CMPT 420: Artificial Intelligence
Professor Tian
Project 1: 8-Puzzle Solution using Breadth-First Search

Team Members

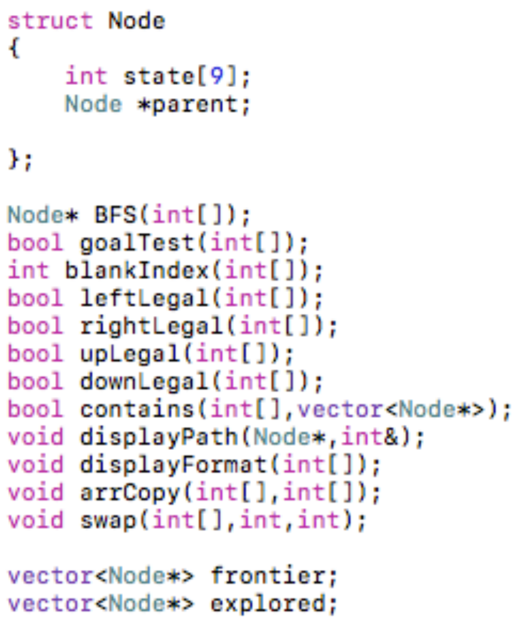
Natalie McGovern
Sanjay Sewbhajan

Board Configuration

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

- Goal state
- Row number
- Column number

Data Structures/Functions

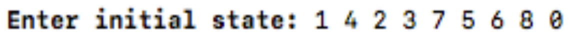


```
struct Node
{
    int state[9];
    Node *parent;

};

Node* BFS(int[]);
bool goalTest(int[]);
int blankIndex(int[]);
bool leftLegal(int[]);
bool rightLegal(int[]);
bool upLegal(int[]);
bool downLegal(int[]);
bool contains(int[],vector<Node*>);
void displayPath(Node*,int&);
void displayFormat(int[]);
void arrCopy(int[],int[]);
void swap(int[],int,int);

vector<Node*> frontier;
vector<Node*> explored;
```

Data Structure/Function	**Description**
struct Node {…};	Stores board state as 1D array. Pointer references parent of board state.
Node* BFS(int[]);	Breadth-First Search algorithm. Accepts initial state as input. Returns goal Node if solution found, NULL otherwise.
bool goalTest(int[]);	Returns true if given state is the goal state.
int blankIndex(int[]);	Returns index of 0 (empty space) within given state.
bool leftLegal(int[]);	Returns true if a given state can make a left move.
bool rightLegal(int[]);	Returns true if a given state can make a right move.
bool upLegal(int[]);	Returns true if a given state can make an up move.
bool downLegal(int[]);	Returns true if a given state can make a down move.
bool contains(int[],vector<Node*>);	Returns true if given state is found within contents of given vector (frontier or explored).
void displayPath(Node*,int&);	Traverses path from goal Node to Node of initial state. Updates path cost.
void displayFormat(int[]);	Displays given state as 3x3 board.
void arrCopy(int[],int[]);	Manual array copy.
void swap(int[],int,int);	Swaps elements at given indices within given state.
vector<Node*> frontier;	Frontier set.
vector<Node*> explored;	Explored set.

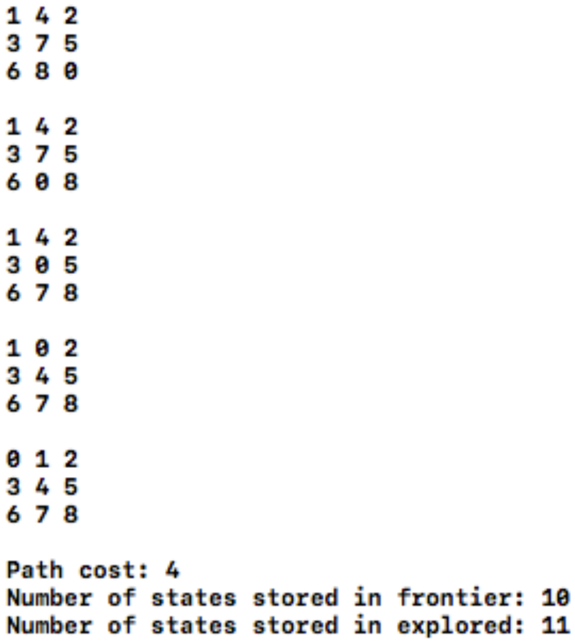
Test Runs

Input



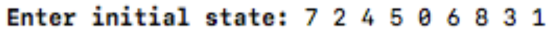
```
Enter initial state: 1 4 2 3 7 5 6 8 0
```

