

159.302 Assignment #1

Search Algorithms

8-Puzzle Problem

n.h.reyes@massey.ac.nz

Objectives

- Understand and apply abstraction techniques to model the 8-Puzzle problem for use with search algorithms.
- Implement multiple search algorithms (and their variants) to solve the 8-Puzzle problem.
- Evaluate and compare performance of the implemented algorithms and their variants using defined metrics.

Before we discuss the assignment instructions...

One more technique to improve the performance of A*

Tie breaking criteria for A*

- In *A, nodes are expanded in order of increasing f-cost ($f = g + h$).

How to break ties?

- When multiple nodes have the same f-cost, tie-breaking strategies can change the order of expansion, even the search performance.

Q:

(10 CBS) (10 DAS) (10 FBAS) (10 GFBAS)

f-cost

Tie breaking criteria for A*



Example #1:

- Here, we have 4 nodes in the priority queue (Q) with identical f-cost values.
- The goal node (**G**) happens to be at the end of the queue.
- How should we break ties to pick the best node for expansion?

Q:

(10 CBS) (10 DAS) (10 FBAS) (10 **GFBAS**)

f-cost

The search node that leads to the goal (G) is at the end of the Q.

Tie breaking criteria for A*

Standard approach:

- If there are multiple partial paths in the Q with the same f-cost, break ties by picking the leftmost element first.
- A* will then expand these search nodes one by one until it reaches the node whose state is the goal

Q:

(10 CBS) (10 DAS) (10 FBAS) (10 GFBAS)

f-cost

The search node that leads to the goal is at the end of the Q.

Tie breaking criteria for A*

Standard approach:

- If there are multiple partial paths in the Q with the same f-cost, break ties by picking the leftmost element first.
- A* will then expand these search nodes one by one until it reaches the node whose state is the goal
- In this particular example, A* will take 4 expansions to find the path to the goal.

Q:

(10 CBS) (10 DAS) (10 FBAS) (10 GFBAS)

f-cost

The search node that leads to the goal is at the end of the Q.

Tie breaking criteria for A*

Standard approach:

- If there are multiple partial paths in the Q with the same f-cost, break ties by picking the leftmost element first.
 - A* will then expand these search nodes one by one until it reaches the node whose state is the goal
-
- In this particular example, A* will take 4 expansions to find the path to the goal
 - Could a different tie-breaking strategy find the optimal path with fewer expansions?

Q:

(10 CBS) (10 DAS) (10 FBAS) (10 GFBAS)

f-cost

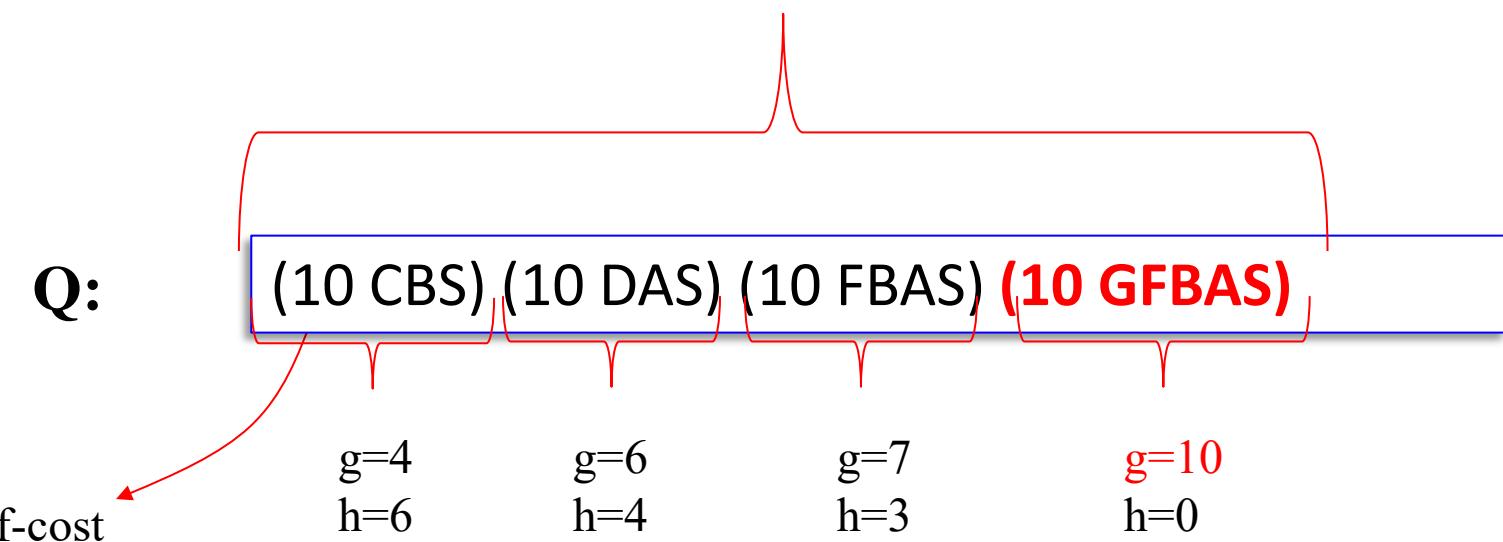
The search node that leads to the goal is at the end of the Q.

