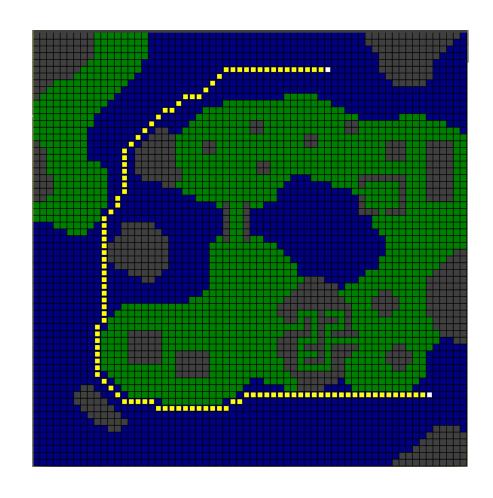


MORE LIKE GIVE ME A+

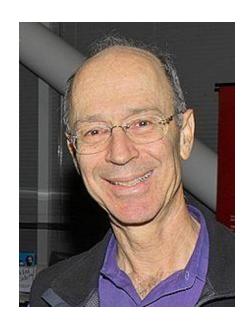
BASIC PURPOSE

A* is a path finding algorithm

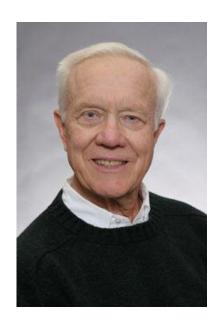
Path finding algorithms are designed to find the cheapest paths from point A to point B



A* INVENTORS



Peter E. Hart



Nils John Nilsson



Bertram Raphael

Stanford Research Institute

HOW IT WORKS

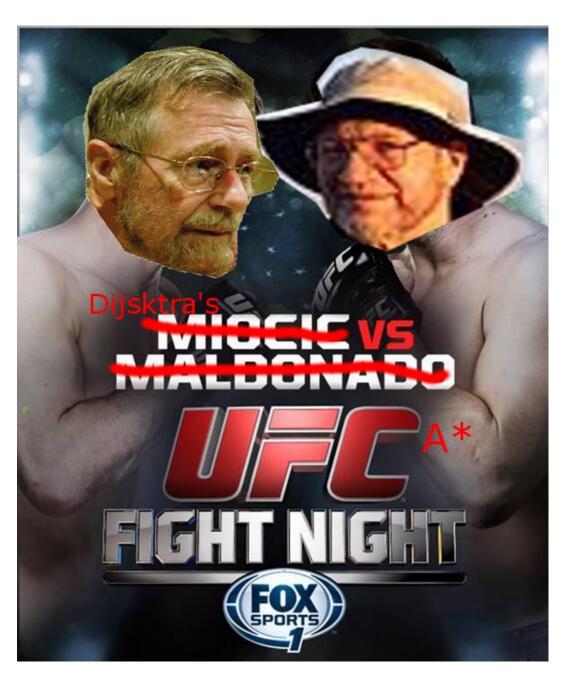
You all know **Dijkstra's**.

This works like Dijkstra's but has a Heuristic (guess of expected cost to end).

They use the heuristic plus current cost to pick next vertex.

$$cost(v_n) = p(v_n) + h(v_n)$$

WHY A*



COMPLEXITY

b = branching factor

m = path length

E= number of edges

V= number of vertices

 $O(b^m)$ perfect heuristic

O(E * log(V)) if heuristic is worst admissible (i.e. 0)

PROS AND CONS

PROS

Much faster than
Dijkstra's if heuristic is
better than cost to
compute it.

CONS

Takes a more time to code than Dijkstra's

Takes more memory than Dijkstra's

Need to find a way to create a Heuristic

If there actually is no path... It won't end

APPLICATIONS



IDA*

IDDFS

Iterative Deepening Depth First Search

DFS with an increasing limited depth

Doesn't remember past execution

IDA* AUTHOR



Richard Korf



Based off IDDFS.

In IDA* the limit is not depth but instead cost. This works well with graphs of non-uniform shape

ADI* EXAMPLE

http://qiao.github.io/PathFinding.js/visual/

QUESTIONS?

CITATIONS

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, & Clifford Stein (2009). Introduction to Algorithms

http://us.blizzard.com/static/_images/games/sc2/wallpapers/wall2/wall2-1920x1200.jpg

http://www.fightbookmma.com/tuf-brazil-3-finale-results/

http://en.wikipedia.org/wiki/Peter_E._Hart

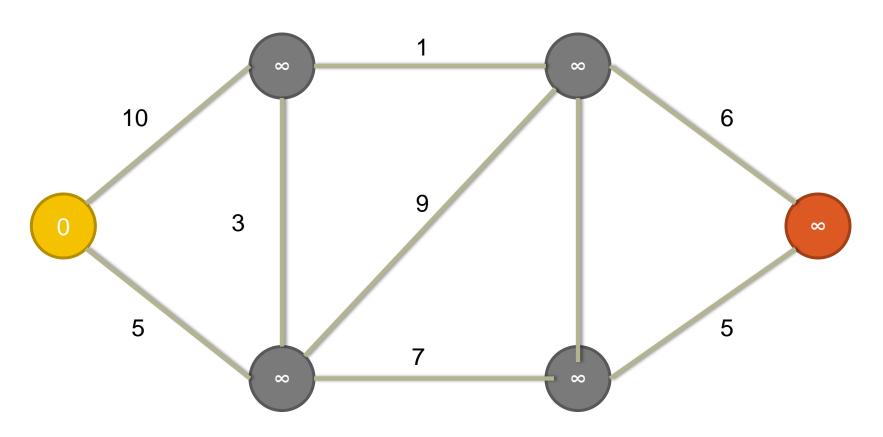
http://en.wikipedia.org/wiki/Nils_John_Nilsson

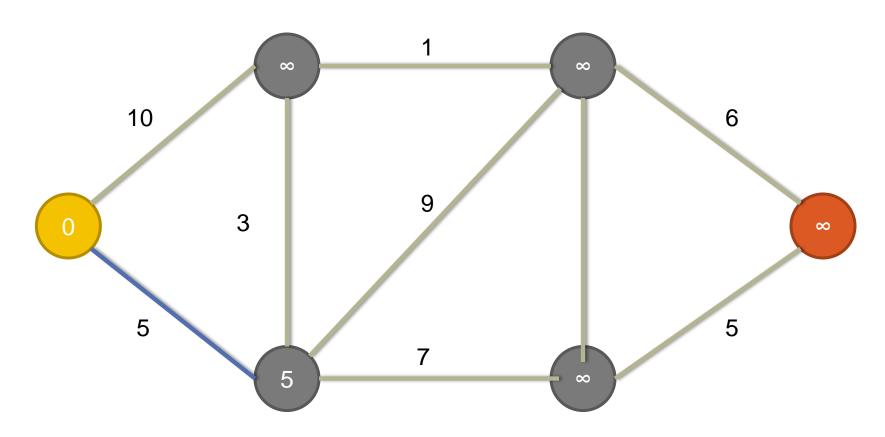
http://en.wikipedia.org/wiki/Bertram_Raphael

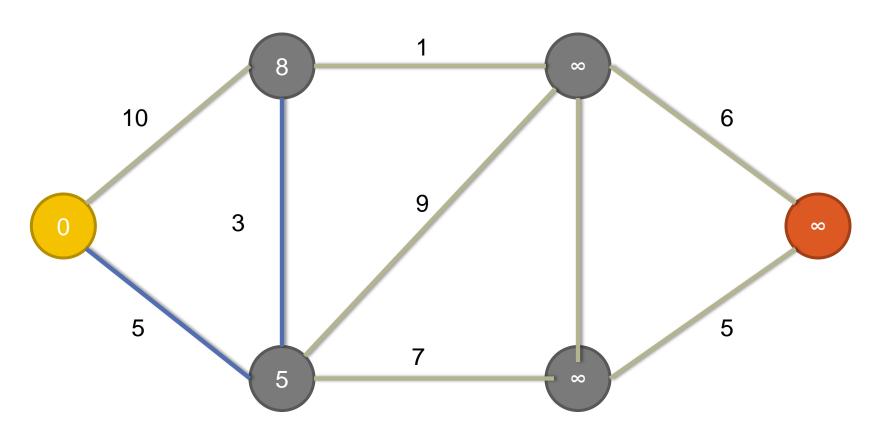
http://en.wikipedia.org/wiki/A*_search_algorithm

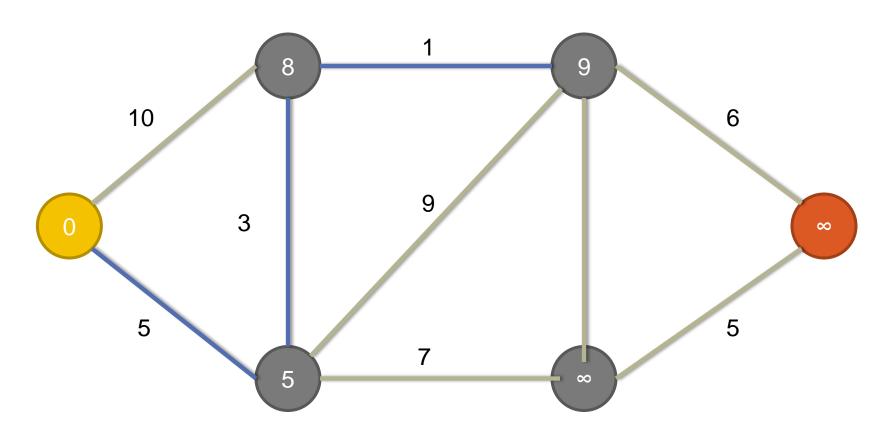
http://en.wikipedia.org/wiki/IDA*

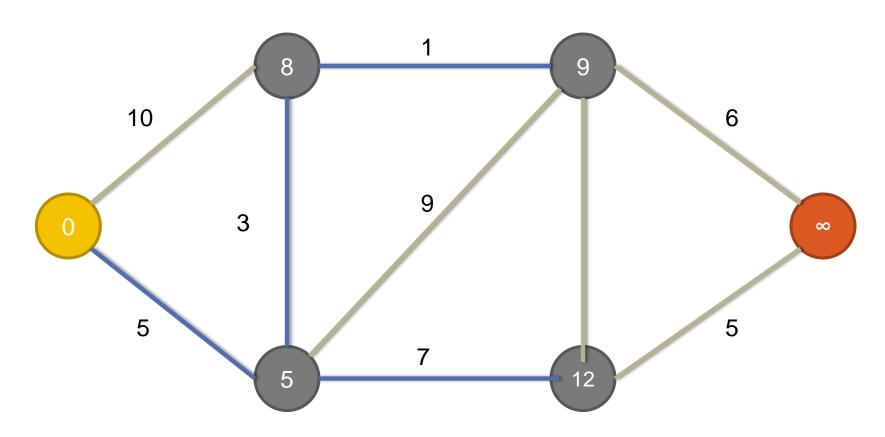
http://qiao.github.io/PathFinding.js/visual/

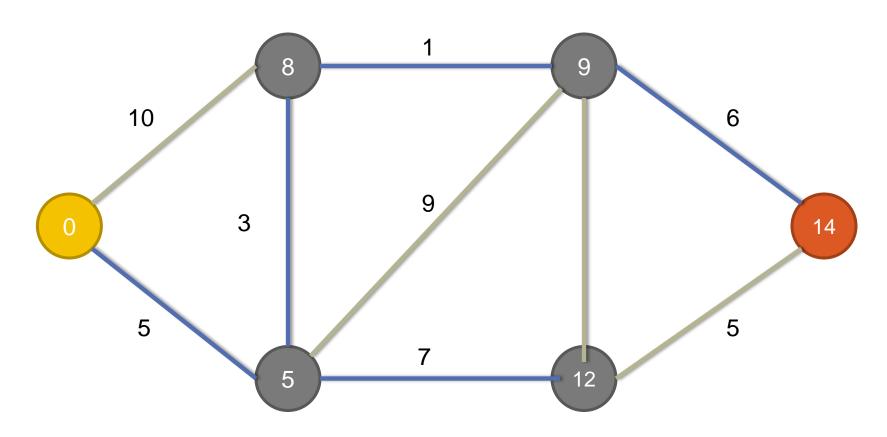


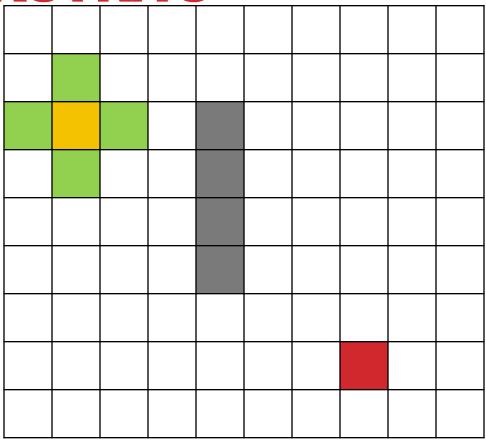


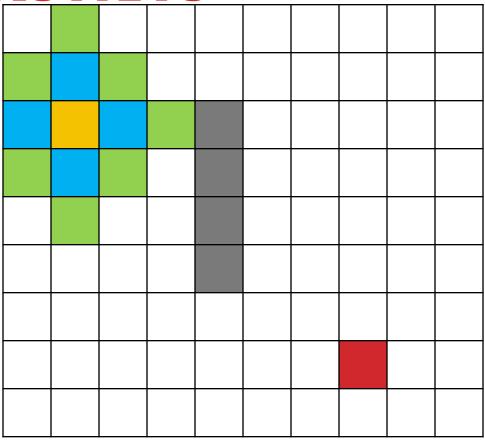


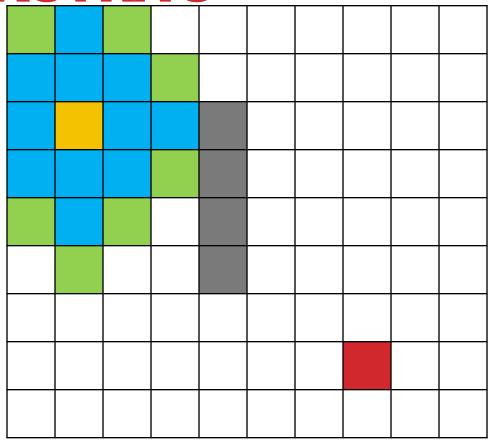


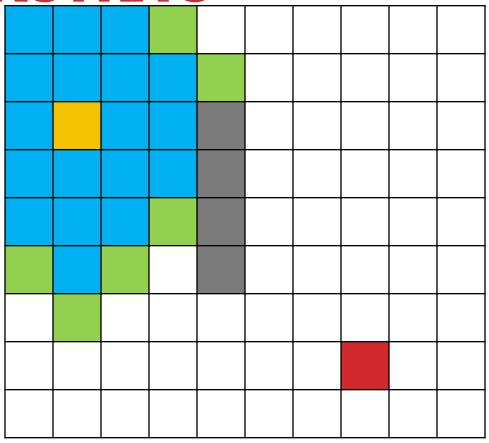


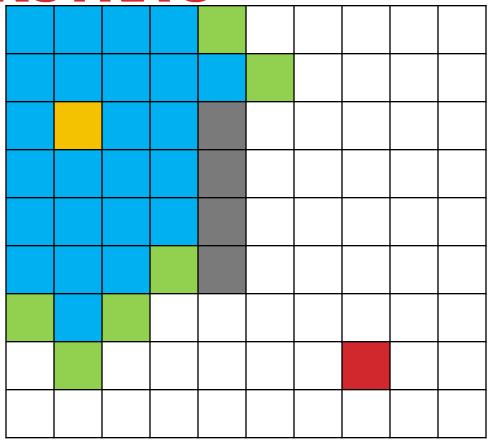