Output



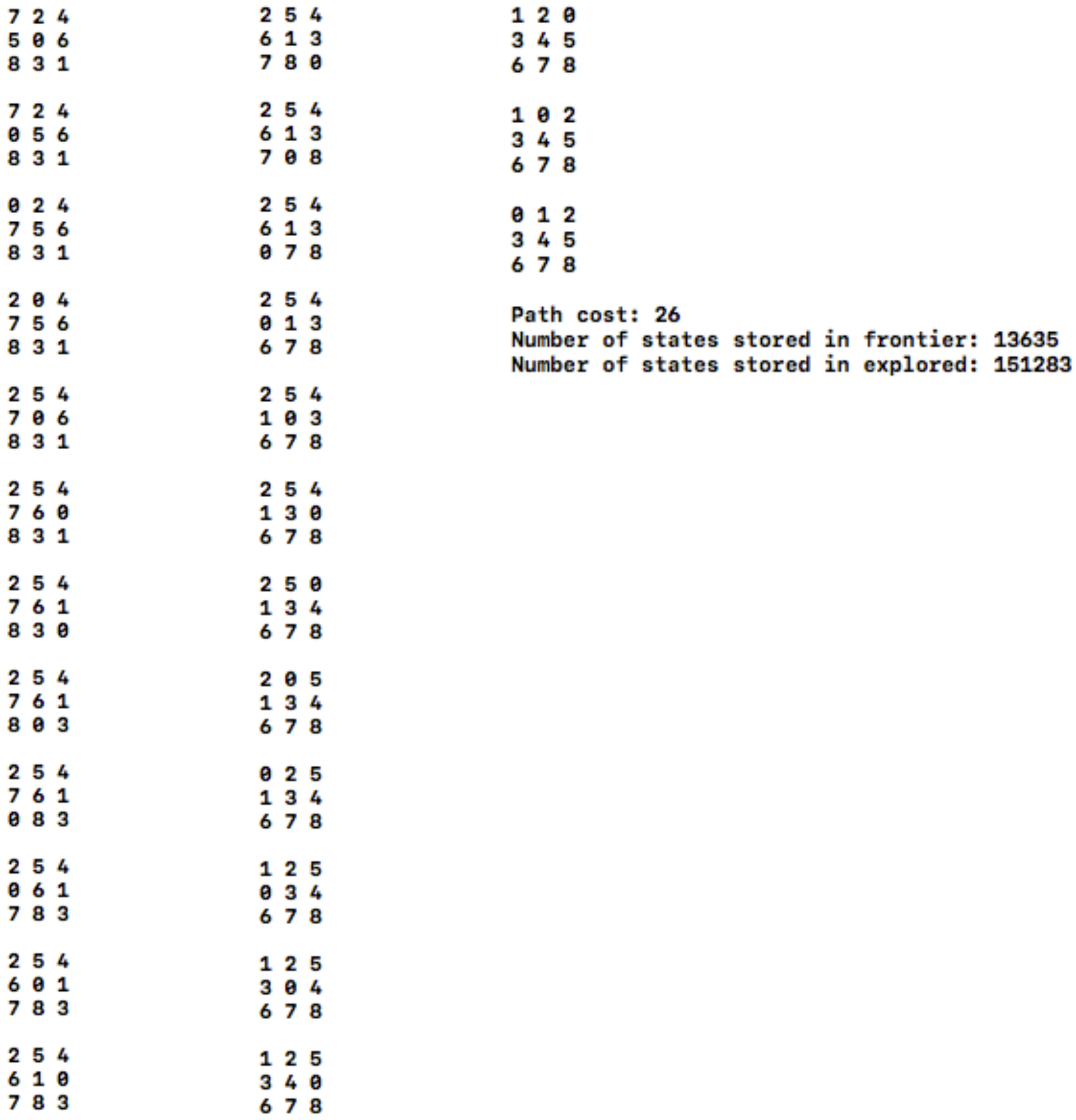
```
1 4 2
3 7 5
6 8 0

1 4 2
3 7 5
6 0 8

1 4 2
3 0 5
6 7 8

1 0 2
3 4 5
6 7 8

0 1 2
3 4 5
6 7 8

Path cost: 4
Number of states stored in frontier: 10
Number of states stored in explored: 11
```

Input



```
Enter initial state: 7 2 4 5 0 6 8 3 1
```

Output



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

7 2 4
0 5 6
8 3 1

0 2 4
7 5 6
8 3 1

2 0 4
7 5 6
8 3 1

2 5 4
7 0 6
8 3 1

2 5 4
7 6 0
8 3 1

2 5 4
7 6 1
8 3 0

2 5 4
7 6 1
8 0 3

2 5 4
7 6 1
0 8 3

2 5 4
0 6 1
7 8 3

2 5 4
6 0 1
7 8 3

2 5 4
6 1 0
7 8 3

2 5 4
6 1 3
7 8 0

2 5 4
6 1 3
7 0 8

2 5 4
6 1 3
0 7 8

2 5 4
0 1 3
6 7 8

2 5 4
1 0 3
6 7 8

2 5 4
1 3 0
6 7 8

2 5 0
1 3 4
6 7 8

2 0 5
1 3 4
6 7 8

0 2 5
1 3 4
6 7 8

1 2 5
0 3 4
6 7 8

1 2 5
3 0 4
6 7 8

1 2 5
3 4 0
6 7 8

1 2 0
3 4 5
6 7 8

1 0 2
3 4 5
6 7 8

0 1 2
3 4 5
6 7 8

Path cost: 26
Number of states stored in frontier: 13635
Number of states stored in explored: 151283
```