Tie breaking criteria for A*

Tie-Breaking in Favour of Larger g-Values

- When multiple nodes have the same f-cost, prioritize expanding the node with the larger g-value to explore deeper paths sooner.

All nodes have the same f-cost

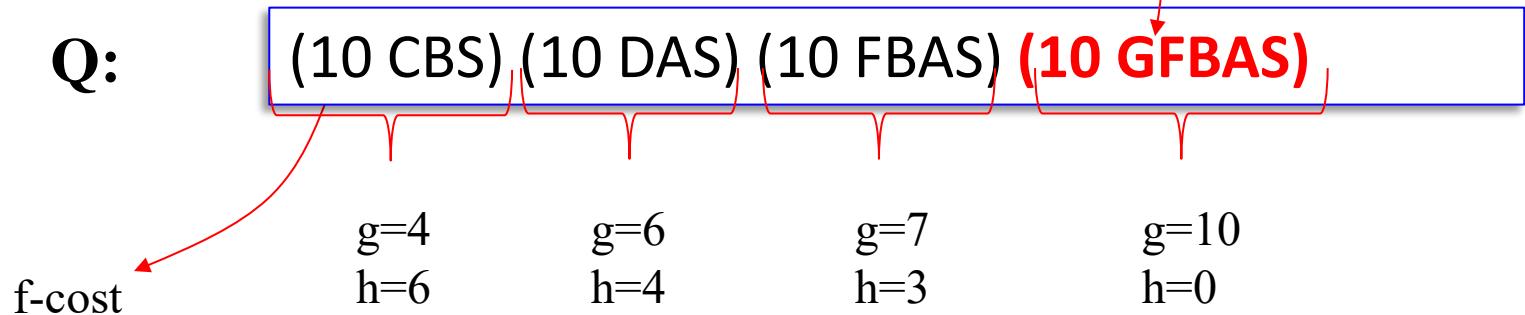


Tie breaking criteria for A*

Tie-Breaking in Favour of Larger g-Values

- When multiple nodes have the same f-cost, prioritize expanding the node with the larger g-value to explore deeper paths sooner.
- In this particular example, A* will take only 1 expansion to find the path to the goal

Q:

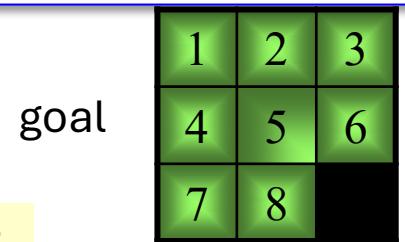


Tie breaking criteria for A*

Tie-Breaking in Favour of Larger g-Values

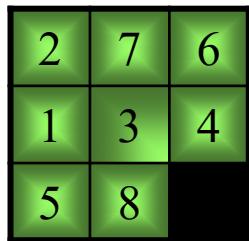
- When multiple nodes have the same f-cost, prioritize expanding the node with the larger g-value to explore deeper paths sooner.

Example #2:



Which of the following nodes in the Q will be expanded first?

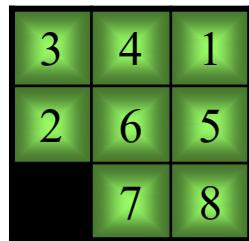
Q (container of partial paths):



$g=10$

$h=12$

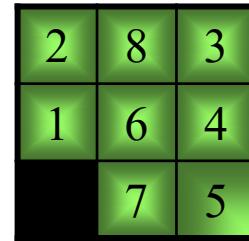
$f=22$



$g=7$

$h=12$

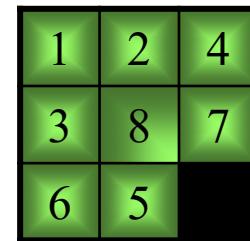
$f=19$



$g=9$

$h=10$

$f=19$



$g=5$

$h=14$

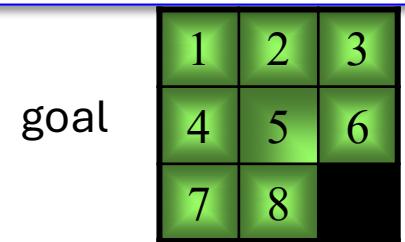
$f=19$

Tie breaking criteria for A*

Tie-Breaking in Favour of Larger g-Values

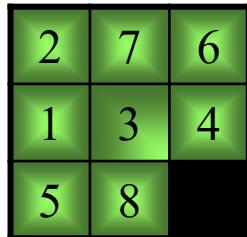
- When multiple nodes have the same f-cost, prioritize expanding the node with the larger g-value to explore deeper paths sooner.

Example #2:

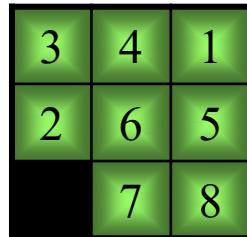


Which of the following nodes in the Q will be expanded first?

