

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.01—Introduction to EECS I
Fall Semester, 2007

Week 11 NanoQuiz

1. Imagine a system with two robots moving in a two dimensional grid. A grid position is specified by (x, y) . Each robot can do the following actions: 'Stay', 'MoveUp', 'MoveDown', 'MoveLeft' and 'MoveRight' (which either stay still or move to one of four adjacent grid points). We want to search for a plan that issues commands to both robots simultaneously and gets them to meet at the same grid square in the fewest number of steps.
 - (a) Which search algorithm should we use? With or without Dynamic Programming? Very briefly, say why.

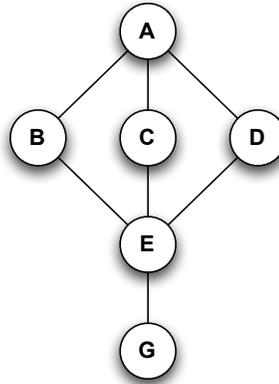
Use breadth-first since we want paths with fewest steps. With DP since there are many ways to reach the same state on a grid and we don't want to explore all those redundant paths.
 - (b) How would you represent the state where robot 1 is at location 3,4 and robot 2 is at location 5,6?

$((3,4), (5,6))$
 - (c) Briefly describe the successors of this state?

Consider the actions for each robot in combination. The successors of the first part of the state are $(3,4), (3,5), (3,3), (2,4), (4,4)$. The pattern is similar for the second component. Each robot has 5 actions and so there are a total of 25 successors.
 - (d) What would be the goal-test function?

The first component of the state should equal the second component `state[0]==state[1]`

NanoQuiz continues on reverse side.



Consider the very simple world shown in the figure. The initial state is A and the goal is G.

2. How many paths would be expanded (taken out of the agenda), by breadth-first search *without* dynamic programming?

5 paths are expanded:

- (a) *Expand (A) - visit B, C, D*
 - (b) *Expand (A B) - visit E*
 - (c) *Expand (A C) - visit E*
 - (d) *Expand (A D) - visit E*
 - (e) *Expand (A B E) - visit G, stop*
3. How many paths would be expanded (taken out of the agenda) by breadth-first search *with* dynamic programming?

5 paths are expanded:

- (a) *Expand (A) - visit B, C, D*
- (b) *Expand (A B) - visit E*
- (c) *Expand (A C) - E already visited, do not add to agenda*
- (d) *Expand (A D) - E already visited, do not add to agenda*
- (e) *Expand (A B E) - visit G, stop*