Contents

1 A01 - Assignment 1: Search Algorithm

Students:

- Deidier Simone - student number: *133020*
- Esposito Davide - student number: *132667*

1.1 Part One: Uninformed Search

1.1.1 Breadth First Search

Pseudocode of the algorithm (*adapted on our problem*):

```
BFS(G, problem):

    start = problem.INITIAL_STATE;
    if problem.goal_test(start.STATE):
        return SOLUTION(start);

    frontier <- FIFO queue;
    explored <- empty set;

    frontier.add(start);
    loop:
        if empty(frontier):
            return failure;

        curr = frontier.pop();
        explored.add(curr);
        if problem.goal_test(curr):
            return SOLUTION(curr);
        else:
            children = curr.expand();
            for child in children:
```

```
            if child not in explored and not in frontier:
                frontier.insert(child);
    goto loop
```

Assumptions

- We added the *explored* list because the graph has some cycles, and with that we can avoid to get stuck in a loop.
- The control on nodes to add them to the expand set, is done on the specific node and not on its state, which means that different nodes with the same state can be added to the *frontier* more times, but every node is considered once.
- As specified in the assingnment, the cost for each action is equal and unitary, so we do not consider them in the implementation.
- When reconstructing the solution, if a node has more than one father, the left-most is choosen (*left-to-right order of successors*).

Execution of the algorithm on our problem:

- **STEP 1**

$$\text{Frontier} = \{q_1\}$$
$$\text{Explored} = \{q_1\}$$
$$\text{curr} = q_1$$

- q_1 is not the goal, so **STEP 2**

$$\text{Frontier} = \{q_2, q_3, q_4, q_5\}$$
$$\text{Explored} = \{q_1, q_2\}$$
$$\text{curr} = q_2$$

- q_2 is not the goal, so **STEP 3**

$$\text{Frontier} = \{q_3, q_4, q_5\}$$
$$\text{Explored} = \{q_1, q_2, q_3\}$$
$$\text{curr} = q_3$$

- q_3 is not the goal, so **STEP 4**

$$\text{Frontier} = \{q_4, q_5\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4\}$$
$$\text{curr} = q_4$$

- q_4 is not the goal, so **STEP 5**

$$\text{Frontier} = \{q_5, q_6\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5\}$$
$$\text{curr} = q_5$$

- q_5 is not the goal, so **STEP 6**

$$\text{Frontier} = \{q_6\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6\}$$
$$\text{curr} = q_6$$

- q_6 is not the goal, so **STEP 7**

$$\text{Frontier} = \{q_7, q_8\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$$
$$\text{curr} = q_7$$

- q_7 is not the goal, so **STEP 8**

$$\text{Frontier} = \{q_8, q_9, q_{10}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$$
$$\text{curr} = q_8$$

- q_8 is not the goal, so **STEP 9**

$$\text{Frontier} = \{q_9, q_{10}, q_{11}, q_{12}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$$
$$\text{curr} = q_9$$

- q_9 is not the goal, so **STEP 10**

$$\text{Frontier} = \{q_{10}, q_{11}, q_{12}, q_{13}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}$$
$$\text{curr} = q_{10}$$

- q_{10} is not the goal, so **STEP 11**

$$\text{Frontier} = \{q_{11}, q_{12}, q_{13}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$$
$$\text{curr} = q_{11}$$

- q_{11} is not the goal, so **STEP 12**

$$\text{Frontier} = \{q_{12}, q_{13}, q_{14}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}\}$$
$$\text{curr} = q_{12}$$

- q_{12} is not the goal, so **STEP 13**

$$\text{Frontier} = \{q_{13}, q_{14}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}\}$$
$$\text{curr} = q_{13}$$

- q_{13} is not the goal, so **STEP 14**

$$\text{Frontier} = \{q_{14}, q_{15}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}\}$$
$$\text{curr} = q_{14}$$

- q_{14} is not the goal, so **STEP 15**

$$\text{Frontier} = \{q_{15}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}\}$$
$$\text{curr} = q_{15}$$

- q_{15} is not the goal, so **STEP 16**

$$\text{Frontier} = \{q_{16}\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}\}$$
$$\text{curr} = q_{16}$$

- q_{16} is the goal, so **STEP 17**, *reconstruct solution*:

$$\text{Solution} = \{q_{16}, q_{15}, q_{13}, q_9, q_7, q_6, q_4, q_1\}$$

- Total number of steps done by this algorithm (*cost of the algorithm*): **17**.
- Total cost of the path found by this algorithm (*cost of the solution*): **7**.