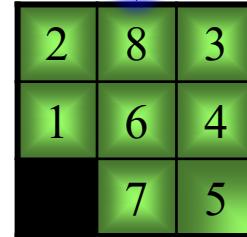
Q (container of partial paths):



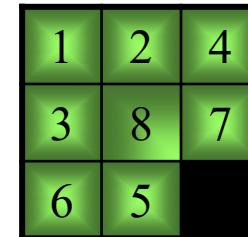
$g=10$
 $h=12$
 $f=22$



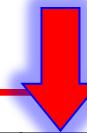
$g=7$
 $h=12$
 $f=19$



$g=9$
 $h=10$
 $f=19$



$g=5$
 $h=14$
 $f=19$



Tie breaking criteria for A*

Apart from using a consistent heuristic and the Strict Expanded List, we can further improve the performance of A* by using a tie-breaking criterion that favours nodes with larger g-values.

Improving the Performance of the A* Algorithm:

1. Consistent Heuristic

- Use a heuristic that satisfies the consistency (monotonicity) property to ensure optimality and reduce the number of node expansions.

2. Strict Expanded List

- Maintain a strict expanded list to prevent revisiting or re-expanding nodes unnecessarily, improving search efficiency.

3. Tie-Breaking in Favour of Larger g-Values

- When multiple nodes have the same f-cost, prioritize expanding the node with the larger g-value to explore deeper paths sooner.

Tie breaking criteria for A*

Apart from using a consistent heuristic and the Strict Expanded List, we can further improve the performance of A* by using a tie-breaking criterion that favours nodes with larger g-values.

Improving the Performance of the A* Algorithm:

1. Consistent Heuristic

- Use a heuristic that satisfies the consistency (monotonicity) property to ensure optimality and reduce the number of node expansions.

2. Strict Expanded List

- Maintain a strict expanded list to prevent revisiting or re-expanding nodes unnecessarily, improving search efficiency.

3. Tie-Breaking in Favour of Larger g-Values

- When multiple nodes have the same f-cost, prioritize expanding the node with the larger g-value to explore deeper paths sooner.

The assignment requires an A* implementation that makes use of all of these performance-enhancing techniques for A*

Tie breaking criteria for A*

Tie-Breaking in Favour of Larger g-Values

This tie-breaking criterion can be implemented using a heap data structure provided by the Standard Template Library (STL). However, you must explicitly define a custom comparator to enforce the tie-breaking rule.

Adding an element to Heap:

- `make_heap()`
 - `push_back()`
 - `push_heap()`
-
- ```
graph LR; A[make_heap()] --> C[User-defined comparator]; B[push_back()] --> C; C[push_heap()] --> C;
```

User-defined  
comparator

# Tie breaking criteria for A\*

## Tie-Breaking in Favour of Larger g-Values

This tie-breaking criteria can be implemented in a Heap data structure, as provided by the Standard Template Library (STL). However, you need to explicitly define a custom comparator that satisfies the tie-breaking criterion.

### Adding an element to Heap:

- `make_heap()`
- `push_back()`
- `push_heap()`

User-defined  
comparator

### Popping an element from the Heap:

- `pop_heap()`
- Retrieve an element
- `pop_back()`

# Selection of Algorithms

- UC with the Expanded List
- A\* with the Strict Expanded List and Tie-Breaking in Favor of Larger g-Values (h=MisplacedTiles)
- A\* with the Strict Expanded List and Tie-Breaking in Favor of Larger g-Values (h= Sum of Manhattan distances)

# Experiments and Analysis

- Test all algorithms on a set of Initial-Goal state pairs.
- Compare the algorithms' performance.
- Use the given test routines (batch\_run mode) to perform the experiments.
- Run the algorithms until:
  - a solution is found/Q turned empty
- Tabulate the results in an Excel file (for submission).

## **8-Puzzle**

**Start-up codes for Assignment #1**

# Start-up Codes

The startup code uses a simple graphics engine to facilitate the visualization of the algorithm in action.

1. On **Windows** machines, the built-in **Graphics Device Interface (GDI)** within the OS will be utilized; no additional libraries need to be installed.
2. On **macOS** and **Linux** machines, the **SDL2** and **SDL\_bgi** libraries will be used. These libraries require installation first. This presentation will demonstrate how to go about the installation process.
3. A **cross-platform** startup code will be provided to the class.

# Start-up Code (Windows version)

# SEARCH

## Start-up Codes: makefile

The start-up system is comprised of multiple files.

