

Intelligent Systems

Exercise 5. Informed Search



Exercise description

The objective of this exercise is to apply the different heuristic search algorithms to find the route that would take an agent from an initial state to a goal state, and practice the designing of heuristics.

Team members

Write the student id, name, and campus of each member in a different line.

1: Carlos Hinojosa A01137566, Campus Monterrey

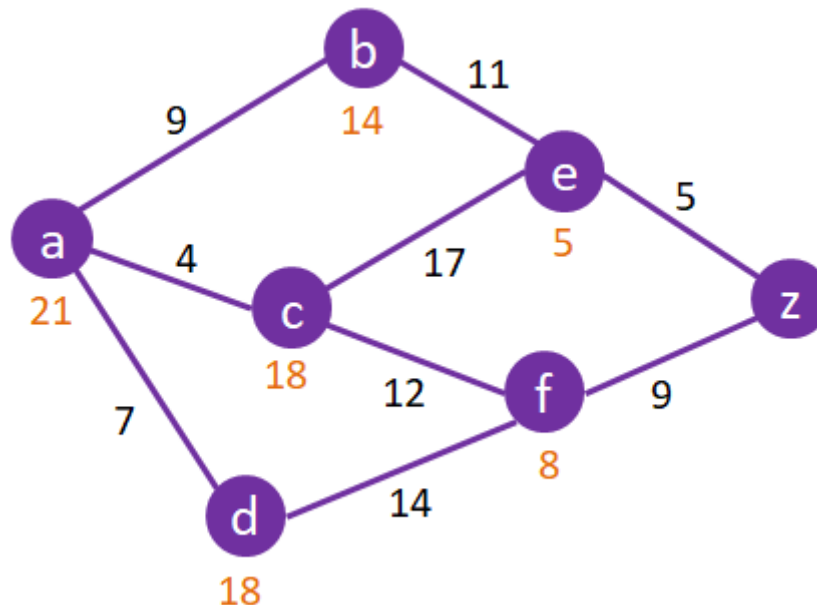
2: Eider Diaz A00828174, Campus Monterrey

3: Miguel Cortes A01270966, Campus Monterrey

Heuristic search algorithms

In this section you must solve some problems using informed search algorithms. Problems that additionally to the problem formulation information use a cost estimation for reaching the closest goal from each state.

Consider the following weighted graph that describes the space of states (nodes) and actions (arcs) for a simple problem:

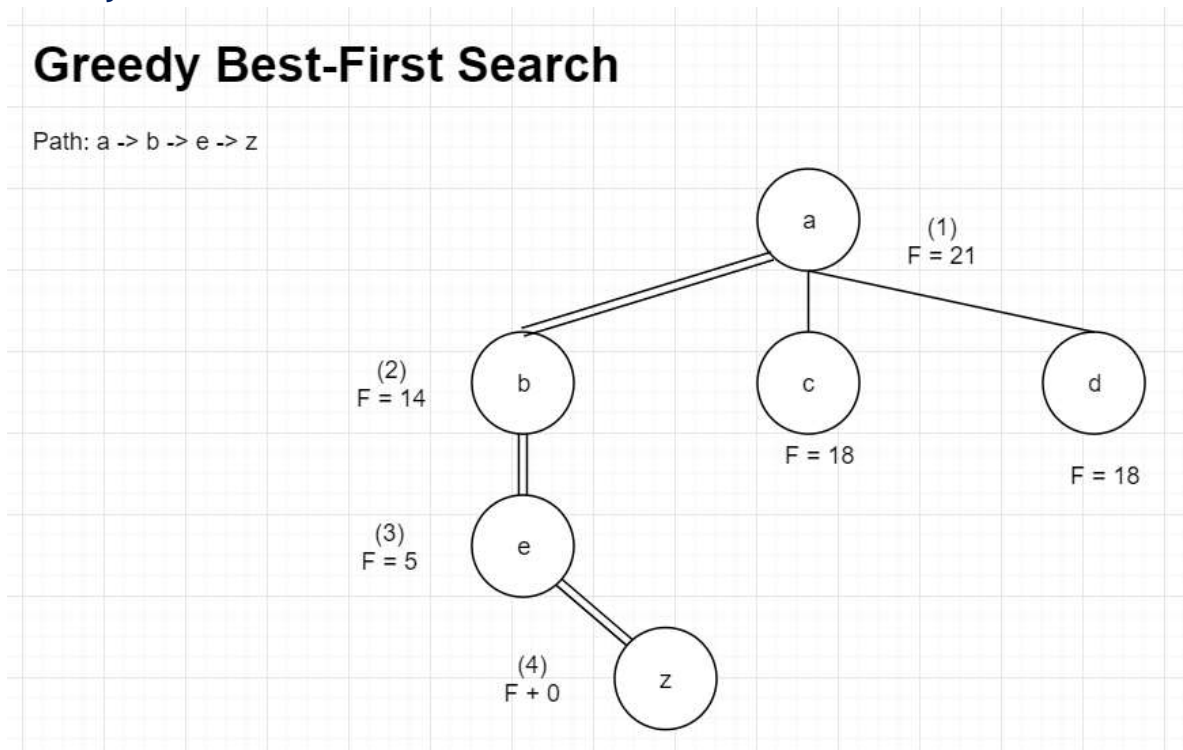


Instructions

- The initial state for each problem is the node a and the goal state is z.
- For each problem, you must draw a search tree showing all generated nodes.

- Every generated node must show the value of its evaluation function.
- Every visited node must show a number indicating its order in the sequence of expansions (starting at 1).
- Generated but not visited nodes must not be numbered.
- In the case of a tie between best nodes, visit the first generated node. If they are siblings, visit the sibling with them in lexicographic order (a, b, c, ...).
- In each case, clearly show the solution found (path from the initial state to the goal state) and its path cost.
- Cross out every eliminated node.

1. Greedy best-first search

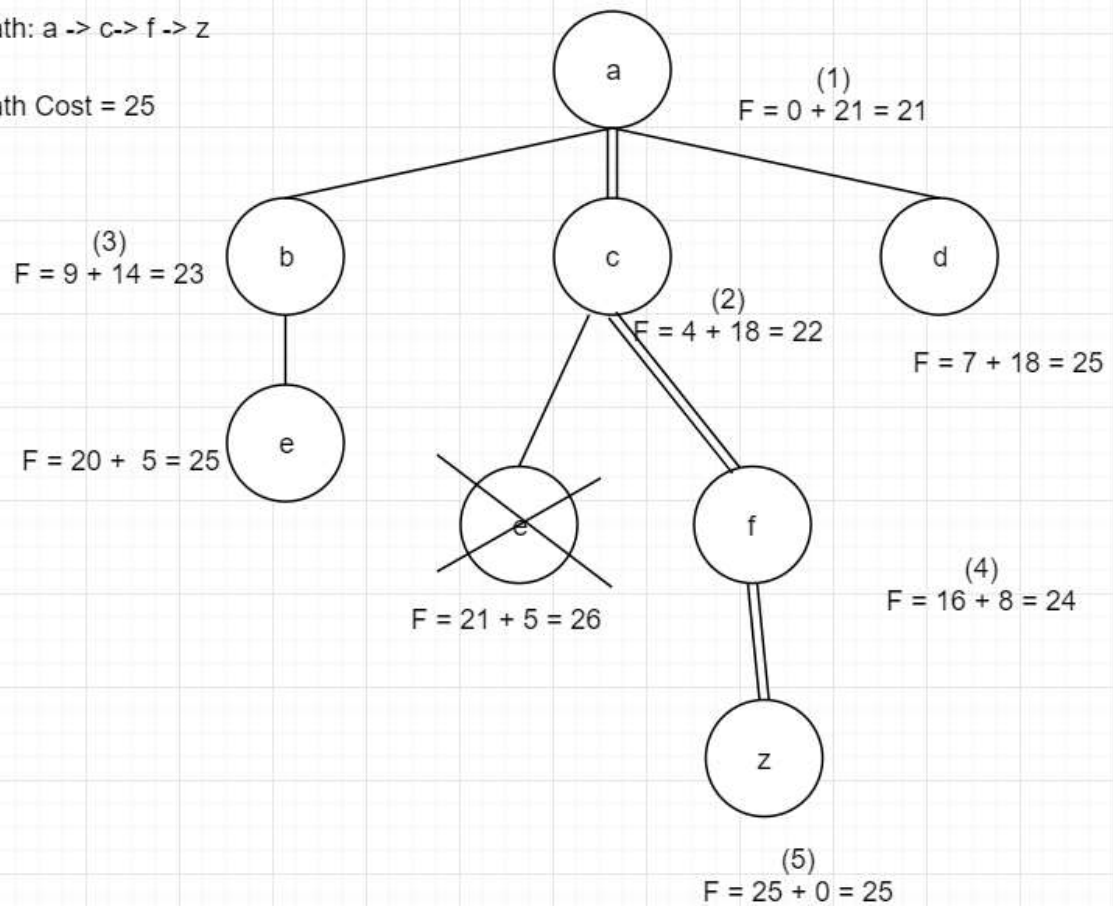


2. A*

A* Search

Path: a -> c -> f -> z

Path Cost = 25



3. Is the heuristic function admissible? Justify your answer.

Yes, the heuristic is admissible since it is always lower or equal than the actual cost for reaching the goal. Failing to do so (overestimating) could make our algorithm overlook an optimal solution in a search problem.

