

## HW01: Search

A paper copy of your answers are due at the beginning of class on the due date. We encourage using  $\text{\LaTeX}$  to produce your writeups. See the Homework Assignments page on the class website for details.

### 1 Graph Search

Given the graph above calculate the frontier and final path returned by each of the algorithms listed below. Resolve ties where necessary using lexicographic ordering. Assume cycle checking so no duplicate paths will be placed on the frontier. Finally stop when the goal is removed from the frontier.

1. Depth First Search
2. Breadth First Search
3. Uniform Cost Search
4. Best First Search
5. A\* Search

## 2 Space Elevator

SpaceY has revolutionized transport to and from space by making the space elevator practical! It can move at speeds in the set  $V = \{v | v \text{ is a power of } 2\}$ . At each time step it can accelerate by doubling its speed, coast (not change speed), or decelerate by splitting its speed in half. In the event the elevator is not moving accelerating would increase its speed to 1 and if it has a speed of 1 then decelerating reduces its speed to 0.

Once an action is selected, the elevator agent then moves a number of squares equal to its NEW speed. For example, if the first action is to accelerate, the agent will have a speed of 1 and move up/down 1 unit. If the second action is to again accelerate, then the new speed will be 2 and the unit will move 2 more units leaving it 3 units from the starting position.

Refer to the diagram above. Assume the elevator starts on the ground, is not moving and has the goal of getting to space as quickly as possible such that it has 0 speed at the top. The answer should be in terms of the height of the elevator which is  $N$ .

1. Describe the state space, start state and goal state for this problem.

We will assume that the space elevator can only go up to space (*i.e.*, it does not have the goal of coming back down). In this case, the starting state  $S$  is simply the first square you start at. The goal state  $G$  will then be the last square on the track.

The initial state  $S$ , the set of actions  $\mathcal{A} = \{accelerate, decelerate, coast\}$ , and transition model  $\Theta$  (which dictates the affect that any particular action will have on any particular state) implicitly define the *state space*.

More concretely, at any particular space, the action that we pick (along with our current speed) will determine which square we end up in. The specific square we are currently in, combined with our current speed comprise a specific *state*, and the specific state we are in will determine what actions are available, as well as their effects.

Note, for example, that the starting state  $S$  has no speed, and begins at the very first square. Meaningful actions are *acceleration* (since *deceleration* and *coast* both result in us staying in state  $S$ ).

2. What is the maximum branching factor of this problem? Briefly justify your answer.

3. At any particular state, there are at most 3 actions you can choose from. Each action maps deterministically to exactly one “result” state. There is no way to end up in a state other than one of these three states. Thus, the branching factor is at most 3.

3. Is the number of spaces left an admissible heuristic? Explain.

Yes. Say there are  $t$  squares between our current square and the final square. No matter which value of  $t$  you pick, if the heuristic function  $h(n)$  estimates the cost to be  $t$  squares, then it *cannot* have overestimated, since it is never true that  $t > t$ .

4. State and justify an alternative, non-trivial heuristic for this state space.

If you have  $t$  remaining squares and your current speed is  $m$  squares per turn, then we can estimate the number of squares remaining to be

$$h(x) = \begin{cases} t & : m = 0 \\ \frac{t}{m} & : \text{else} \end{cases}$$

Note that your speed must always be nonnegative; since  $t/0$  is undefined, we must define the case where  $m = 0$  manually. Otherwise,  $h(n) = t/m, \forall m > 0$  will *always* be less than  $t$ , which makes this an admissible heuristic. Note that this is not necessarily a consistent heuristic.

5. Is breadth-first search guaranteed to find an optimal solution to this problem?

Yes. It is possible to construct the state space as a tree with unweighted edges, which necessarily means that BFS will find the optimal solution. For example, consider that an optimal solution will find the *smallest number of moves* between states  $S$  and  $G$ . If every state in the state space represents a current square coupled with a current speed, then the optimal solution will also be the most shallow solution, since the edges will not have weights.

6. Is depth-first search complete for this problem?

Yes. There are a finite (though possible *huge*) number of states, which means that DFS will find *one* of them.

### 3 Upgraded Space Elevator

*This problem is for grad students only.* The SpaceY elevator has been upgraded! It can accelerate by unit increases in its speed, doubling its speed as before, or even tripling its speed.

1. What impact does this have on the state space? I.e. does this change the state space description and/or the maximum branching factor?
2. State and justify an alternative, non-trivial heuristic for this state space.
3. Diagram the first three levels of the search tree for this problem.