Executable filename

**Search.exe** : Main.o graphics.o puzzle.o algorithm.o  
→ g++ -std=c++11 -o **Search.exe** Main.o graphics.o puzzle.o algorithm.o -l gdi32 - static-libgcc -static-libstdc++

**Main.o** : Main.cpp graphics.h puzzle.h algorithm.h  
→ g++ -std=c++11 -c -Wno-write-strings Main.cpp

**puzzle.o** : puzzle.cpp puzzle.h  
→ g++ -std=c++11 -c -Wno-write-strings puzzle.cpp

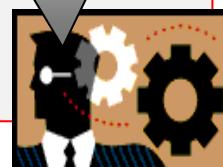
**algorithm.o** : algorithm.cpp algorithm.h  
→ g++ -std=c++11 -c -Wno-write-strings algorithm.cpp

**graphics.o** : graphics.cpp graphics.h  
→ g++ -std=c++11 -c -Wno-write-strings graphics.cpp

**clean:**  
del \*.o  
del \*.exe

Must be preceded by a **tab**.

The program requires that you “**build**” it using a **makefile**.



# SEARCH

## Start-up Codes: makefile (Windows version)

```
Search.exe : Main.o graphics.o puzzle.o algorithm.o
g++ -std=c++11 -o Search.exe Main.o graphics.o puzzle.o algorithm.o -l gdi32 -
static-libgcc -static-libstdc++

Main.o : Main.cpp graphics.h puzzle.h algorithm.h
g++ -std=c++11 -c -Wno-write-strings Main.cpp

puzzle.o : puzzle.cpp puzzle.h
g++ -std=c++11 -c -Wno-write-strings puzzle.cpp

algorithm.o : algorithm.cpp algorithm.h
g++ -std=c++11 -c -Wno-write-strings algorithm.cpp

graphics.o : graphics.cpp graphics.h
g++ -std=c++11 -c -Wno-write-strings graphics.cpp

clean:
del *.o
del *.exe
```

The program uses a simple graphics library (**graphics.cpp** and **graphics.h**) and **links** to the built-in graphics device interface (**gdi32**) of the Windows operating system. Because of this, the program only compiles in the Windows environment.



# Start-up Code (Linux and macOS version)

# SEARCH

## Start-up Codes: makefile (Linux and macOS version)

```
search.out: main.o puzzle.o algorithm.o
g++ -O2 -std=c++11 -l/usr/local/include -L/usr/local/lib -o search.out main.o puzzle.o
algorithm.o -ISDL_bgi -ISDL2
```

```
main.o : main.cpp puzzle.h algorithm.h
g++ -O2 -std=c++11 -l/usr/local/include -c -Wno-write-strings main.cpp
```

```
puzzle.o : puzzle.cpp puzzle.h
g++ -O2 -std=c++11 -c -Wno-write-strings puzzle.cpp
```

```
algorithm.o : algorithm.cpp algorithm.h
g++ -O2 -std=c++11 -c -Wno-write-strings algorithm.cpp
```

```
clean:
rm *.o
rm *.out
```

Link to the **SDL2** graphics library

The Linux and macOS version of the program uses the **SDL2** and **SDL\_bgi** graphics libraries. These libraries need to be installed first before your program can utilize them.



# SEARCH

## Start-up Codes: makefile (Linux and macOS version)

```
search.out: main.o puzzle.o algorithm.o
g++ -O2 -std=c++11 -l/usr/local/include -L/usr/local/lib -o search.out main.o puzzle.o
algorithm.o -ISDL_bgi -ISDL2
```

```
main.o : main.cpp puzzle.h algorithm.h
g++ -O2 -std=c++11 -l/usr/local/include -c -Wno-write-strings main.cpp
```

```
puzzle.o : puzzle.cpp puzzle.h
g++ -O2 -std=c++11 -c -Wno-write-strings puzzle.cpp
```

```
algorithm.o : algorithm.cpp algorithm.h
g++ -O2 -std=c++11 -c -Wno-write-strings algorithm.cpp
```

```
clean:
rm *.o
rm *.out
```

Links to the SDL\_bgi  
graphics library



# SEARCH

## Start-up Codes: makefile (cross-platform version)

```
1 CC := g++
2 TARGET := search
3
4 # Detect the operating system
5 ifeq ($(OS),Windows_NT)
6 CFLAGS := -O2 -std=c++11 -Wall -c
7 LFLAGS := -lgdi32
8
9 EXTENSION := .exe
10 CLEANUP := del
11 CLEANUP_OBJS := del *.o
12
13
14
15 # Find all source files (.cpp) and header files (.h)
16 SRCS := $(wildcard *.cpp) $(wildcard */*.cpp)
17 HDRS := $(wildcard *.h) $(wildcard */*.h)
18
19 else
20 UNAME_S := $(shell uname -s)
21 ifeq ($(UNAME_S),Darwin)
22 # macOS
23 EXTENSION := .out
24 CFLAGS := -O2 -std=c++11 -Wall -I/usr/local/include -L/usr/local/lib -c -Wno-write-strings
25 LFLAGS := -L/usr/local/lib -lSDL_bgi -lSDL2
26 CLEANUP := rm -f
27 CLEANUP_OBJS := rm -f *.o
28
29 # Find all source files (.cpp) and header files (.h)
30 SRCS := main.cpp puzzle.cpp algorithm.cpp
31 HDRS := main.h puzzle.h algorithm.h
32 else ifeq ($(UNAME_S),Linux)
33 # Linux
34 EXTENSION := .out
35 CFLAGS := -O2 -std=c++11 -Wall -I/usr/local/include -L/usr/local/lib -c -Wno-write-strings
36 LFLAGS := -lSDL_bgi -lSDL2
37 CLEANUP := rm -f
38 CLEANUP_OBJS := rm -f *.o
39
40 # Find all source files (.cpp) and header files (.h)
41 SRCS := main.cpp puzzle.cpp algorithm.cpp
42 HDRS := main.h puzzle.h algorithm.h
43 endif
44
45
46
47 # Create object file names based on source file names
48 OBJS := $(SRCS:.cpp=.o)
49
50
51 # Rule to build the executable
52 $(TARGET)$(EXTENSION): $(OBJS)
53 $(CC) -O2 -std=c++11 -o $@ $(OBJS) $(LFLAGS)
54
55 # Rule to build object files
56 %.o: %.cpp $(HDRS)
57 $(CC) $(CFLAGS) $< -o $@
```