1.1.2 Depth-first Search

Pseudocode of the algorithm (*adapted on our problem*):

```
DFS(G, problem):

    start = problem.INITIAL_STATE;
    if problem.goal_test(start.STATE):
        return SOLUTION(start);

    frontier <- LIFO queue;
    explored <- empty set;

    frontier.add(start);
    loop:
        if empty(frontier):
            return failure;

        curr = frontier.pop();
        explored.add(curr);
        if problem.goal_test(curr):
            return SOLUTION(curr);
        else:
            children = curr.expand();
            for child in children:
                if child not in explored and not in frontier:
                    frontier.push(child);
        goto loop
```

Assumptions

- Same assumptions as the *BFS* algorithm.
- The *frontier* set is implemented as a *LIFO* stack (*Last In First Out*).

Execution of the algorithm on our problem:

- **STEP 1**

$$\text{Frontier} = \{q_1\}$$
$$\text{Explored} = \{q_1\}$$
$$\text{curr} = q_1$$

- q_1 is not the goal, so **STEP 2**

$$\text{Frontier} = \{q_2, q_3, q_4, q_5\}$$
$$\text{Explored} = \{q_1, q_5\}$$
$$\text{curr} = q_5$$

- q_5 is not the goal, so **STEP 3**

$$\text{Frontier} = \{q_2, q_3, q_4\}$$
$$\text{Explored} = \{q_1, q_2, q_4\}$$
$$\text{curr} = q_4$$

- q_4 is not the goal, so **STEP 4**

$$\text{Frontier} = \{q_2, q_3, q_6\}$$
$$\text{Explored} = \{q_1, q_2, q_4, q_6\}$$
$$\text{curr} = q_6$$

- q_6 is not the goal, so **STEP 5**

$$\text{Frontier} = \{q_2, q_3, q_7, q_8\}$$
$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8\}$$
$$\text{curr} = q_8$$

- q_8 is not the goal, so **STEP 6**

$$\text{Frontier} = \{q_2, q_3, q_7, q_{11}, q_{12}\}$$
$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8, q_{12}\}$$
$$\text{curr} = q_{12}$$

- q_{12} is not the goal, so **STEP 7**

$$\text{Frontier} = \{q_2, q_3, q_7, q_{11}\}$$
$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8, q_{12}, q_{11}\}$$
$$\text{curr} = q_{11}$$

- q_{11} is not the goal, so **STEP 8**

$$\text{Frontier} = \{q_2, q_3, q_7, q_{11}, q_{13}, q_{14}\}$$
$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8, q_{12}, q_{11}, q_{14}\}$$
$$\text{curr} = q_{14}$$

- q_{14} is not the goal, so **STEP 9**

$$\text{Frontier} = \{q_2, q_3, q_7, q_{11}, q_{13}\}$$

$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8, q_{12}, q_{11}, q_{14}, q_{13}\}$$

$$\text{curr} = q_{13}$$

- q_{13} is not the goal, so **STEP 10**

$$\text{Frontier} = \{q_2, q_3, q_7, q_{11}, q_{15}\}$$

$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8, q_{12}, q_{11}, q_{14}, q_{13}, q_{15}\}$$

$$\text{curr} = q_{15}$$

- q_{15} is not the goal, so **STEP 11**

$$\text{Frontier} = \{q_2, q_3, q_7, q_{11}, q_{16}\}$$

$$\text{Explored} = \{q_1, q_2, q_4, q_6, q_8, q_{12}, q_{11}, q_{14}, q_{13}, q_{15}, q_{16}\}$$

$$\text{curr} = q_{16}$$

- q_{16} is the goal, so **STEP 12**, *reconstruct solution*:

$$\text{Solution} = \{q_{16}, q_{15}, q_{13}, q_{11}, q_8, q_6, q_4, q_1\}$$

- Total number of steps done by this algorithm (*cost of the algorithm*): **12**.
- Total cost of the path found by this algorithm (*cost of the solution*): **7**.

