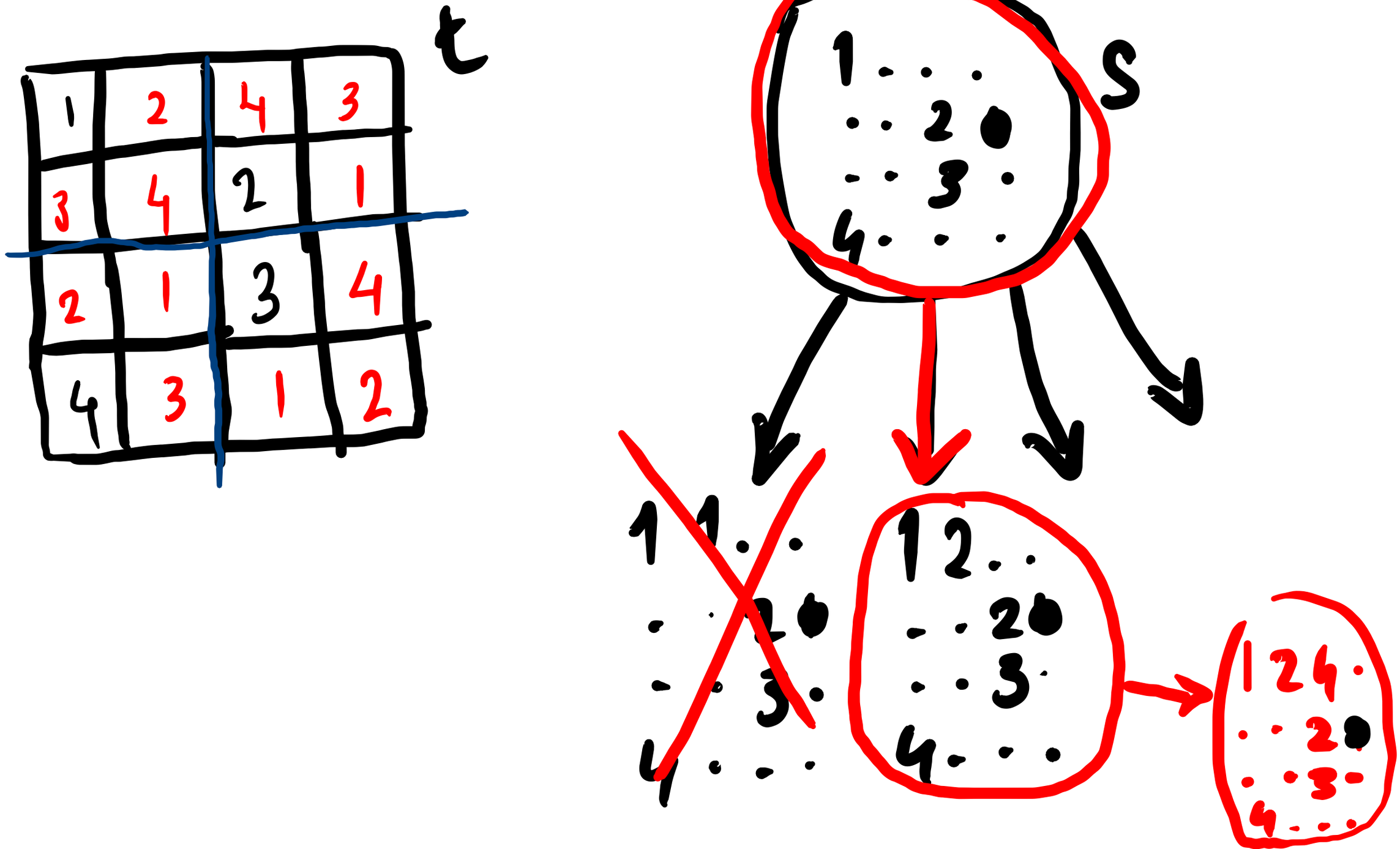
Sudoku

Path finding

(G, s, t)

Find a path in G from s to t.

What is G, s, t for Sudoku?