For Windows

For macOS

For Linux

I am providing a cross-platform makefile that works on Windows, Linux and macOS machines

 back



# Prerequisites (for macOS machines)

# SEARCH

## SDL2 and SDL\_bgi Installation (macOS)

- requires HOMEBREW package manager installation first
- Compiler: Apple clang version 15.0.0 (clang-1500.3.9.4)



### Homebrew Installation:

1. Open a terminal
2. Install homebrew by copying and pasting the following command:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

The same command can be found from  
the website of homebrew: <https://brew.sh/>



# SEARCH



## SDL2 and SDL\_bgi Installation (macOS)

- After installing the HOMEBREW package manager, we can use it to install SDL2.

### 1. Install SDL2 using Homebrew.

Open a terminal.

Type the command: **brew install SDL2**

comments: This will install SDL2 library files inside the following folders:

- /usr/local/include/SDL2
- /usr/local/lib

### 2. Download source file of SDL BGI from our Stream website.

link: <https://stream.massey.ac.nz/mod/resource/view.php?id=5569626&redirect=1>

Source file: [SDL\\_bgi-3.0.0.tar.gz](#)

Reference: <https://sdl-bgi.sourceforge.io/>

# SEARCH

## SDL2 and SDL\_bgi Installation (macOS)

- Next, we need to build the `SDL_bgi` library from source.

### 3. Decompress the (source file) archive.

Source file: `SDL_bgi-3.0.0.tar.gz`

This will generate a folder, named: `SDL_bgi-3.0.0`

### 4. Compile `SDL_bgi` from sources

To compile and install `SDL\_bgi`, start Terminal and run the following commands:

```
% cd SDL_bgi-3.0.0/src
% make
% make install
```



`libSDL_bgi.so`

# SEARCH

## SDL2 and SDL\_bgi Installation (macOS)

### SAMPLE COMPIILATION RESULTS

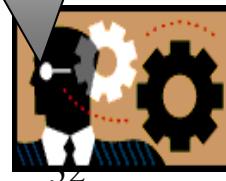
```
it092024:src napoleon$ make
make: python: Command not found
usage: mkdir [-pv] [-m mode] directory_name ...
*** Building on Darwin ***
```

```
gcc -O2 -std=gnu99 -g -c -Wall -I /usr/local/include -I /usr/local/include/SDL2 -fPIC SDL_bgi.c
gcc -shared -o libSDL_bgi.so SDL_bgi.o -ISDL2 ; \
 strip -x libSDL_bgi.so
```

```
it092024:src napoleon$ make install
make: python: Command not found
usage: mkdir [-pv] [-m mode] directory_name ...
*** Building on Darwin ***
```

```
/usr/bin/install -m 755 libSDL_bgi.so /usr/local/lib ; \
 /usr/bin/install -m 644 SDL_bgi.h /usr/local/include/SDL2 ; \
 /usr/bin/install graphics.h /usr/local/include ; \
/usr/bin/install/doc/graphics.3.gz /usr/share/man/man3/
install: /usr/share/man/man3/: No such file or directory
make: *** [install] Error 71
```

Note: we will not be using python, so don't worry about the error reports.



# SEARCH

## SDL2 and SDL\_bgi Installation (macOS)

- LIST OF FILES (and their destination folders) THAT WILL BE INSTALLED IN YOUR MACHINE:

File:

Directory:

- `SDL_bgi.h` -> `/usr/local/include/SDL2`
- `SDL.h` -> `/usr/local/include/SDL2`
- `graphics.h` -> `/usr/local/include`
- `libSDL_bgi.so` -> `/usr/local/lib`
- `libSDL2.a` -> `/usr/local/lib`

# SEARCH



## Building and Running the Test Code (macOS and Linux)

- Download **test code** from our stream website.
  - link: <https://stream.massey.ac.nz/mod/resource/view.php?id=5569627&redirect=1>
- 
1. Open a terminal where the **test code** resides
  2. Type the following commands:
    - `make clean`
    - `make`
  3. After a successful compilation and linking, an executable named **search.out** will be generated.
  4. Run the executable.
    - `./search.out`

# Prerequisites (for Linux machines)

# SEARCH

## SDL2 Installation (Linux)

- Similar to macOS machines, Linux machines also require both SDL2 and SDL\_bgi to run the start-up codes.

### Steps:

#### 1. Download [SDL2-2.28.1.tar.gz](#)

- link: <https://github.com/libsdl-org/SDL/releases/tag/release-2.28.1>

#### 2. Decompress [SDL2-2.28.1.tar.gz](#)

#### 3. Open a terminal run the following commands in succession:

- cd [SDL2-2.28.1.tar.gz](#)
- cd build
- ./configure
- make
- sudo make install

# SEARCH

## SDL\_bgi Installation (Linux)

### Steps:

1. **Download the source file of SDL BGI from our Stream website.**
  - link: <https://stream.massey.ac.nz/mod/resource/view.php?id=5569626&redirect=1>
  - Reference: <https://sdl-bgi.sourceforge.io/>
  - Source file: **SDL\_bgi-3.0.0.tar.gz**
2. **Move `SDL_bgi-3.0.0.tar.gz` to your desired folder.** In this example, I have moved it inside the folder, named: **/home/nhreyes/Documents/BGI\_linux/**
3. **Decompress the (source file) archive.** This will generate a folder, named: **SDL\_bgi-3.0.0**
  - e.g. In this example, I have decompressed it inside the folder, named: **/home/nhreyes/Documents/BGI\_linux/SDL\_bgi-3.0.0/**

# SEARCH

## SDL\_bgi Installation (Linux)

### Steps:

#### **4. Compile the source files.**

- To compile and install `SDL\_bgi`, start a Terminal (i.e. command prompt) and run the following commands:
  - `cd SDL_bgi-3.0.0/src`
  - `make`
  - `sudo make install`

# SEARCH

## SDL\_bgi Installation (Linux)

### Steps:

5. **Manually copy [SDL\\_bgi.h](#) file into the appropriate folder, so the compiler can see it.**

e.g. I have compiled SDL\_bgi inside a folder that is specific to my machine, named:  
[`/home/nhreyes/Documents/BGI\_linux/SDL\_bgi-3.0.0/src/`](#)

Therefore, the command below needs to be modified, to refer to the correct folder in your machine.

```
sudo cp /home/nhreyes/Documents/BGI_linux/SDL_bgi-3.0.0/src/SDL_bgi.h /usr/local/include/SDL2/
```

source

destination

# SEARCH

## SDL2 and SDL\_bgi Installation (Linux)

- The following is the summary of files that correspond to the libraries and their location in my system:

File:

Directory:

- `SDL_bgi.h` -> `/usr/local/include/SDL2`
- `graphics.h` -> `/usr/include`
- (originally intended, but this does not apply) `graphics.h` -> `/usr/local/include`
- `libSDL_bgi.so` -> `/usr/lib`
- (originally intended, but this does not apply) `libSDL_bgi.so` -> `/usr/local/lib`

# SEARCH



## Building and Running the Test Code (macOS and Linux)

- Download **test code** from our stream website.
  - link: <https://stream.massey.ac.nz/mod/resource/view.php?id=5569627&redirect=1>
- 
1. Open a terminal where the **test code** resides
  2. Type the following commands:
    - `make clean`
    - `make`
  3. After a successful compilation and linking, an executable named **search.out** will be generated.
  4. Run the executable.
    - `./search.out`

# Start-up Codes

# SEARCH

## Start-up Codes: algorithm implementation

- Place your algorithm implementations inside **algorithm.cpp**
- Include the function prototypes inside **algorithm.h** (already pre-defined)

### algorithm.cpp

```
#include "algorithm.h"
using namespace std;

///////////////
// Search Algorithm: A* using the Strict Expanded List
///////////////

string aStar_ExpandedList (string const initialState, string const goalState, int &pathLength,
int &numOfStateExpansions, int &maxQLength, float &actualRunningTime,
int &numOfDeletionsFromMiddleOfHeap, int &numOfLocalLoopsAvoided,
int &numOfAttemptedNodeReExpansions, heuristicFunction heuristic) {

 // put your implementation codes here...

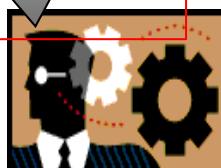
}
```

Please don't change the function names and their formal parameter definitions as the program refers to them when batch runs are executed to perform the experiments.

Place your algorithm implementations inside these files.