1.1.3 Uniform Cost Search (*Dijkstra*)

Pseudocode of the algorithm (*adapted on our problem*):

```
UCS(G, problem):

    start = problem.INITIAL_STATE;
    if problem.goal_test(start.STATE):
        return SOLUTION(start);

    frontier <- priority queue;
    explored <- empty set;

    frontier.add(start, 0);
    loop:
        if empty(frontier):
            return failure;

        curr = frontier.get_lowest_cost();
        explored.add(curr);
```

```
if problem.goal_test(curr):
    return SOLUTION(curr);
else:
    children = curr.expand();
    for child in children:
        if child not in explored and not in frontier:
            frontier.push(child, child.PATH_COST);
        else if child in frontier
            and child.PATH_COST < frontier.get_cost(child):
            frontier.update(child, child.PATH_COST);
goto loop
```

Assumptions

- Same assumptions as the *BFS* algorithm.
- The *frontier* set is implemented as a *queue*, that contains all the nodes with the path cost, and the with the function *get_lowest_cost()* we can get the node with the lowest path cost inside the set (*priority queue*).

Execution of the algorithm on our problem:

- **STEP 1**

$$\text{Frontier} = \{(q_1, 0)\}$$
$$\text{Explored} = \{q_1\}$$
$$\text{curr} = q_1$$

- q_1 is not the goal, so **STEP 2**

$$\text{Frontier} = \{(q_2, 1), (q_3, 1), (q_4, 1), (q_5, 1)\}$$
$$\text{Explored} = \{q_1, q_2\}$$
$$\text{curr} = q_2$$

- q_2 is not the goal, so **STEP 3**

$$\text{Frontier} = \{(q_3, 1), (q_4, 1), (q_5, 1)\}$$
$$\text{Explored} = \{q_1, q_2, q_3\}$$
$$\text{curr} = q_3$$

- q_3 is not the goal, so **STEP 4**

$$\text{Frontier} = \{(q_4, 1), (q_5, 1)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4\}$$
$$\text{curr} = q_4$$

- q_4 is not the goal, so **STEP 5**

$$\text{Frontier} = \{(q_5, 1), (q_6, 2)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5\}$$
$$\text{curr} = q_5$$

- q_5 is not the goal, so **STEP 6**

$$\text{Frontier} = \{(q_6, 2)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6\}$$
$$\text{curr} = q_6$$

- q_6 is not the goal, so **STEP 7**

$$\text{Frontier} = \{(q_7, 3), (q_8, 3)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$$
$$\text{curr} = q_7$$

- q_7 is not the goal, so **STEP 8**

$$\text{Frontier} = \{(q_8, 3), (q_9, 4), (q_{10}, 4)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$$
$$\text{curr} = q_8$$

- q_8 is not the goal, so **STEP 9**

$$\text{Frontier} = \{(q_9, 4), (q_{10}, 4), (q_{11}, 4), (q_{12}, 4)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$$
$$\text{curr} = q_9$$

- q_9 is not the goal, so **STEP 10**

$$\text{Frontier} = \{(q_{10}, 4), (q_{11}, 4), (q_{12}, 4), (q_{13}, 5)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}$$
$$\text{curr} = q_{10}$$

- q_{10} is not the goal, so **STEP 11**

$$\text{Frontier} = \{(q_{11}, 4), (q_{12}, 4), (q_{13}, 5)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$$
$$\text{curr} = q_{11}$$

- q_{11} is not the goal, so **STEP 12**

$$\text{Frontier} = \{(q_{12}, 4), (q_13, 5), (q_{14}, 5)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}\}$$
$$\text{curr} = q_{12}$$

- q_{12} is not the goal, so **STEP 13**

$$\text{Frontier} = \{(q_{13}, 5), (q_{14}, 5)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}\}$$
$$\text{curr} = q_{13}$$

- q_{13} is not the goal, so **STEP 14**

$$\text{Frontier} = \{(q_{14}, 5), (q_{15}, 6)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}\}$$
$$\text{curr} = q_{14}$$

- q_{14} is not the goal, so **STEP 15**

$$\text{Frontier} = \{(q_{15}, 6)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}\}$$
$$\text{curr} = q_{15}$$

- q_{15} is not the goal, so **STEP 16**

$$\text{Frontier} = \{(q_{16}, 7)\}$$
$$\text{Explored} = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}\}$$
$$\text{curr} = q_{16}$$

- q_{16} is the goal, so **STEP 17**, *reconstruct solution*:

$$\text{Solution} = \{q_{16}, q_{15}, q_{13}, q_9, q_7, q_6, q_4, q_1\}$$

- Total number of steps done by this algorithm (*cost of the algorithm*): **17**.
- Total cost of the path found by this algorithm (*cost of the solution*): **7**.
- The UCS algorithm is equal to the BFS, this is because each edge has the same unitary cost and we are using the left-to-right order of successors.

