CS3243 Assignment 4

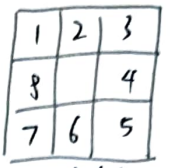
Zhuang Jianning
A0214561M
T03

3. 8-Puzzle

f(state) = # of mismatched tiles compared to goal state

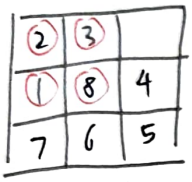
always choose action that lead to state with lowest cost

action valid only if f(next_state) ≤ f(current_state)



goal state

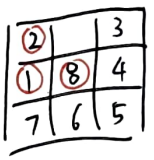
a)

① 

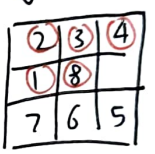
initial state

f(initial) = 4

empty space can move left or down



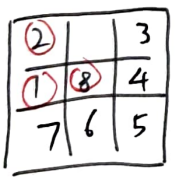
f(left) = 3



f(down) = 5

f(left) < f(initial)

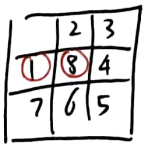
left chosen

② 

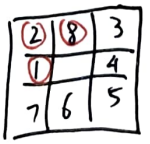
current

f(current) = 3

empty space can move left, down or right



f(left) = 2

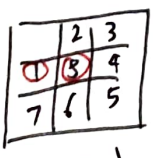


f(down) = 3

back to ①
f(right) = 4

f(left) < f(current)

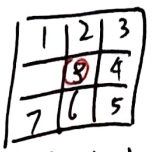
left chosen

③ 

current

f(current) = 2

empty space can move down or right



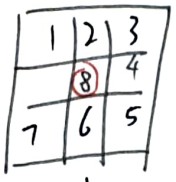
f(down) = 1

back to ②
f(right) = 3

f(down) < f(current)

down chosen

④



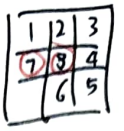
current

f(current) = 1

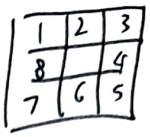
empty space can move up, down, right

back to (3)

f(up) = 2



f(down) = 2



f(right) = 0 ✓

goal state

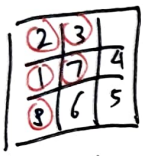
right chosen

sequence of actions ⇒ [left, left, down, right]

goal state reached is a global minimum

b)

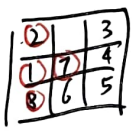
①



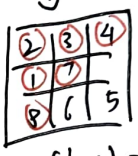
initial state

f(initial) = 5

empty space can move left or down



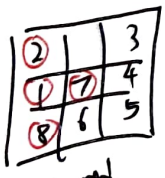
f(left) = 4 ✓



f(down) = 6

left chosen

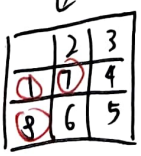
②



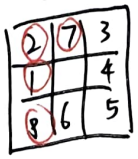
current

f(current) = 4

empty space can move left, down or right



f(left) = 3 ✓



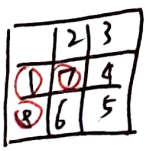
f(down) = 4

back to ①

f(right) = 5

left chosen

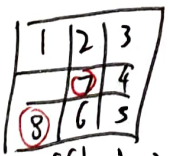
③



current

f(current) = 3

empty space can move down or right



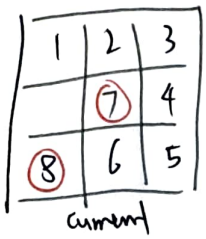
f(down) = 2 ✓

back to ②

f(right) = 4

down chosen

④

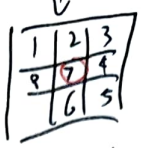


f(current) = 2

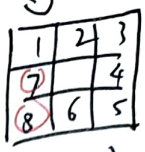
empty space can move up, down, right

back to ③

f(up) = 3



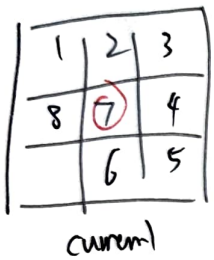
f(down) = 1 ✓



f(right) = 2

down chosen

⑤

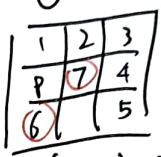


f(current) = 1

empty space can move up or right

back to ④

f(up) = 2



f(right) = 2

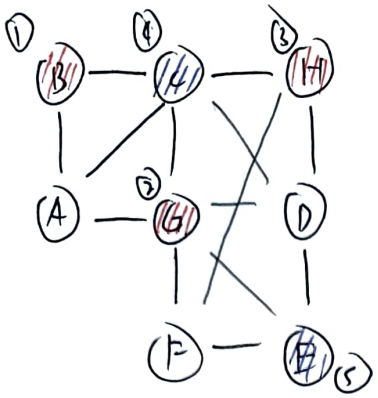
2 1 — 1

both moves lead to states with higher cost than current

current is returned as a local minimum

sequence of actions ⇒ [left, left, down, down]

4. a)



b) f(state) = # pairs of adjacent vertices with the same colour

always choose action that leads to lowest cost
↳ changing colour of single vertex

tie break: try to use existing colours

① f(initial) = 12

G is connected to most vertices (5) of the same colour
changing colour of G will result in lowest f(next_state) cost of 12 − 5 = 7
try to use colours already currently used (red) to open up options e.g. colour G red

A & C cannot be resolved if change G to red

② f(current) = 12 − 5 = 7

C, H, D, E connected to most vertices (3) of same colour
only changing H to red allowed
colour H red

③ f(current) = 7 − 3 = 4

C, D, E connected to most vertices (2) of the same colour
new colour has to be used, tie break by alphabet
colour C blue

④ f(current) = 4 − 2 = 2
E is connected to most vertices (2) of same colour
colour E blue ⑤ f(goal) = 0