Programming Project 1:

Solving the 8-puzzle game using A\* algorithm:

Compiler: Java

Input: String of numbers for both initial and goal states

Example:

If initial state is 2 8 3 1 6 4 7 0 5 and goal state is 1 2 3 8 0 4 7 6 5 then give input as below in the command line arguments

2 8 3 1 6 4 7 0 5 1 2 3 8 0 4 7 6 5

'0' represents the empty slot in the board.

8-Puzzle formulation:

States: A state class of 8-puzzle will represent the location of each value and blank space on the board

Initial State: Any solvable state can be set as initial state

Actions: Movement of blank spaces which will generates the successors. The operations can be moving hole up, down, right, left

Successor function: This will generate the successor of the current state. For example if we move the hole to top for the initial state of 2 8 3 1 6 4 7 0 5 then the successor will be 2 8 3 1 0 4 7 6 5

Goal test: checking whether the generated the successor is equal to goal

Path cost: Each step cost is 1 and so the path cost is number of steps in the path

Program structure:

The program has four classes Main, EightpuzzleAstar, Puzzlestate, Puzzlenode.

Main class will take the string inputs of initial state and goal state from the command line arguments and convert them into 1-d arrays.

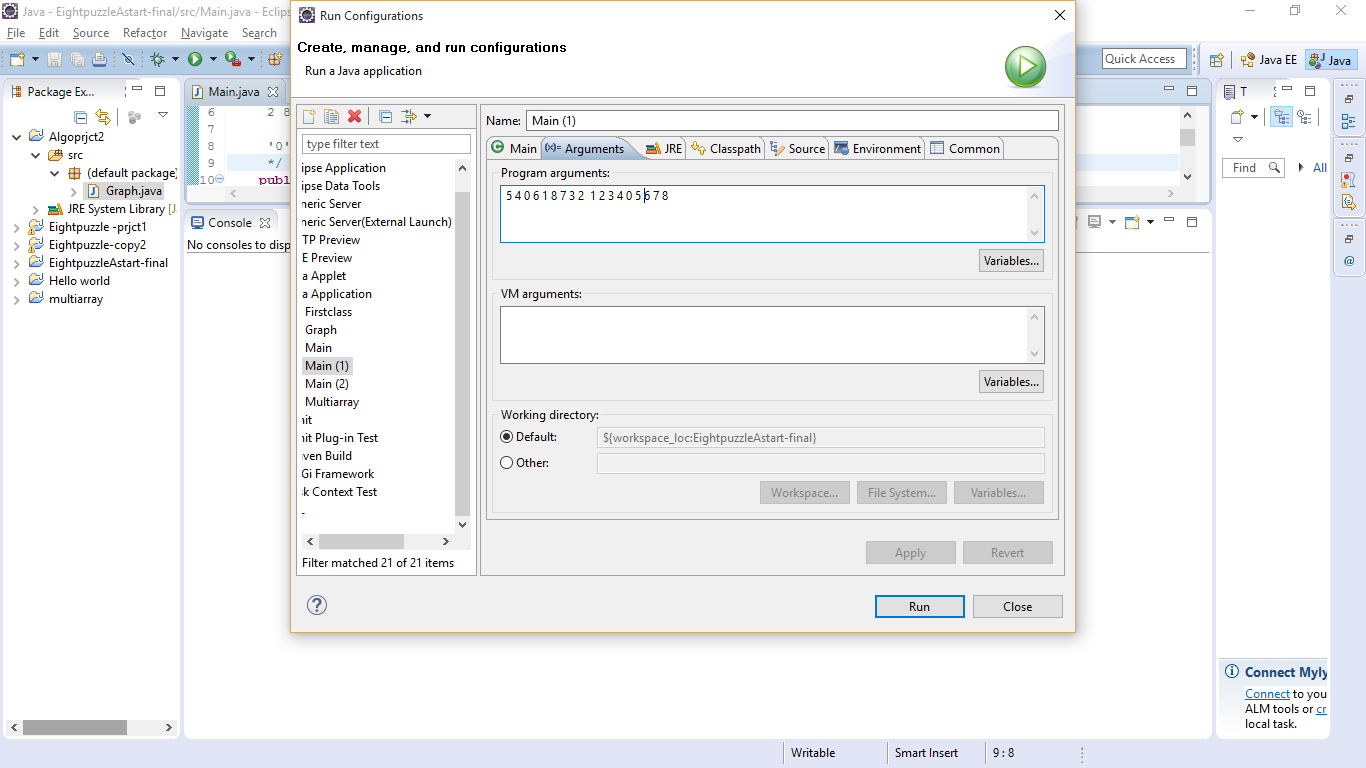
Puzzlestate class represents state of the puzzle. The single dimensional array's initial state and goal state passed from input are converted into 2-D arrays (3 by 3) and then manipulations like calculating manhattan heuristic distance, performing move operations on board are done on them. Here the '0' in the array represents the hole in the board but it is printed as empty space while printing interim states and the goal state.

