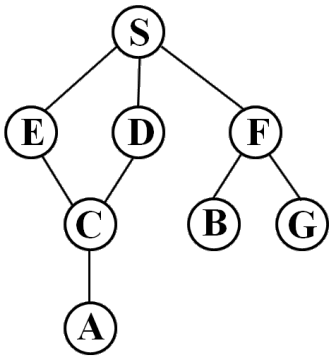
50.021 -AI

Alex

Week 08: Search

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

DFS/BFS on a graph



- start state is **S**
- goal state is **G**
- children are pushed on the fringe/frontier/agenda (same thing, different words) in **reverse** alphabetical order. If you use DFS and expand S, then you push SF first, then SE, and SD last, and that means that SD will be at the front of the fringe! This is different if you would use BFS!

  Depending on whether you run BFS or DFS the order of taking them from the fringe is different.

In the following it helps to maintain the fringe as a set of partial paths (of course you do not need if you have a memory like a poker pro), and for graph search the set of explored nodes – both at every time step.

1. DFS with Tree Search and cycle pruning

Cycle pruning means here: when taking a node off the fringe and expanding it into a set of paths, then you do not push paths onto the frontier if that path would contain a cycle. S-E-C-E has a cycle.

Task:

- What is the sequence of partial paths that are pushed on to the fringe using this strategy? Write it down.
- what is the final path found?

2. DFS with Graph Search

Task:

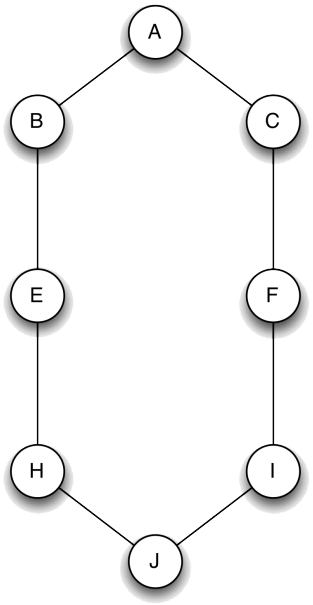
- What is the sequence of partial paths that are pushed on to the fringe using this strategy? Write it down.
- what is the final path found?

3. BFS with graph search

Task:

- What is the sequence of partial paths that are pushed on to the fringe using this strategy? Write it down.
- will you push SEC on the fringe ? yes or no?
- what is the final path found?

another graph



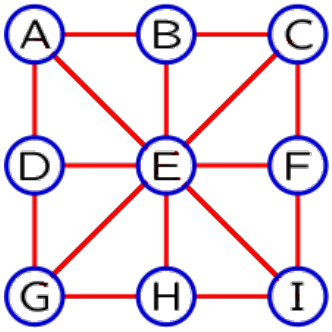
- start state is **A**

- goal state is **F**
- children are pushed on the fringe/frontier/agenda (same thing, different words) in **reverse** alphabetical order. Depending on whether you run BFS or DFS the order of taking them from the fringe is different.
- no cycles: we are not considering paths that revisit the same state (within the path)

Task: You are running tree search.

- What path will be found by DFS?
- What path will be found by BFS?

even another graph



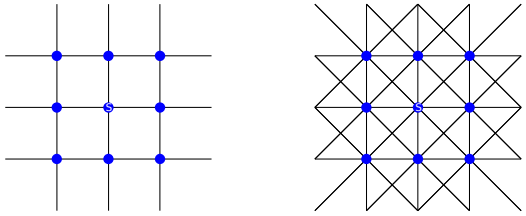
- start state is **A**
- goal state is **I**
- children are pushed on the fringe/frontier/agenda (same thing, different words) in **reverse** alphabetical order. Depending on whether you run BFS or DFS the order of taking them from the fringe is different.
- no cycles: we are not considering paths that revisit the same state (within the path)
- we are not pushing paths into the fringe/frontier/agenda if a path to that state already exists on the agenda. but it is a tree search, not a graph search, so no explored set is maintained.

Task:

- What is the final path found by a breadth-first search?
- What is the final path found by a depth-first search?

- Which of the following is guaranteed to be true about the shortest path between any two arbitrary states in the graph above?
  - The path includes state E
  - The path includes at most three states
  - The path includes at least three states

Grids



Consider a uniform grid that extends infinitely in all directions.

We start out at the grid point marked S. Assume the goal is far away and not shown in the figures.

BFS

In this problem, we will examine the effects of various pruning strategies on breadth-first search.

Answer the following questions about breadth-first search with no pruning (not even Cycle Pruning).

- In the left grid, how many paths with one edge are ever added to the agenda(frontier)?
- In the left grid, how many paths of with 2 edges are ever added to the agenda(frontier)?
- In the right grid, how many paths with one edge are ever added to the agenda(frontier)?
- In the right grid, how many paths with 2 edges are ever added to the agenda(frontier)?

Answer the following questions about breadth-first search with Cycle Pruning, which prunes from the tree paths that contain any state more than once.

- In the left grid, how many paths with one edge are ever added to the agenda(frontier)?
- In the left grid, how many paths of with 2 edges are ever added to the agenda(frontier)?
- In the right grid, how many paths with one edge are ever added to the agenda(frontier)?
- In the right grid, how many paths with 2 edges are ever added to the agenda(frontier)?

Answer the following questions about breadth-first tree search with Dynamic Programming, which would never add the same node twice to the agenda

- In the left grid, how many paths with one edge are ever added to the agenda(frontier)?
- In the left grid, how many paths of with 2 edges are ever added to the agenda(frontier)?
- In the right grid, how many paths with one edge are ever added to the agenda(frontier)?
- In the right grid, how many paths with 2 edges are ever added to the agenda(frontier)?

Uniform Cost Search

Answer the following questions about uniform-cost search with only cycle pruning in the eight-action grid (the grid on the right). Your answers should be decimal numbers accurate to within 10^1

- What is the cost of the second node expanded?
- What is the cost of the sixth node expanded?
- What is the cost of the tenth node expanded?