

ICS 171, Summer 2000: Lecture 2 Solutions

(1) A farmer has a goat, a wolf and a cabbage on the west side of a river. He wants to get all of his animals and his cabbage across the river onto the east side. The farmer has a row boat but he only has enough room for himself and one other thing. The wolf will eat the goat if they are left together alone. The goat will eat the cabbage if they are left together alone. How can the farmer get everything on the east side?

(a) Formulate this puzzle as search: i.e. give a state space representation, start state, goal state, and operators.

There are many possible ways to formulate problem as a search problem.

State Space Representation:

We can represent the states of the problem with two sets W and E. Contained in the sets are the symbols f, g, w, c representing the farmer, goat, wolf and cabbage.

Another representation could be to use an array such as $\{1,0,0,1\}$ where the position correspond to the location of the farmer, goat, wolf and cabbage. A 0 represents the west side and 1 represents the east side. Thus, for example, $\{1,0,0,1\}$ means that the farmer and the cabbage are on the east side and the goat and wolf are on the west side.

Note that we didn't explicitly represent the boat because it is not necessary. The boat must always be with the farmer, thus we can abstract away this detail.

Operators:

Move f from the E set to the W set and vice versa.

Move f and one of g,c,w from the E set to the W set and vice versa.

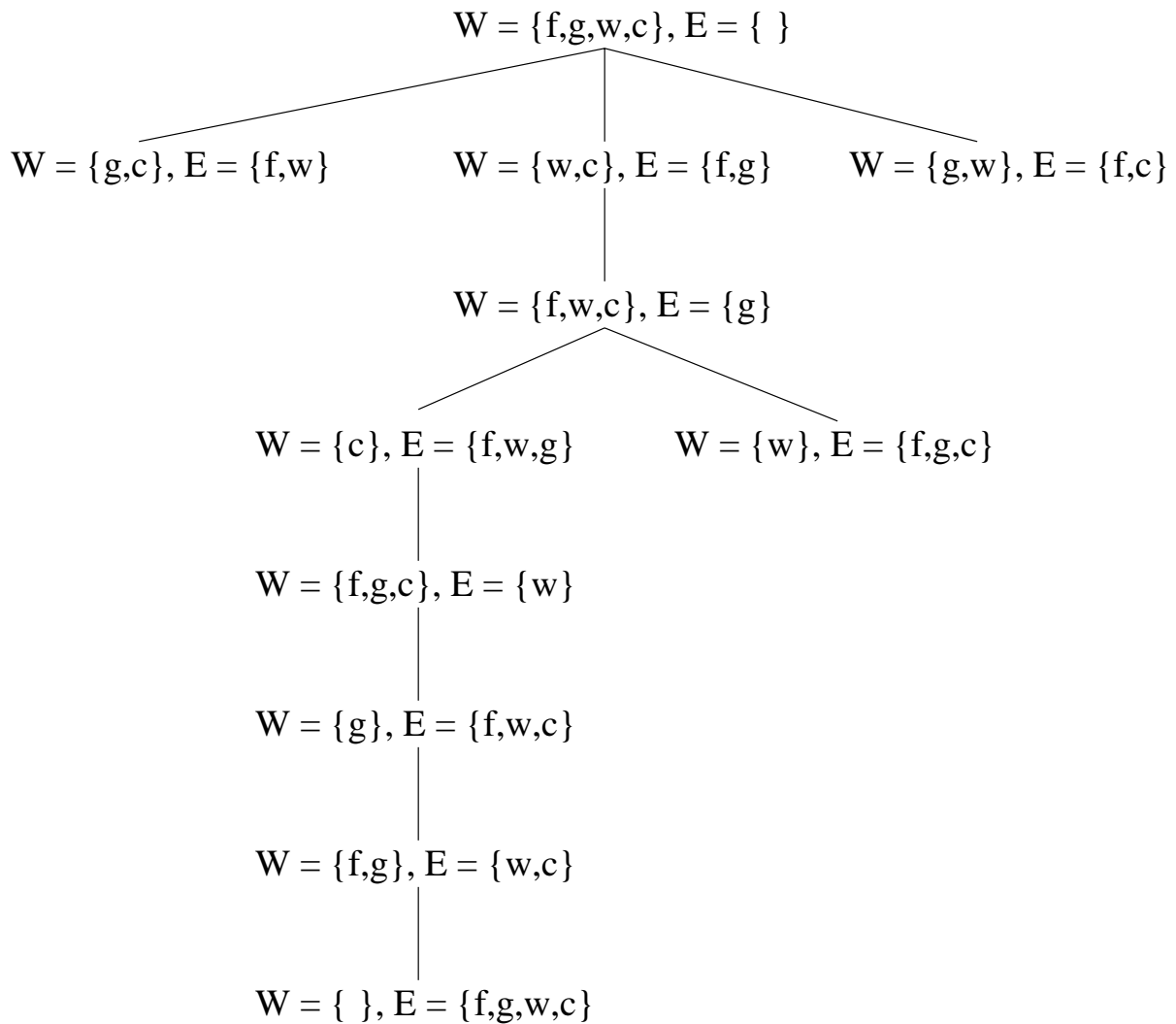
Start State:

$W = \{f,g,c,w\}$, $E = \{\emptyset\}$ (\emptyset is the empty set)

Goal State:

$W = \{\emptyset\}$, $E = \{f,g,c,w\}$

(b) Solve this problem using search (any method of your choice). Draw the search tree and show the final solution.



In words:

1. Farmer takes goat across the river. $W = \{w,c\}$, $E = \{f,g\}$
2. Farmer comes back alone. $W = \{f,w,c\}$, $E = \{g\}$
3. Farmer takes wolf across. $W = \{c\}$, $E = \{f,g,w\}$
4. Farmer comes back with goat. $W = \{f,g,c\}$, $E = \{w\}$
5. Farmer takes cabbage across. $W = \{g\}$, $E = \{f,w,c\}$
6. Farmer comes back alone. $W = \{f,g\}$, $E = \{w,c\}$
7. Farmer takes goat across. $W = \{\emptyset\}$, $E = \{f,g,w,c\}$

(2) Question 3.3 in course text (Russell & Norvig, page 87)

(solution from R & N)

(a) Initial State: looking at first page of first book.

Goal Test: Looking at entry listing the number of Mr. Jimwill Zollicoffer in Alameda.

Operators: look at new book; look at new page; look at new entry on page.

Path Cost: We can measure this in seconds – roughly 10 for a new book, 2 for a new page and 0.1 for a new entry.

(b) Everything is the same, except that the goal test is to find a listing for someone with first name Jimwill (or first initial J!) in Alameda, and to verify that it is correct (perhaps by calling the number). Note that the search strategy is very different when you know the last name, but the formulation as a search problem can be similar in each case.

(c) Initial State: You are somewhere in the Amazon jungle.

Goal Test: You are at the sea.

Operators: Follow stream upstream for a bit; follow stream downstream for a bit; go inland (in any direction) for a bit.

Path Cost: Distance travelled.

The best search strategy might be to go downstream until you reach a large river and then follow that downstream, but strategy is distinct from problem formulation.

(d) Initial State: No regions colored.

Goal Test: All regions colored, and no two adjacent regions have the same color.

Operators: Assign a color to a region.

Path Cost: Number of assignments.

(e) Initial State: As described in the text.

Goal State: Monkey has bananas.

Operators: Hop on crate; Hop off crate; Push crate from one spot to another; Walk from one spot to another; grab bananas (if standing on crate).

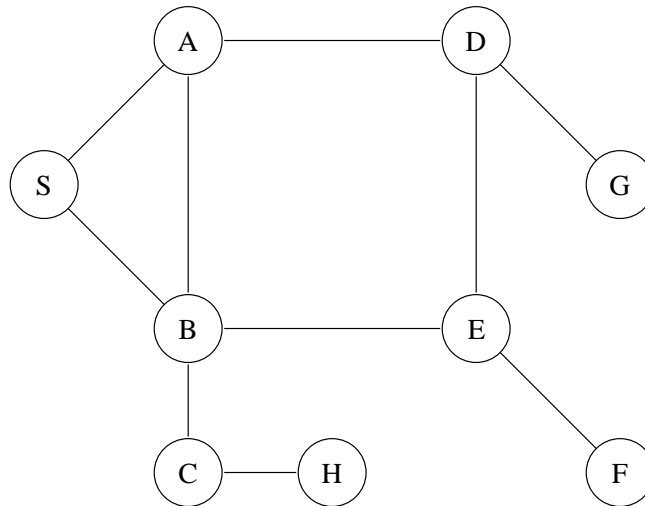
Path Cost: Number of operations.

