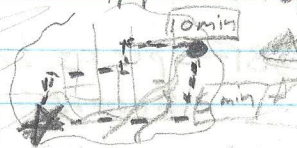


Graph Search Algorithms

3/17/14
AZ

- Shortest Path Problems - find shortest path b/w 2 nodes

Applications: robotics (commercial, domestic, search & rescue)

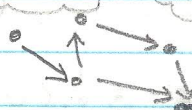


route planning (Google Maps)

Game playing (Tic-Tac-Toe, "Go")

GRAPHS - nodes & edges

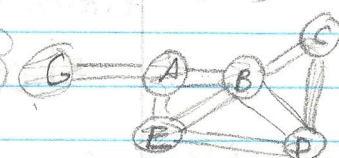
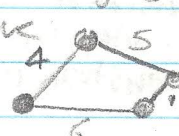
► Directed



► Undirected



*Edges can have weights



6 Nodes
8 Edges

Graph Traversal ALGORITHMS

► Specify an order to search through nodes of a graph

► start @ source of node → target node

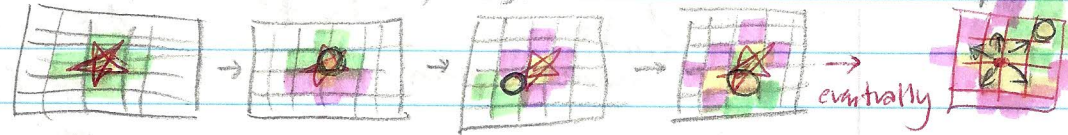
► The "Frontier" contains nodes that we've seen, but haven't

► Each iteration, frontier = 1 node &

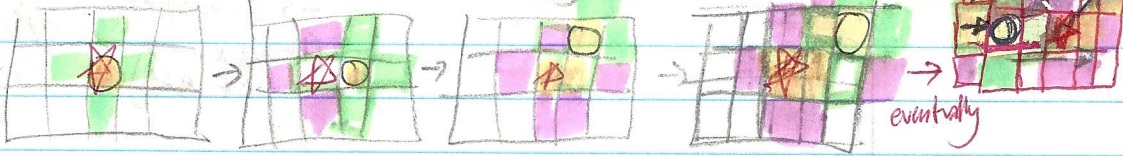
frontier += this node's neighbors

► Breadth First Search (BFS):

JS. I know, self. node I guess.



► Depth First Search (DFS):



BFS

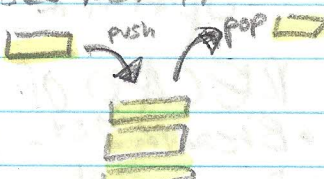
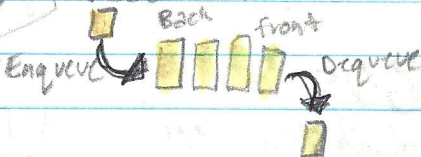
vs

DFS

uses "first in first out"

uses "last in first out"

STACK



QUEUE

Starts @
tree root, explores
neighbors 1st...

current node

next nodes to add
to frontier

Frontier

starts @
root → arbitrary node
→ root for a graph
then goes down line
A.P.A.P. (P.P.P.) before
backtracking