1.2 Part Two: Informed Search

1.2.1 Heuristic function

The function $h(n)$ we implemented for the informed search works in the following way:

- We used a 4-bit representation of the state of all the nodes in the graph keeping track of the Wolf, Goat, Cabbage and the farmer location (*MSB for the Wolf, second bit for the Goat, third for the Cabbage and LSB for the farmer, e.g. the state 1010 means that the farmer is on the second side of the river with the Goat, the Wolf and Cabbage are on the other side*).
- The function assigns a cost of *100* to all the failure nodes (*failure states represented in red in the graph*), those one where the couple Wolf-Goat or Goat-Cabbage are left alone in one of the two sides.
- The functions assigns a cost of *10* to all the danger nodes (*represented in yellow in the graph*), those one where a next wrong decision will get us to a failure node.
- For all the other nodes, the function assigns a value corresponding to the number of missing items in the first side of the river, excluding the farmer position (*e.g. to the state 0101 is assigned a cost of 1*).

$$h(n) = \begin{cases} 0 & \text{if} \quad n = (000*)_2 \\ 1 & \text{if} \quad n = (010*)_2 \\ 2 & \text{if} \quad n = (101*)_2 \\ 10 & \text{if} \quad \text{danger state} \\ 100 & \text{if} \quad \text{failure state} \end{cases}$$

where n is the binary value of the node (*state*) and $*$ indicates whatever value for that bit in the state representation (*don't care - DC*).

The function we provide is ***admissible***, because for every node n the function returns a value that is ≥ 0 and for the goal node returns a value that is $= 0$. Furthermore, we know that our function is not ***consistent***, because there are a couple of nodes where the function is not compliant with the relation:

$$h(n) \leq c(n, a, n') + h(n')$$

This result is relevant to us, since we know that consinstency $\implies$ admissibility, but admissibility $\not\Rightarrow$ consistency. Nevertheless, as can be seen in the examples of the algorithms, the heuristic function we proposed is functional for solving the problem. If the function had also been consistent, the algorithm `A*` would have taken fewer steps to find the solution to the problem.

1.2.2 Greedy Best-First

Pseudocode of the algorithm (*adapted on our problem*):

```
GBF(G, problem, h):

    start = problem.INITIAL_STATE;
    if problem.goal_test(start.STATE):
        return SOLUTION(start);

    frontier <- priority queue on heuristic h;
    frontier.HEURISTIC <- h;
    explored <- empty set;

    frontier.add(start, 0);
    loop:
        if empty(frontier):
            return failure;

        curr = frontier.pop();
        explored.add(curr);
        if problem.goal_test(curr):
            return SOLUTION(curr);
        else:
            children = curr.expand();
            for child in children:
                if child not in explored and not in frontier:
                    frontier.add(child);
        goto loop
```

Assumptions

- We added the *explored* list because the graph has some cycles, and with that we can avoid to get stuck in a loop.
- As specified in the assingnment, the cost for each action is equal and unitary, so we do not consider them in the implementation.
- The *frontier* set is implemented as a *priority queue*, where lowest heuristic values are prioritized and for the same values of the heuristic function the last one inserted is prioritized (*LIFO*).

Execution of the algorithm on our problem:

- **STEP 1**

$$\text{Frontier} = \{(q_1, 10)\}$$

$$\text{Explored} = \{q_1\}$$

$$\text{curr} = q_1$$

- q_1 is not the goal, so **STEP 2**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_4, 2), (q_5, 100)\}$$
$$\text{Explored} = \{q_1, q_4\}$$
$$\text{curr} = q_4$$

- q_4 is not the goal, so **STEP 3**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_5, 100), (q_6, 2)\}$$
$$\text{Explored} = \{q_1, q_4, q_6\}$$
$$\text{curr} = q_6$$

- q_6 is not the goal, so **STEP 4**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_5, 100), (q_7, 10), (q_8, 10)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7\}$$
$$\text{curr} = q_7$$

- q_7 is not the goal, so **STEP 5**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_5, 100), (q_8, 10), (q_9, 10), (q_{10}, 100)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_9\}$$
$$\text{curr} = q_9$$

