Picks next node off frontier...
BFS node = First in
DFS node = last in

Graph Search Algorithms Cont. Greedy Algorithm

Created 3/19/19 A2

Greedy algorithm - one that chooses the best looking option at each step


Breadth First search

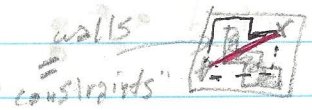

Greedy Best First search

GBF - picks "best" node according to heuristic

- heuristic - approx. measure of how close you are to target

Guide for a heuristic - easy to compute (e.g. Euclidean dist. or Manhattan dist.)

solution for easier version of problem, excluding constraints



→ approx. to shortest path

- Breadth 1st Search = optimal - always returns shortest path
  - Greedy BFS is not always optimal, but is usually faster

A* Search - developed @ Stanford to navigate robot through room of obstacles in 1968

- combines strength of BFS & GBF - fast!
  - finds shortest path!
- Calculations:
  - A* chooses the node on the frontier which minimizes:
    - looks @ nodes close to source 1st (thoroughness of BFS) → steps from source
    - + approx. steps to target → uses heuristic to prioritize nodes closer to target (speed of GBF)

Dijkstra's algorithm:
- Like BFS for weighted graphs
- optimal, but slow

} Greedy Algo

RECAP of SEARCH ALGOS

- Breadth 1st - FIFO, optimal but slow
- Depth 1st - LIFO, Not opt. & meandering
- Greedy Best 1st - goes for target, fastest but not optimal
- A* - Best of both worlds optimal & fast
- Dijkstra - Explores in order of optimal, but slow

- weighted A* - Opt. & fast for weighted edges