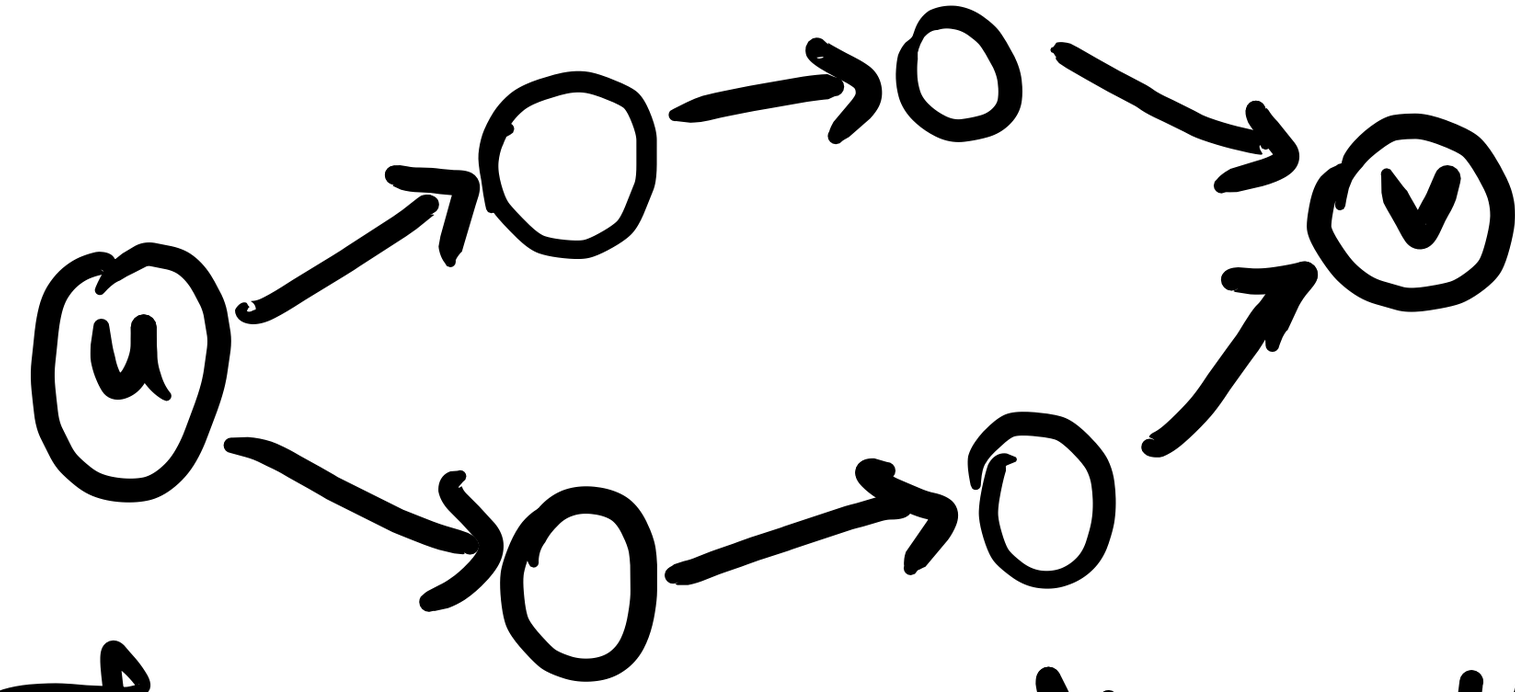

t
1 2 4 3
3 4 2 1
2 1 3 4
4 3 1 2
s
1 . . .
. . 2 .
. . 3 .
4 . . .
1 1 . .
. . 2 .
. . 3 .
4 . . .
1 2 . .
. . 2 .
. . 3 .
4 . . .
1 2 4 .
. . 2 .
. . 3 .
4 . . .