Puzzlenode class tracks the parent node, level cost, heuristic cost and evaluation cost of the search node.

EightpuzzleAstar class contains the various methods like A star search on 8-game puzzle board, checking the board is equal to goal board, priority queue for maintaining open list of nodes.

Execution Results:

Example 1:



5 4

6 1 8

7 3 2

5 4

6 1 8

7 3 2

5 1 4

6 8

7 3 2

5 1 4

6 3 8

7 2

5 1 4

6 3 8

7 2

5 1 4

6 3

7 2 8

5 1 4

6 3

7 2 8

5 1 4

6 2 3

7 8

5 1 4

6 2 3

7 8

5 1 4

2 3

6 7 8

1 4

5 2 3

6 7 8

1 4

5 2 3

6 7 8

1 2 4

5 3

6 7 8

1 2 4

5 3

6 7 8

2 4

1 5 3

6 7 8

2 4

1 5 3

6 7 8

2 4

1 5 3

6 7 8

2 4 3

1 5

6 7 8

2 4 3

1 5

6 7 8

2 3

1 4 5

6 7 8

2 3

1 4 5

6 7 8

1 2 3

4 5

6 7 8

1 2 3

4 5

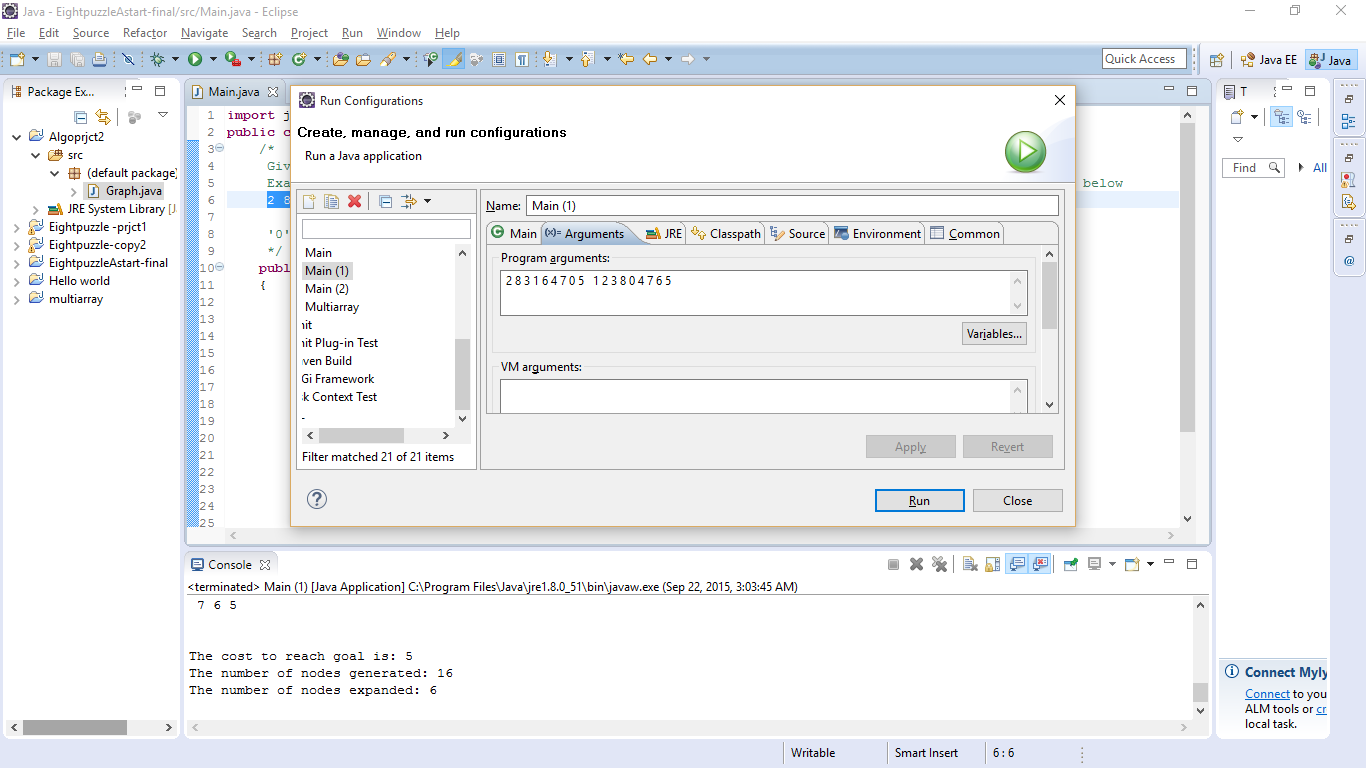
6 7 8

The cost to reach goal is: 22

The number of nodes generated: 9114

The number of nodes expanded: 3288

Example 2:



Result:

