Fundamentals of Computer Science

Homework Set 9

December 20, 2023

1. (4') In this question, we use a **breadth-first** approach to solve eight-puzzle problem.
You should first read the following program implementation, and then fill the output of the program in the form. The status should be printed in row major, with the blank cell replaced by '0'. For example, the initial status can be expressed as "123405786" without quotes

```
procedure BFS_Eight_Puzzle(InitState)
if InitState is the target state:
    print "end" and return
Q <= initialize an empty queue
push the InitState into Q
print InitState in row-major order
while (Q is not empty) do:
    S <= pop an element from Q
    for D in [Left, Down, Right, Up] sequentially:
        if there is such a block which can move D:
            S' <= apply D to the block
            if S' has not been pushed to Q in the past:
                print S' in row-major order
                if S' is the target state:
                    print "end"
                    delete Q and return
                else:
                    push S' into Q
```

1	2	3
4		5
7	8	6

Example Output (from the initial state "123456708")	
Line 1	123456708
Line 2	123456780
Line 3	end
Line 4	

Your Answer	
Line 1	123405786
Line 2	123450786
Line 3	103425786
Line 4	123045786
Line 5	123475608
Line 6	123453786
Line 7	123456780
Line 8	end
Line 9	
Line 10	
Line 11	
Line 12	
Line 13	
Line 14	
Line 15	
Line 16	

2. (6’) In this question, we use **the number of out-of-place tiles as a heuristic** to solve eight-puzzle problem.
You should first read the following program implementation, and then fill the output of the program in the form. The status should be printed in row major, with the blank cell replaced by ‘0’. For example, the initial status can be expressed as "123408765" without quotes.
In this program, we use a data structure named **priority queue**, which can pop the element with the highest priority. In computer science, data structures such as heap, self-balancing binary search tree, etc. can all be implemented as priority queue. When comparing two states, the state with the smaller **heuristic** (i.e., number of out-of-place tiles) has a higher priority. If the **heuristic** of two states are equal, the state that was pushed into the priority queue **later** has a higher priority.

```
procedure A*_Eight_Puzzle(InitState)
if InitState is the target state:
    print “end” and return
Q <= initialize an empty priority queue
push the InitState into Q
print InitState in row-major order
while (Q is not empty) do:
    S <= pop the element with the highest priority from Q
    for D in [Left, Down, Right, Up] sequentially:
        if there is such a block which can move D:
            S’ <= apply D to the block
            if S’ has not been pushed to Q in the past:
                print S’ in row-major order
                if S’ is the target state:
                    print “end”
                    delete Q and return
                else:
                    push S’ into Q
```

1	2	3
4		8
7	6	5

Your Answer	
Line 1	123408765
Line 2	123480765
Line 3	103428765
Line 4	123048765
Line 5	123468705
Line 6	123468750
Line 7	123468075
Line 8	123460758
Line 9	120463758
Line 10	123406758
Line 11	103426758
Line 12	123046758
Line 13	123456708
Line 14	123456780
Line 15	end
Line 16	
Line 17	