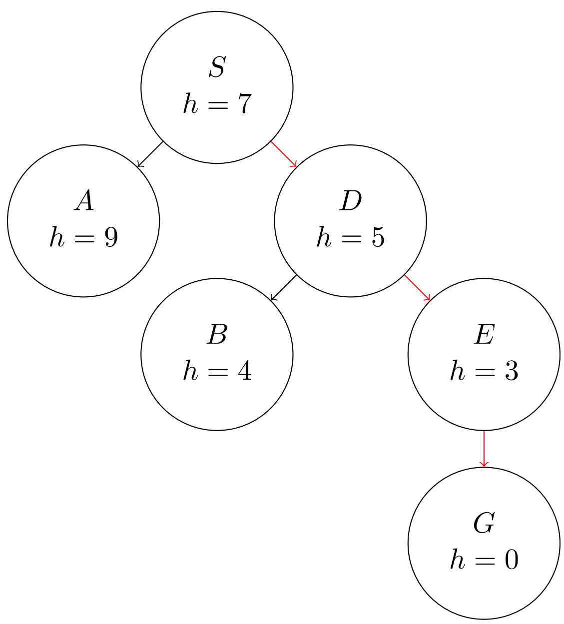
Harvard CS 182: Artificial Intelligence September 14, 2022

Section 2 Solutions: Informed Search and Motion Planning

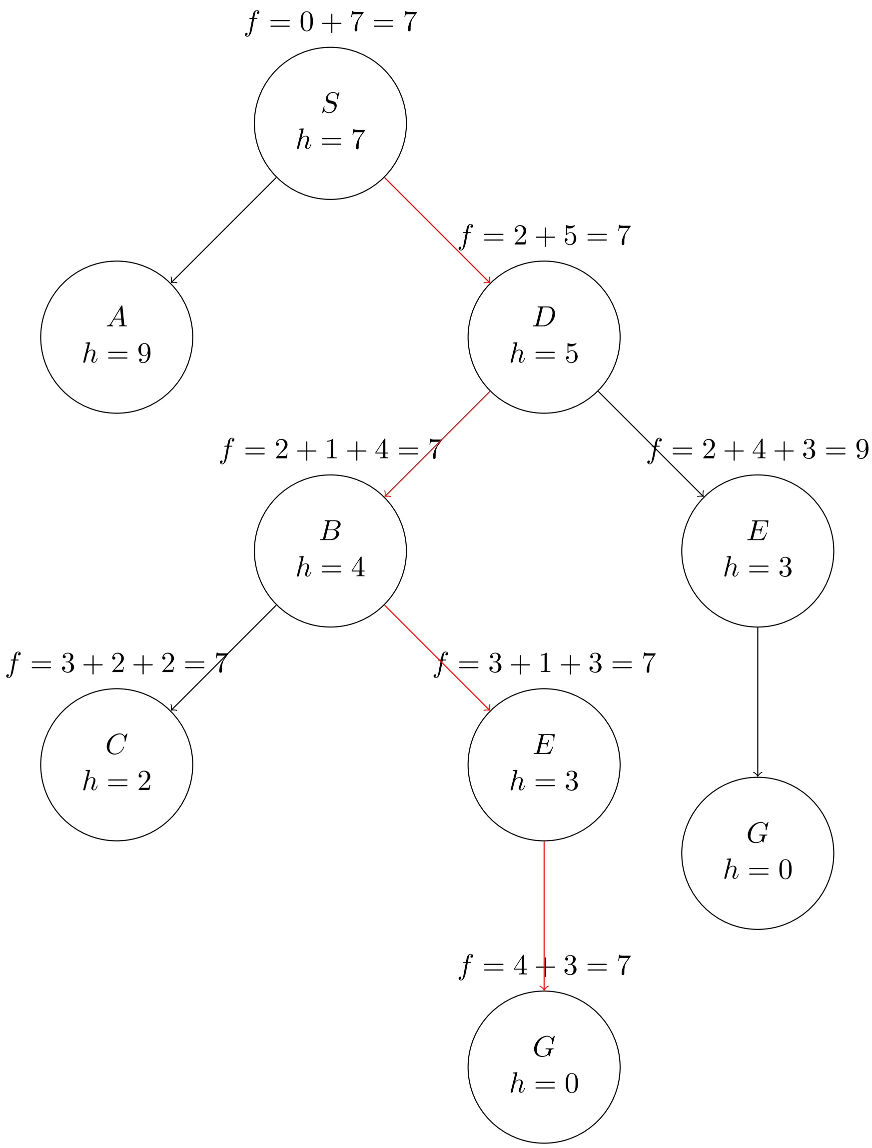
Lecturer: Stephanie Gil *Authors: Lauren Cooke and Chris Lee*