(f) Initial State: You are at some location in the town.

Goal State: You are at the drug store.

Operators: Walk forward a bit; Turn right; Turn left; Look for Drug Store signs; (perhaps also “knock on locked door” and “ask native where Drug Store is”). Path cost: Number of steps.

(3) Consider the following graph representing the state space and operators of a navigation problem:



- S is the start state and G is the goal state.
- When placing expanded child nodes on a queue, assume that the child nodes are placed in alphabetical order (i.e. if node S is expanded the queue will be: A B).
- Assume that we never generate child nodes that appear as ancestors of the current node in the search tree.

(a) What is the order that breadth first search will expand the nodes?

S A B B D A C E C E E – G

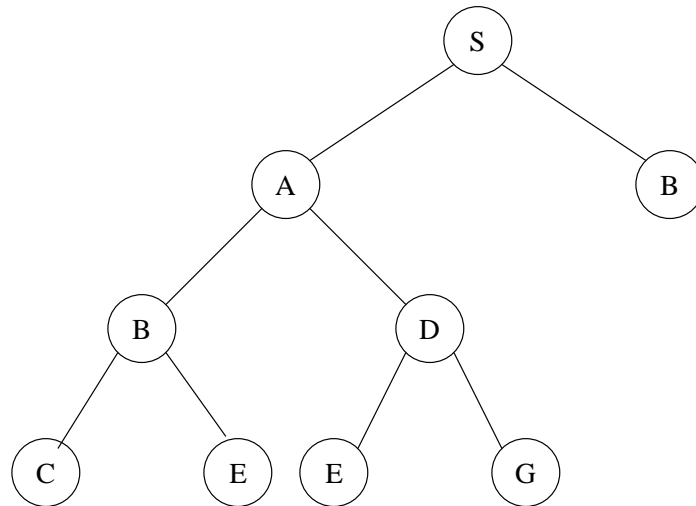
(b) What is the order that depth first search will expand the nodes?

S A B C H E D – G

(c) What is the order that iterative deepening search will expand the nodes?

S S A B S A B D – G

Note that when we are searching with a depth limit of d we do not expand nodes at that level but we do test them to see if they are goal states. For example, the diagram below shows the search tree on the third iteration (depth limit = 3). For this tree we expand nodes S A B D. Nodes C, E, E and G are tested to see if they are goal states but they are not expanded.

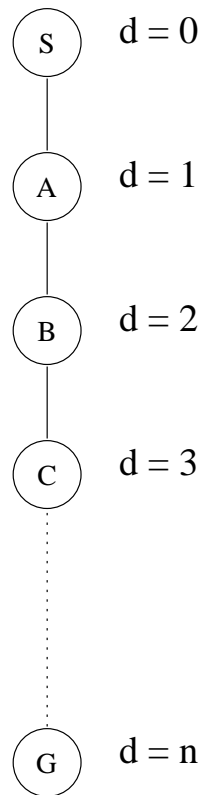


Alternatively, if we consider nodes at the depth limit as “expanded” (i.e. in the above example C, E, and G) then our solution is:

S A B S A B D B A C E S A B C E D E – G

(4) Describe a search space where iterative deepening performs much worse than depth first search. (from R&N 3.9)