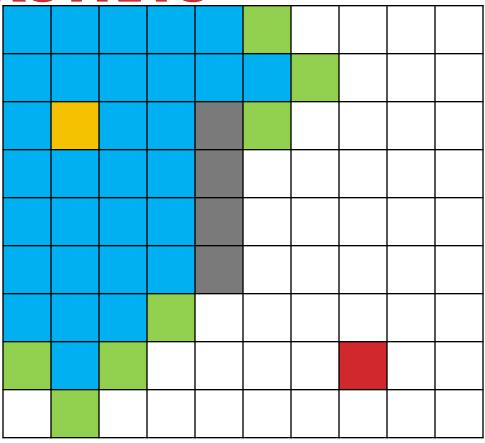


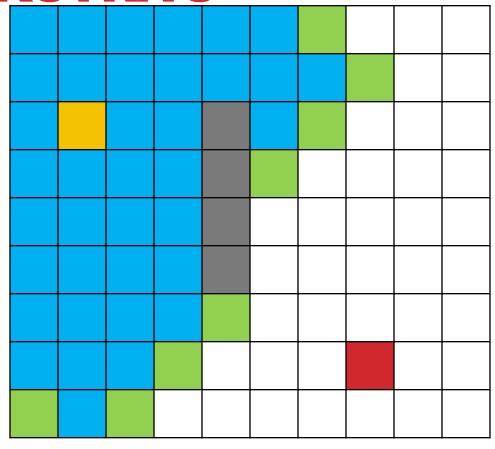


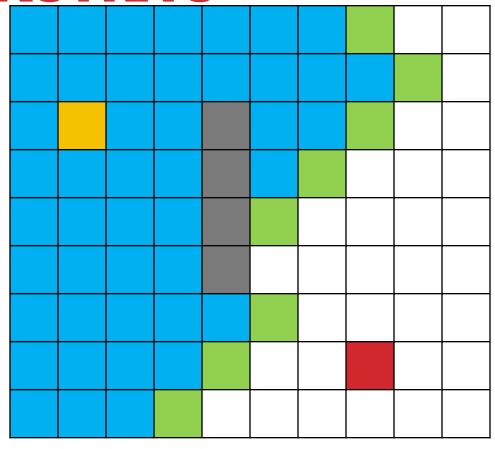


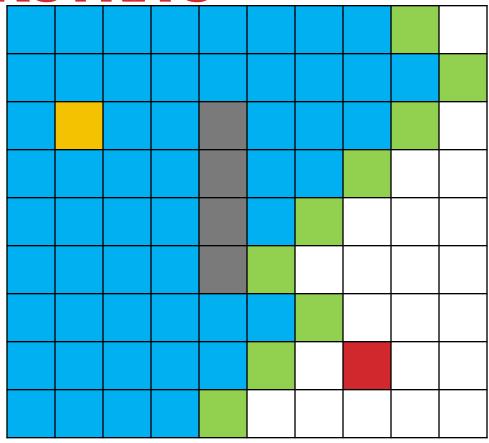


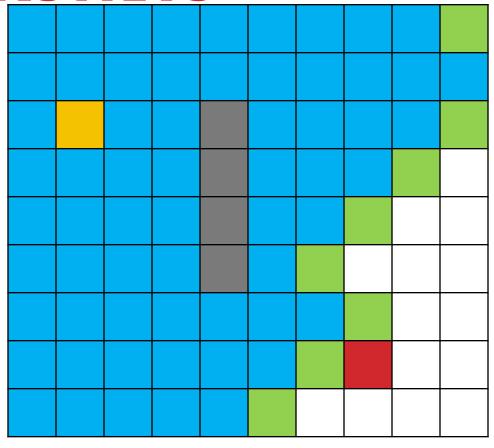


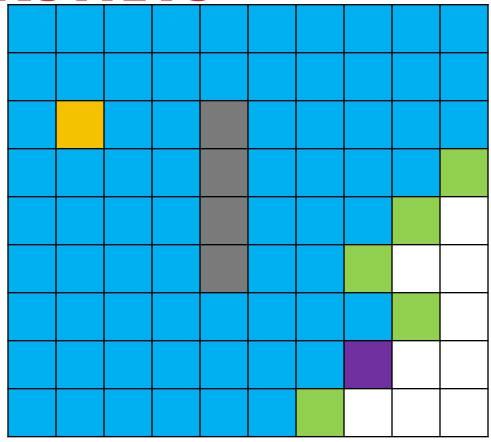






