Total Points: 25

15 puzzle is a sliding puzzle game with numbered squares arranged in 4X4 grid with one tile missing.

Initial State

1		2	4
5	7	3	8
9	6	11	12
13	10	14	15

The puzzle is solved when the numbers are arranged in order.

Goal State

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

The actions are defined in terms of direction where empty square can be moved to

UP (U), Down(D), Left(L), Right(R)

Write a program which performs a breadth-first search to find the solution to any given board position for 15 puzzle
The pseudocode is below:

Figure 3.9

---

function BREADTH-FIRST-SEARCH(*problem*) **returns** a solution node or *failure*
  node ← NODE(*problem*.INITIAL)
  if *problem*.IS-GOAL(*node*.STATE) **then return** *node*
  frontier ← a FIFO queue, with *node* as an element
  reached ← {*problem*.INITIAL}
  while not IS-EMPTY(*frontier*) **do**
    node ← POP(*frontier*)
    for each *child* **in** EXPAND(*problem*, *node*) **do**
      s ← *child*.STATE
      if *problem*.IS-GOAL(*s*) **then return** *child*
      if *s* is not in *reached* **then**
        add *s* to *reached*
        add *child* to *frontier*
  return *failure*

function UNIFORM-COST-SEARCH(*problem*) **returns** a solution node, or *failure*
  return BEST-FIRST-SEARCH(*problem*, PATH-COST)

Breadth-first search and uniform-cost search algorithms.

Input
The input should be given in the form of a sequence of numbered tiles for initial board configuration, '0' indicating the empty space (see example below)

Output

1. Moves
2. Number of Nodes expanded
3. Time Taken
4. Memory Used

Example
> 1 0 2 4 5 7 3 8 9 6 11 12 13 10 14 15
Moves: RDLDDRR
Number of Nodes expanded: 361

TIme Taken: 0.238
Memory Used: 704kb

Hint
- You can use hashset to keep track of explored nodes and fast lookup

Submission
Please upload the code to gradescope. The starter code written in python is helpful for compatibility with gradescope autograder.
- Source Code
- Readme.txt including instruction to run the code, include version of compiler you are using (e.g. java 1.8, c++11)

Programming Language
You can choose from C++, Java, Python.
Currently the gradescope autograder isn't compatible with Java and C++, I'm working on it, in the mean time I'll grade them manually, please upload it to gradescope as well.

Rubric

Print the moves to reach the solution => 10
Check for repeated states => 3
Print number of nodes expanded => 3
Print total memory usage => 3
Print total time taken => 3
Coding style, comments, readme instruction => 3

Your submission should be your own work. You can take reference from aima codebase but you are not allowed to use the code directly. All submissions will be checked for plagiarism using MOSS (Measure Of Software Similarity).