Solution 1 *There are many correct solutions to this problem. Rather than needing to find the absolute best restaurant on the street, our heuristic will be good enough by finding a delicious restaurant that you will enjoy. One shortcut we could employ would be checking online reviews (yelp, google, etc), and finding a restaurant with 4 or more stars with at least 100 reviews. Another heuristic would be to ask at least 20 locals what their favorite restaurant is on the street and going to eat at the most popular choice.*

Solution 2 *Beginning our path at our start state* s*, we can either go to node* A *or* D*. We select* D *given its lower cost of* 5 *versus* A*'s* 9*. From* D*, we can either travel to* B *or* E*. We choose* E *given its lower cost of* 3 *versus* B*'s* 4*. From* E *we can then access the goal state* G*. We can also represent our solution in a tree, where red edges indicate the desired path.*

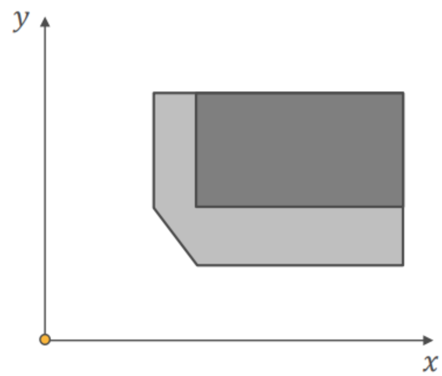


Solution 3 *To solve, we compute $f(x) = g(x) + h(x)$ for all nodes and choose the node with the lowest sum.*



Solution 4 *The main challenge is that the nodes are infinitesimally close together, meaning that travelling between any two points in space that are a non-infinitesimal positive difference from each other requires traversing through infinitely many states. For example, suppose we must travel from point $(0, 0)$ to $(0, 1)$. Then suppose we have some large integer n and define $\varepsilon = \frac{1}{n}$. Along the optimal path on the x-axis, we must also encounter the nodes at $(0, \varepsilon), (0, 2\varepsilon), \cdots (0, (n-1)\varepsilon)$ before we reach the goal state. Since n can be arbitrarily large, the number of states encountered also becomes arbitrarily large, and the algorithm is not guaranteed to terminate.*

Solution 5 *The configuration space is shown below, with the gray areas representing inaccessible nodes because of the obstacle. The dark gray area is where the bottom left corner of the triangle is directly in the obstacle, and the light gray area is where the corner is not within the obstacle but where the triangle intersects the obstacle somewhere else.*



Solution 6 *The visibility graph formed using the obstacle on the left will lead to the optimal path. Let T be the red triangle, and note that the visibility graph would give the optimal solution if the obstacle was T. To see why the optimal path cannot include any points inside T, we can use a similar argument as was used in class to prove the optimality of the visibility graph for spaces with polygonal obstacles. You will formally prove this in PSet 1 Problem 4.*

The visibility graph formed using the obstacle on the right will not necessarily give the optimal path – in fact, it is not guaranteed to return any path at all, since the vertices of the equilateral triangle are no longer visible from each other. If the straight line from the start to the goal passes through this obstacle, then the visibility graph will have no edges.