Artificial Intelligence *Alex Clemmer u0458675*

# HW01: Search

A paper copy of your answers are due at the beginning of class on the due date. We encourage using 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.

This question does not specifically ask for a particular answer; I will thus assume you want me to show you the frontier at each step, and then the final result at the end.

1. Depth First Search

   Start
   AB
   BC
   BD
   B Goal
   G ← D ← C ← A ← Start

2. Breadth First Search

   Start
   AB
   BC
   CD
   G ← D ← B ← Start

3. Uniform Cost Search

```
Start
AB
BC
CD
D
Goal ← D ← B ← Start
```

4. Best First Search

```
Start
AB
ACD
AC
Goal ← D ← B ← Start
```

5. A* Search

```
Start
AB
ACD
AC
Goal ← D ← B ← Start
```

## 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 $V = \{v|v$ 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 the transition model $\boldsymbol{\Theta}$ (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 in the case of the starting state $S$, we have no speed, and we are 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.

No. The fact that $t$ squares exist between your current space and the the final space $G$ does not imply that you need to make $t$ moves to get there (remember that we are using unweighted edges), especially since you can choose to increase your speed. Thus $t$ can be an overestimate.

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 roughly the number of moves it will take to reach the goal at the current speed.

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

The typical case (*i.e.*, the case that we are moving toward the goal, so that $m > 0$) is represented by $h(n) = t/m$. Since we must necessarily decelerate at some point, this will always underestimate the true cost, which will potentially more moves as we reduce our speed.

The problem mentions that you can go "up/down", which means that our estimate must account for the possibility that we are moving in the opposite direction of the goal. We define this to be some constant $c$ (since making it $\infty$ or $t$ would make it a non-admissable heuristic).

If we are not moving (*i.e.*, if $m = 0$), our typical estimate of distance would be $t/0$, which is undefined. Thus, we define the case where $m = 0$ manually to be $c$.

This estimate will *always* be less than or equal to the actual $t$, which makes this an admissible heuristic.

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

Yes. The search problem as we have formulated it does not have weighted edges, which necessarily means that BFS will find the optimal solution. That is, observe that an optimal solution will be the *smallest number of moves* between states $S$ and $G$.

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

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

# 3 Upgraded Space Elevator

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