4. Is the heuristic function consistent? Justify your answer.

Yes, in the heuristic employed before once a node is expanded, the cost to reach it was the lowest possible. If we were to use an heuristic that was admissible but not consistent, it wouldn't be guaranteed that once a node expands we have arrived to it at the lowest possible cost.

Designing heuristics

Design 2 heuristics for the problem of sliding block puzzle. A Sliding Block Puzzle, like the one you can find on <https://www.proprofs.com/games/sliding-block-puzzle/>, consists of a number of yellow blocks just inside a small box, a single red block, and an opening.



Instructions

- Click on one of the white buttons with cursor to choose your puzzle.
- Click, hold, and drag the colored blocks to move them into the empty spaces in the frame one at a time.
- Repeat and move the red block progressively to the frame exit and out of the frame.

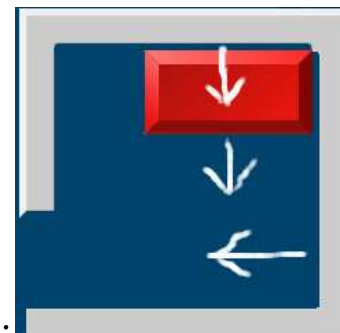
Heuristic 1

1. How do you evaluate the states with this heuristic? Add one example.

The first heuristic we propose estimates the number of times one has to move the red piece in order to complete the puzzle. For this estimation it is assumed that there are no yellow pieces in play making it really easy to complete an estimation, here is an example:



State:



Heuristic estimation:

In this example it is easy to see that if we were to ignore the yellow pieces, the puzzle could be completed in 3 steps, so this state would receive a heuristic estimation of $h(n) = 3$.

2. Is this an admissible heuristic? Justify your answer.

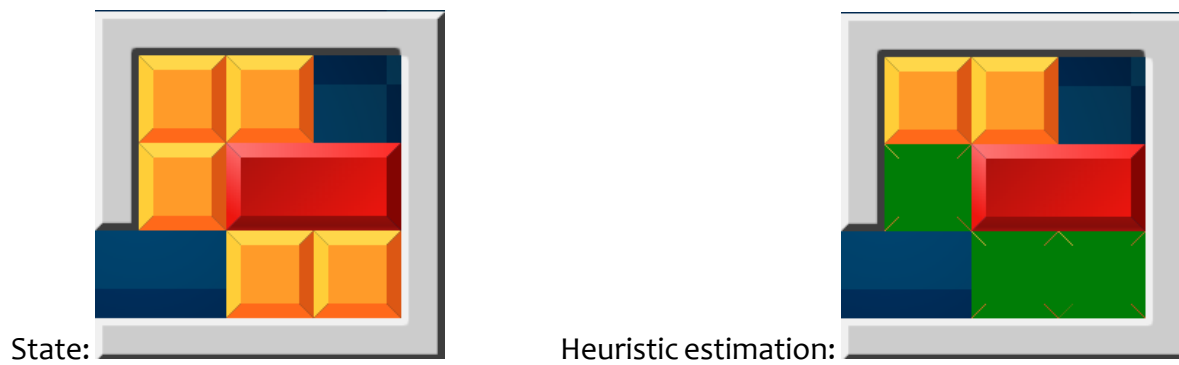
It is admissible because it underestimates the number of moves that it will take to solve the puzzle. In fact, even if there were no yellow blocks, the number of moves it would

take to solve it would be *at least* the number of moves the heuristic computes, so the admissibility condition $h(n) \leq h^*(n)$ is met.

Heuristic 2

1. How do you evaluate the states with this heuristic? Add one example.

The second heuristic we propose consists in counting the number of yellow pieces to the left or below the red piece (troublesome pieces) to estimate the number of moves we need to clear the path in order to solve the puzzle. We choose the pieces to the left and below the red piece because those are the pieces directly blocking the goal of the red piece while the ones that are right or up are already “out of the way”. Here is an example:



Here the 3 pieces highlighted in green are counted by the heuristic estimation so that $h(n) = 3$.

2. Is this an admissible heuristic? Justify your answer.

It is admissible because it assumes that you will need *at least* 1 action per yellow piece to get the piece out of the way. When the puzzle is almost solved it takes exactly one action to move the troublesome pieces out of the way, but more often than not this will be an underestimation since it will take more than one move per troublesome piece, so $h(n) \leq h^*(n)$.