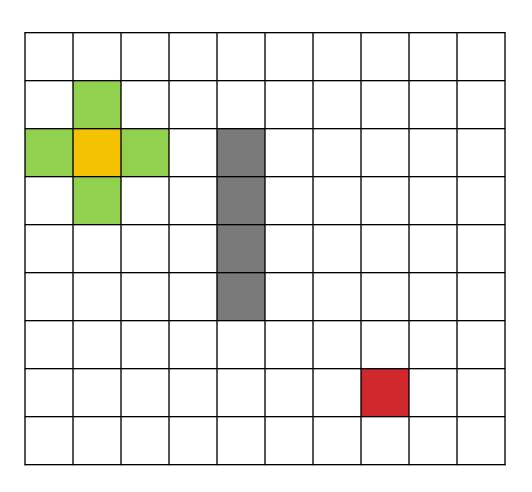




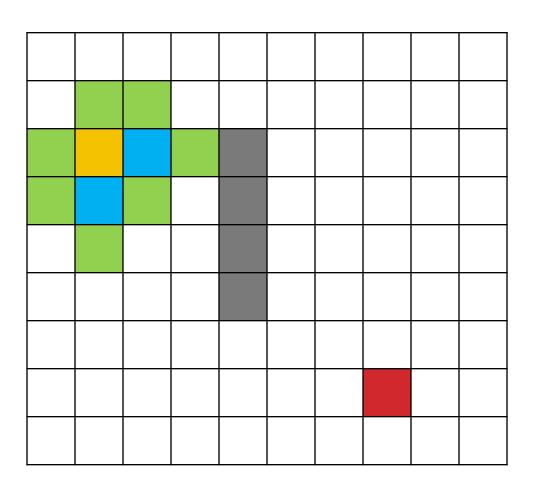




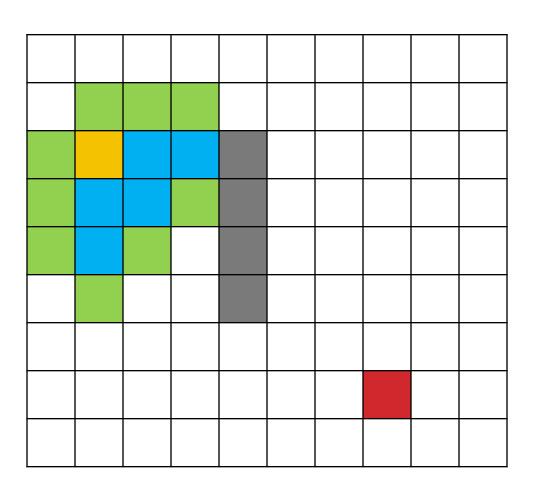




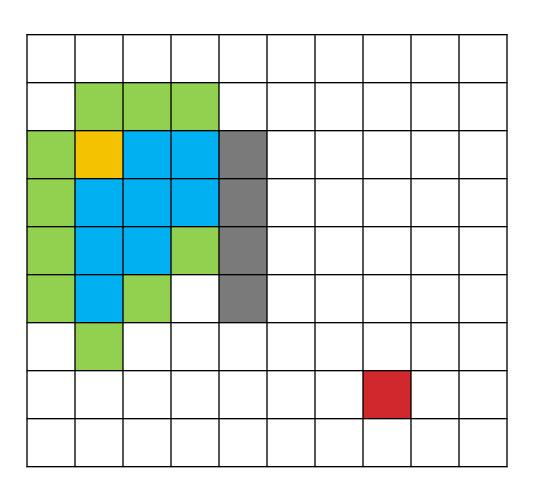




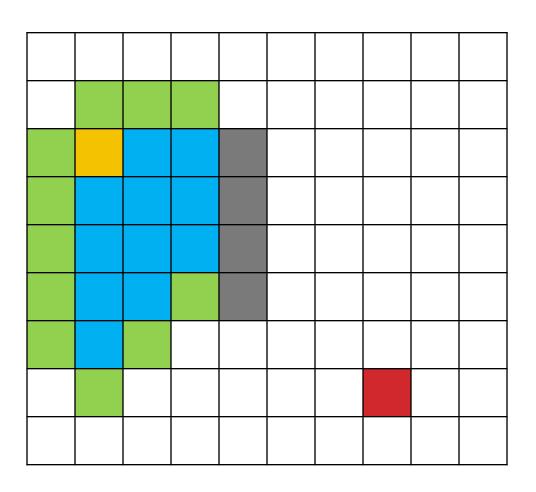




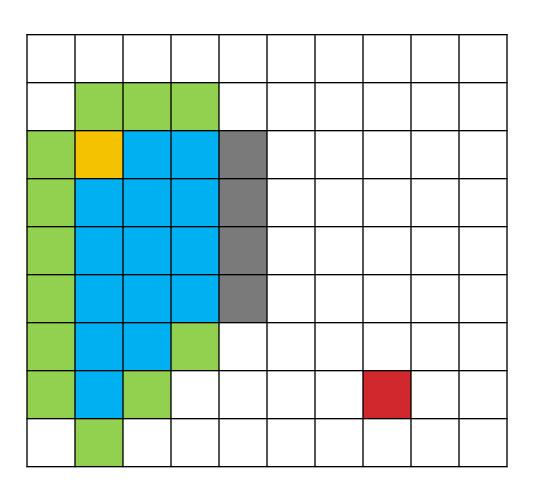




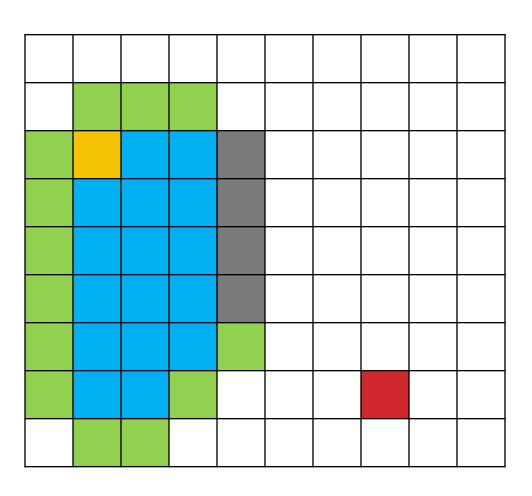




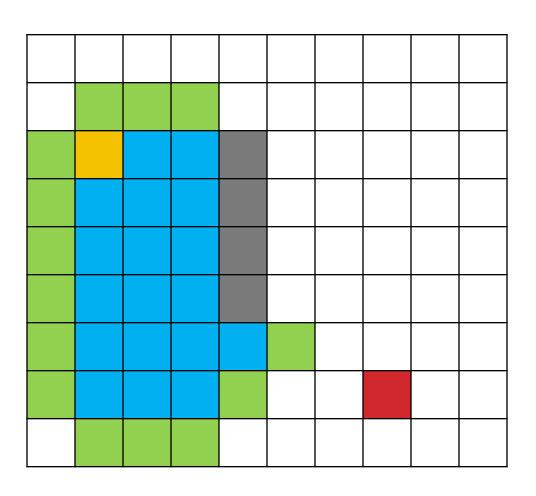




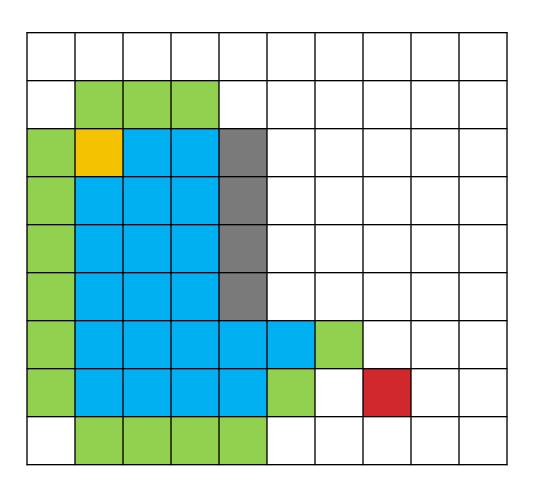




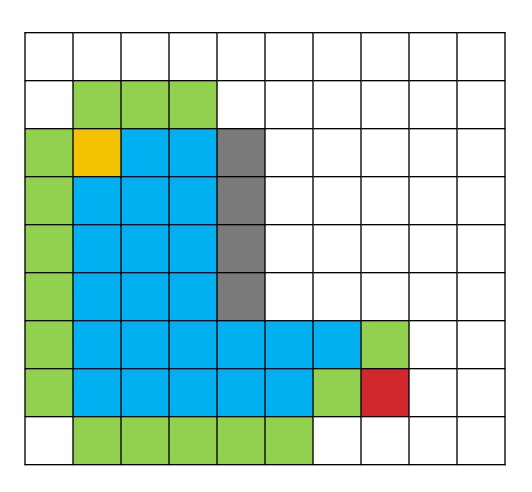




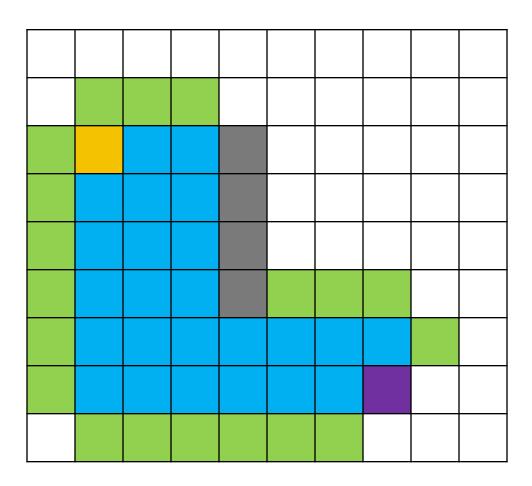




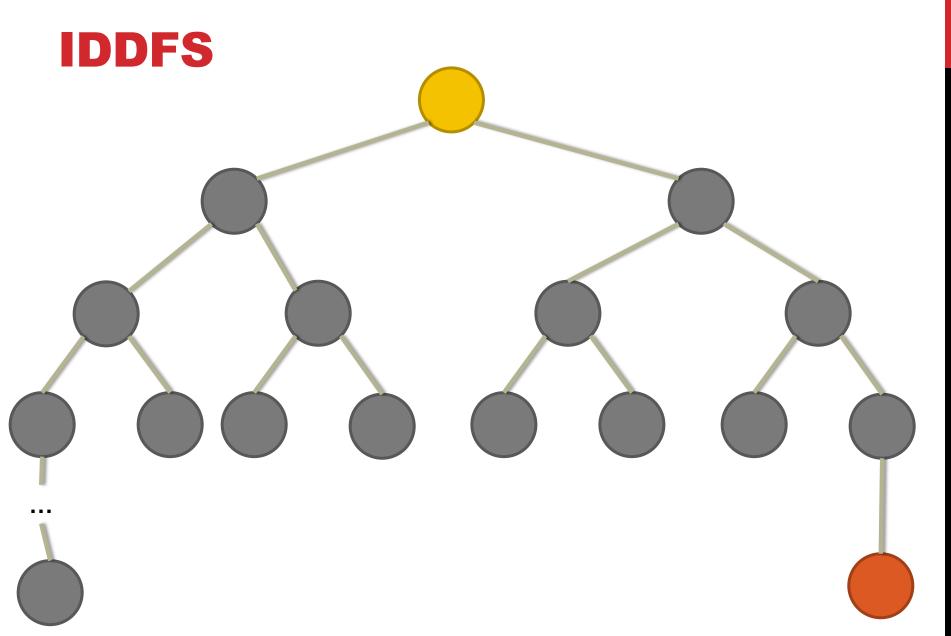


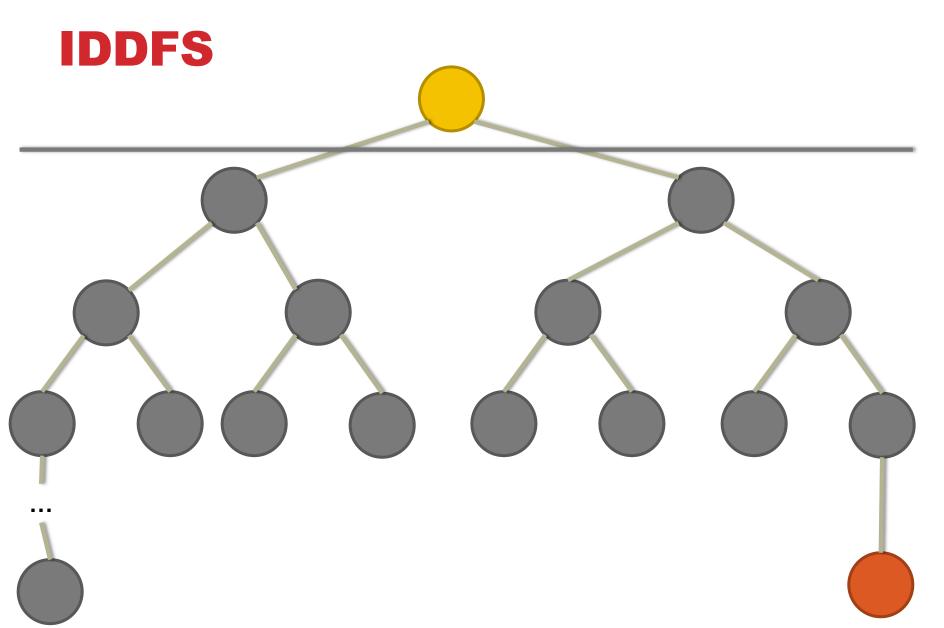


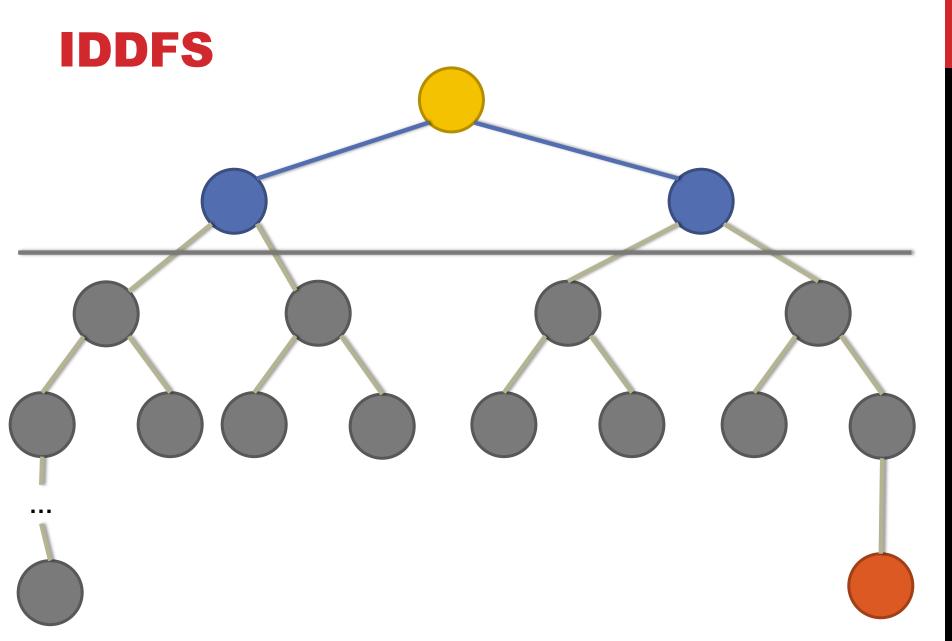


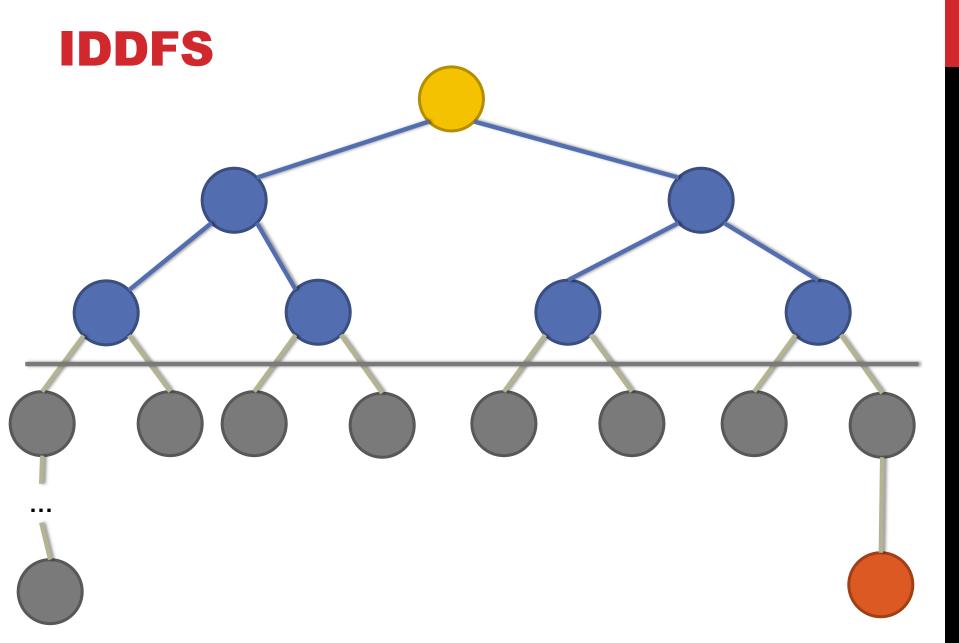




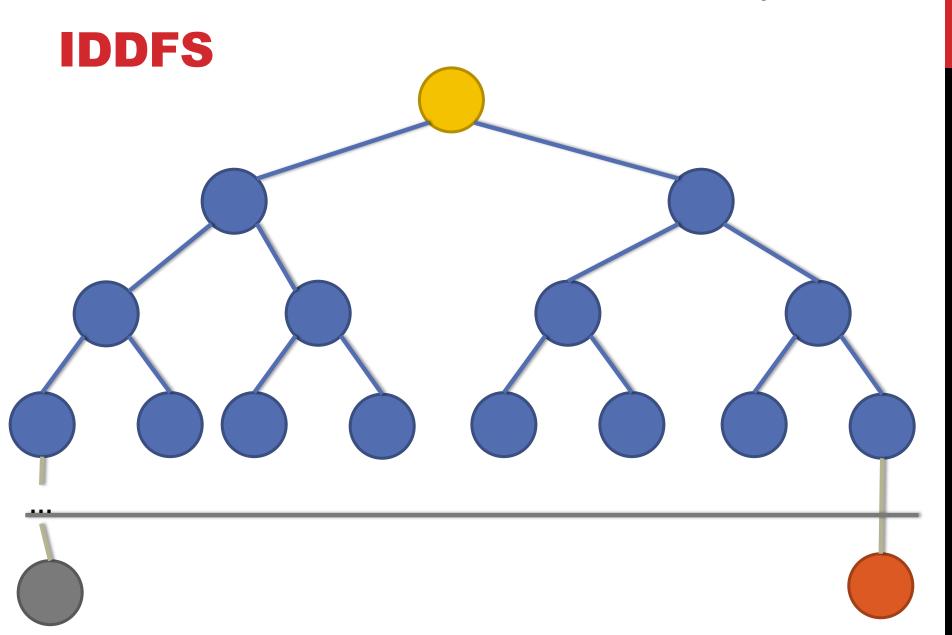








DEPTH LIMIT: 3



DEPTH LIMIT: 4

