UNIFORM COST SEARCH

Uniform cost search (UCS) is a breath-first search where the node to be expanded is the one with the lowest accumulated path cost in the frontier. Let’s use it to search for the best path from Arad to Giurigu. “Best path” is defined as shortest distance.

We will use Graph Search:

```
function GRAPH-SEARCH(problem, fringe) return a solution, or failure
   closed ← an empty set
   fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
   loop do
      if fringe is empty then return failure
      node ← REMOVE-FRONT(fringe)
      if GOAL-TEST(problem, STATE[node]) then return node
      if STATE[node] is not in closed then
         add STATE[node] to closed
         for child-node in EXPAND(STATE[node], problem) do
            fringe ← INSERT(child-node, fringe)
      end
   end
```

1
this page intentionally left blank

you can separate the two pages in this handout
Uniform Cost Search using priority queue

Frontier (fill from L–R):

Start state is Arad.
Goal state is Giurgiu.

Draw nodes in the Fringe as they are expanded. When they get removed for goal-testing, cross them out and put them in the closed set as states.

**Draw nodes as a circle** with a letter inside, and **include the total path cost**.

**Treat the Fringe as a priority queue**. When performing “Remove-Front,” take the node with the smallest accumulated cost.

a. What path is returned by the algorithm?

b. What is its path cost (total distance)?

c. What did you observe that was interesting about the algorithm?