- q_9 is not the goal, so **STEP 6**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_5, 100), (q_8, 10), (q_{10}, 100), (q_{13}, 1)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_9, q_{13}\}$$
$$\text{curr} = q_{13}$$

- q_{13} is not the goal, so **STEP 7**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_5, 100), (q_8, 10), (q_{10}, 100), (q_{15}, 1)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_9, q_{13}, q_{15}\}$$
$$\text{curr} = q_{15}$$

- q_{15} is not the goal, so **STEP 8**

$$\text{Frontier} = \{(q_2, 100), (q_3, 100), (q_5, 100), (q_8, 10), (q_{10}, 100), (q_{16}, 0)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_9, q_{13}, q_{15}, q_{16}\}$$
$$\text{curr} = q_{16}$$

- q_{16} is the goal, so **STEP 9**, *reconstruct solution*:

$$\text{Solution} = \{q_{16}, q_{15}, q_{13}, q_9, q_7, q_6, q_4, q_1\}$$

- Total number of steps done by this algorithm (*cost of the algorithm*): **9**.
- Total cost of the path found by this algorithm (*cost of the solution*): **7**.

1.2.3 A*

Pseudocode of the algorithm (*adapted on our problem*):

```
A*(G, problem, h):

    func g <- CUMULATIVE_COST;
    lambda f(n) <- g(n) + h(n);
    return GBF(G, problem, f)
```

Assumptions

- Same assumptions as the *Greedy Best-First* algorithm.

Execution of the algorithm on our problem:

- **STEP 1**

$$\text{Frontier} = \{(q_1, 10)\}$$
$$\text{Explored} = \{q_1\}$$
$$\text{curr} = q_1$$

- q_1 is not the goal, so **STEP 2**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_4, 12), (q_5, 110)\}$$
$$\text{Explored} = \{q_1, q_4\}$$
$$\text{curr} = q_4$$

- q_4 is not the goal, so **STEP 3**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_6, 14)\}$$
$$\text{Explored} = \{q_1, q_4, q_6\}$$
$$\text{curr} = q_6$$

- q_6 is not the goal, so **STEP 4**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_7, 24), (q_8, 24)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7\}$$
$$\text{curr} = q_7$$

- q_7 is not the goal, so **STEP 5**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_8, 24), (q_9, 34), (q_{10}, 124)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_8\}$$
$$\text{curr} = q_8$$

- q_8 is not the goal, so **STEP 6**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_9, 34), (q_{10}, 124), (q_{11}, 34), (q_{12}, 124)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_8, q_{11}\}$$
$$\text{curr} = q_{11}$$

- q_{11} is not the goal, so **STEP 7**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_9, 34), (q_{10}, 124), (q_{12}, 124), (q_{13}, 35), (q_{14}, 134)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_8, q_{11}, q_9\}$$
$$\text{curr} = q_9$$

- q_9 is not the goal, so **STEP 8**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_{10}, 124), (q_{12}, 124), (q_{13}, 35), (q_{14}, 134)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_8, q_{11}, q_9, q_{13}\}$$
$$\text{curr} = q_{13}$$

- q_{13} is not the goal, so **STEP 9**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_{10}, 124), (q_{12}, 124), (q_{14}, 134), (q_{15}, 36)\}$$
$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_8, q_{11}, q_9, q_{13}, q_{15}\}$$
$$\text{curr} = q_{15}$$

- q_{15} is not the goal, so **STEP 10**

$$\text{Frontier} = \{(q_2, 110), (q_3, 110), (q_5, 110), (q_{10}, 124), (q_{12}, 124), (q_{14}, 134), (q_{16}, 36)\}$$

$$\text{Explored} = \{q_1, q_4, q_6, q_7, q_8, q_{11}, q_9, q_{13}, q_{15}, q_{16}\}$$

$$\text{curr} = q_{16}$$

- q_{16} is the goal, so **STEP 11**, *reconstruct solution*:

$$\text{Solution} = \{q_{16}, q_{15}, q_{13}, q_9, q_7, q_6, q_4, q_1\}$$

- Total number of steps done by this algorithm (*cost of the algorithm*): **11**.
- Total cost of the path found by this algorithm (*cost of the solution*): **7**.

1.3 Sources

- S. Russel, P. Norvig. Artificial Intelligence: a Modern Approach (*4th ed*).
- Lecture Slides and Assignment Lecture Material from TDT4136 - Introduction to Artificial Intelligence course (*2024 AUTUMN*)