Neighbors of V

Take topmost, left most empty cell

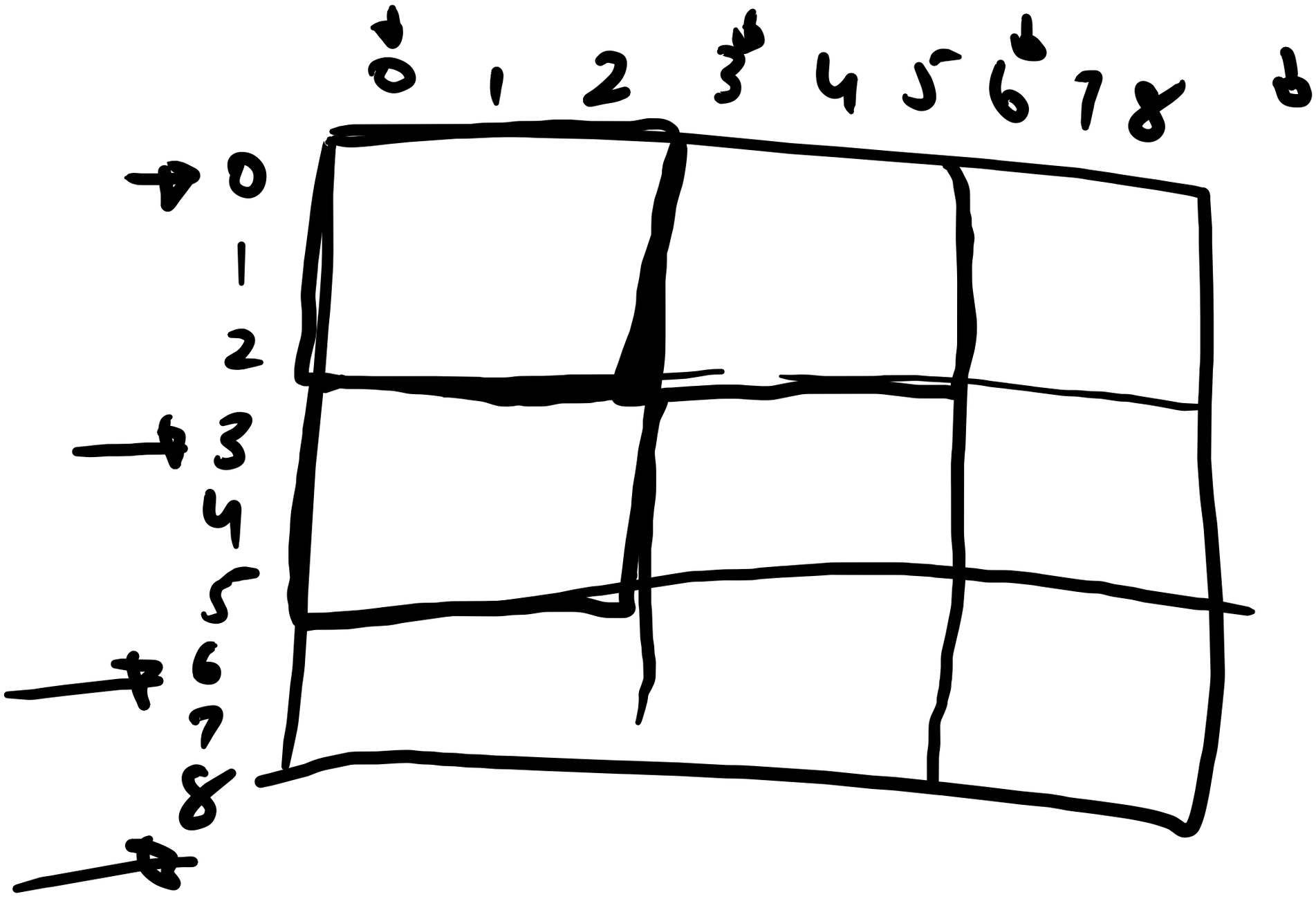
Fill all possible values on that cell.

In t++ graph



There are vertices u and v
s.t there are multiple paths from u to v.

In Sudoku graph
if there is a path from $s \rightsquigarrow v$
then this path is unique.
(Hw. Prove!)


0 1 2 3 4 5 6 7 8
0
1
2
3
4
5
6
7
8