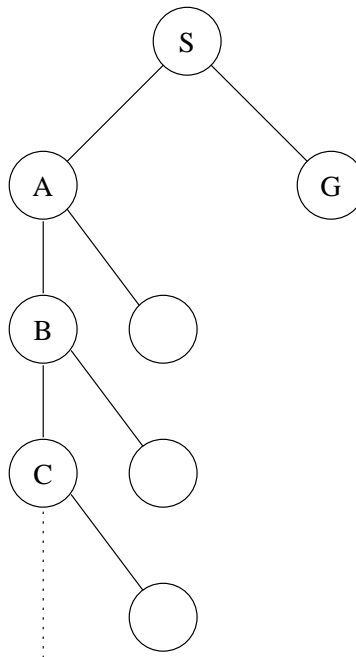
Consider a linear search tree where the branching factor is 1 and there is a goal state at depth n :



DFS search will go straight to the goal state expanding n nodes. This is order $O(n)$.

Iterative deepening will expand $1 + 2 + 3 + 4 + \dots + n$ nodes. This is order $O(n^2)$.

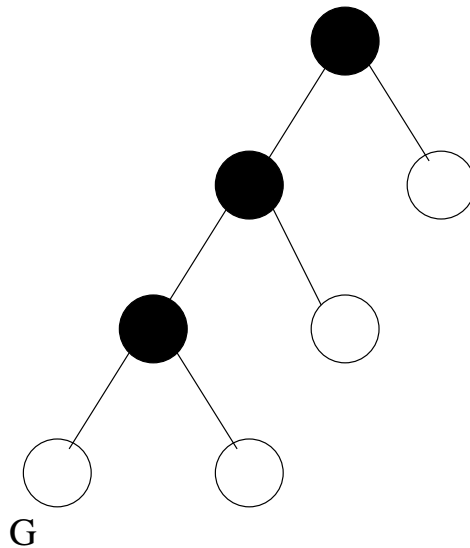
(5) Construct a search tree where it is possible that DFS will use more memory than BFS.



BFS will find the shallow goal immediately. DFS could get sidetracked and could keep exploring the left side of the tree forever generating an extremely long path (which must be stored on the Q).

(6) Suppose we have a problem space where there is a uniform branching factor b and there is a single goal node at depth m . What is the *minimum* number of nodes expanded and the storage needed for BFS and DFS? (Hint: this question asks about the best case performance of BFS and DFS).

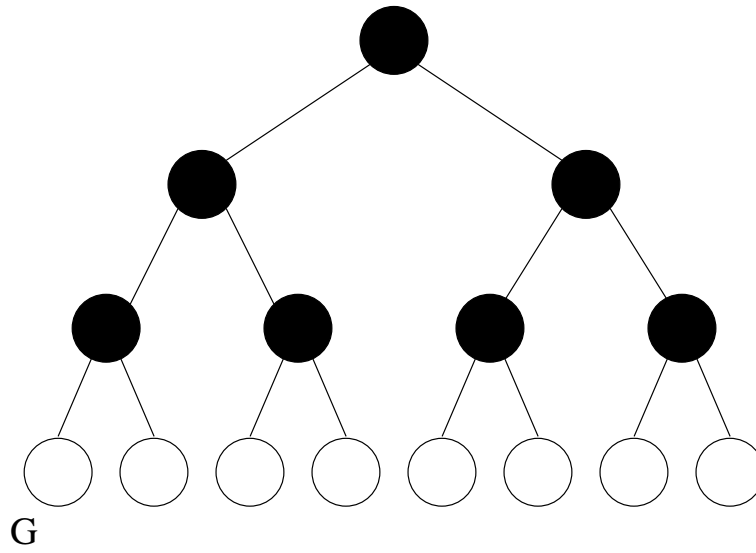
Consider the case where DFS goes straight to the goal node on level m .



Expanded nodes are shown in black, nodes that have not been expanded are shown in white. Note that the goal node G has not been expanded since in our algorithm description after we pull a node from the queue we test to see if it is a goal state first and then expand it.

In this case DFS expands only m nodes. It will need to store on the queue $(d - 1)(b - 1) + b$ nodes.

For BFS, the best case will occur when the goal is on the far left hand side (assuming a left to right ordering).



Again, expanded nodes are shown in black, nodes that have not been expanded are shown in white. In this case BFS expands $O(b^{m-1})$ nodes. BFS will need to store $O(b^m)$ nodes on the queue.