- Skeleton functions are provided in the start-up codes.

 back



# SEARCH

## Start-up Codes: function prototypes for algorithms

### algorithm.h

```
#ifndef __ALGORITHM_H__
#define __ALGORITHM_H__

#include <ctime>
#include <string>
#include <iostream>
#include <algorithm>
#include <cstdlib>
#include <cstring>
#include <vector>

#include "puzzle.h" //optional only, may be removed

////////////////////////////

//Function prototypes

string uc_explist(string const initialState, string const goalState, int& pathLength, int &numOfStateExpansions, int& maxQLength,
 float &actualRunningTime, int &numOfDeletionsFromMiddleOfHeap, int &numOfLocalLoopsAvoided, int &numOfAttemptedNodeReExpansions);

string aStar_ExpandedList(string const initialState, string const goalState, int& pathLength, int &numOfStateExpansions, int& maxQLength,
 float &actualRunningTime, int &numOfDeletionsFromMiddleOfHeap, int &numOfLocalLoopsAvoided, int &numOfAttemptedNodeReExpansions, heuristicFunction heuristic);

#endif
```

function name

Formal parameters

Feel free to add more variables, functions, classes,  
etc. inside **algorithm.cpp** and **algorithm.h**.

However, you are not allowed to change the **names**  
and **formal parameters** of the algorithms. The batch  
experiment routines refer to them to automate the  
execution of the **experiments**.

 back



# SEARCH

## Main function (arguments)

The program requires some arguments.

```
int main(int argc, char* argv[]){

 string path;

 if(argc < 3){
 cout << "<< SEARCH ALGORITHMS >>" << endl;
 cout << "please include missing parameters." << endl;
 cout << "SYNTAX #1: search.exe <TYPE_OF_RUN = \"batch_run\" or \"single_run\" or \"animate_run\"> ALGORITHM_NAME \"INITIAL STATE\" \"GOAL STATE\" " << endl;
 cout << "SYNTAX #2: search.exe <TYPE_OF_RUN = \"batch_run\"> ALGORITHM_NAME" << endl;
 exit(_Code:0);
 }
}
```

Calling the program:

search.exe **type\_of\_run** [algorithm] [start\_state] [goal\_state]

Mode of execution

The main function parameters allow the program to switch between a number of modes of execution.



back

# SEARCH

## Main function (arguments)

The program requires some arguments.

```
int main(int argc, char* argv[]){

 string path;

 if(argc < 3){
 cout << "<< SEARCH ALGORITHMS >>" << endl;
 cout << "please include missing parameters." << endl;
 cout << "SYNTAX #1: search.exe <TYPE_OF_RUN = \"batch_run\" or \"single_run\" or \"animate_run\"> ALGORITHM_NAME \"INITIAL STATE\" \"GOAL STATE\" " << endl;
 cout << "SYNTAX #2: search.exe <TYPE_OF_RUN = \"batch_run\"> ALGORITHM_NAME" << endl;
 exit(_Code:0);
 }
}
```

search.exe **type\_of\_run** [algorithm] [start\_state] [goal\_state]

Mode of execution

1. **single\_run** run a single algorithm, and show results in text mode

2. **animate\_run** run a single algorithm, and animate solution

3. **batch\_run** run a one or more algorithms in text mode

# SEARCH

## Main function (arguments)

The program requires some arguments.

```
int main(int argc, char* argv[]){

 string path;

 if(argc < 3){
 cout << "<< SEARCH ALGORITHMS >>" << endl;
 cout << "please include missing parameters." << endl;
 cout << "SYNTAX #1: search.exe <TYPE_OF_RUN = \"batch_run\" or \"single_run\" or \"animate_run\"> ALGORITHM_NAME \"INITIAL STATE\" \"GOAL STATE\" " << endl;
 cout << "SYNTAX #2: search.exe <TYPE_OF_RUN = \"batch_run\"> ALGORITHM_NAME" << endl;
 exit(_Code: 0);
 }
}
```

As an example, we can use the **animate\_run** parameter setting, to instruct the program to apply the A\* algorithm (with the strict expanded list, and Sum of Manhattan distances as heuristic) to solve a given problem instance of the puzzle. Once the solution is found, it will display an animation of how the puzzle is solved.

```
C:> search.exe "animate_run" astar_explist_manhattan "608435127" "123456780"
```

# SEARCH

## Main function (arguments)

The program requires some arguments.

```
int main(int argc, char* argv[]){

 string path;

 if(argc < 3){
 cout << "<< SEARCH ALGORITHMS >>" << endl;
 cout << "please include missing parameters." << endl;
 cout << "SYNTAX #1: search.exe <TYPE_OF_RUN = \"batch_run\" or \"single_run\" or \"animate_run\"> ALGORITHM_NAME \"INITIAL STATE\" \"GOAL STATE\" " << endl;
 cout << "SYNTAX #2: search.exe <TYPE_OF_RUN = \"batch_run\"> ALGORITHM_NAME" << endl;
 exit(_Code: 0);
 }
}
```

Here are some examples of how we can run the program.

search.exe “**single\_run**” “**astar\_explist\_manhattan**” “042158367” “012345678”

search.exe “**animate\_run**” “**astar\_explist\_manhattan**” “042158367” “012345678”

search.exe “**batch\_run**” “**astar\_explist\_manhattan**”

search.exe “**batch\_run**” “**all**”



## How to run the program using the **single\_run** mode?

```
single_run <algorithm>
```

Run an algorithm using a single pair of (start-goal) combination

# SEARCH

## Main function (required arguments)

Running the program:

Example:

```
search.exe "single_run" "astar_explist_manhattan" "042158367" "012345678"
```

type of run = "batch\_run" or "single\_run" or "animate\_run"

Algorithm name

Algorithm name (you are not allowed to change these names)

"uc\_explist"

"astar\_explist\_manhattan"

"astar\_explist\_misplacedtiles"

INITIAL STATE

GOAL STATE

For testing a single algorithm on a single pair of (start-goal) pair

50

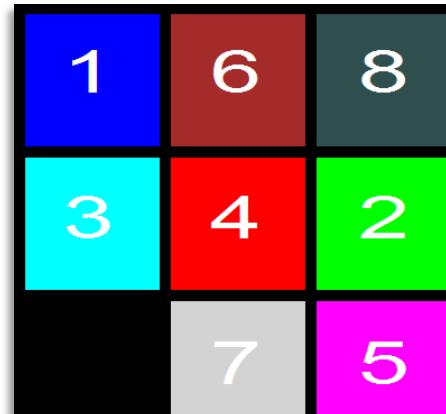
back



## How to run the program using the **animate\_run** mode?

```
animate_run <algorithm>
```

Run an algorithm using a single pair of (start-goal) combination, and see the animation of solution.



# SEARCH

## Main function (required arguments)

Running the program:

Example:

```
search.exe "animate_run" "astar_explist_manhattan" "042158367" "012345678"
```

Algorithm name

INITIAL STATE

GOAL STATE

type of run = "batch\_run" or "single\_run" or "animate\_run"

Algorithm name (you are not allowed to change these names)

- "uc\_explist"
- "astar\_explist\_manhattan"
- "astar\_explist\_misplacedtiles"

For testing a single algorithm on a single pair of (start-goal) pair

52



# SEARCH

## Animated Solution ([run\\_one.bat](#))

Using graphics to show the animated solution

```
if (algorithmSelected == "uc_explist") {

 path = uc_explist(initialState, goalState, pathLength, numOfStateExpansions, maxQLength,
 actualRunningTime, numOfDeletionsFromMiddleOfHeap, numOfLocalLoopsAvoided,
 numOfAttemptedNodeReExpansions);

}
...and so on, and so forth...
```

```
if(typeOfRun == "animate_run"){
 if (pathLength == 0) cout << "\n\n*---- NO SOLUTION fo

 if (path != "") {
 AnimateSolution(initialState, goalState, path);
 }
}
```

e.g. path="URRDLURRDLR"

The “animate\_run” parameter invokes the AnimateSolution function.

An animation function is provided for you.



Skeleton function



back

# SEARCH

## Animated Solution ([animate\\_run](#))

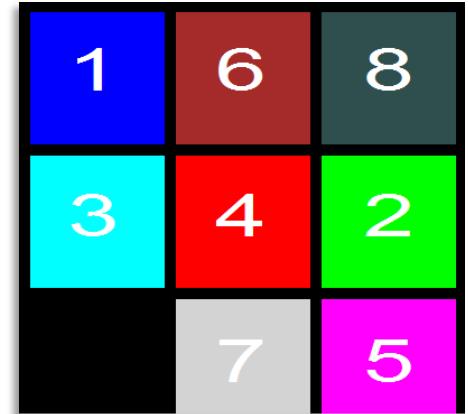
astar\_explist\_manhattan

===== << EXPERIMENT RESULTS >> =====

Initial state: 608435127  
Goal state: 123456780

Path Length: 23  
Num of State Expansions: 662  
Max Q Length: 392  
Actual Running Time: 0.190000  
Num of Deletions from MiddleOfHeap: 9  
Num of Attempted Node ReExpansions: 683

press any key to start.



./search "animate\_run" astar\_explist\_manhattan "608435127" "123456780"

Try [animate\\_run](#)

◀ back



## How to run the program using the **batch\_run** mode?

```
batch_run <algorithm>
```

Run an algorithm using a **set** of (start-goal) state combinations

# SEARCH

## Main function (required arguments)

Running the program:

Example:

```
search.exe "batch_run" "astar_explist_manhattan"
```

type of run = "batch\_run" or "single\_run" or "animate\_run"

Algorithm name (you are not allowed to change these names)

"uc\_explist"  
"astar\_explist\_manhattan"  
"astar\_explist\_misplacedtiles"

Algorithm name

**Batch runs** are provided to run multiple experiments in one call to the program.

There is a **predefined set of puzzle problems** that will be used in the batch run:

```
string list_of_initialStates[5] = { "120483765","208135467","704851632","536407182","638541720" };
string goalState = { "123456780" };
```

 back



## How to run the program using the **batch\_run all** mode?

**batch\_run <all>**

- Run multiple algorithms using a **set** of (start-goal) state combinations.
- We will use this program feature for generating the complete experiment results for submission.

# SEARCH

## Main function (required arguments)

Running the program:

Example:

```
search.exe "batch_run" "all"
```

All algorithms

Algorithm name (you are not allowed to change these names)

"uc\_explist"  
"astar\_explist\_manhattan"  
"astar\_explist\_misplacedtiles"

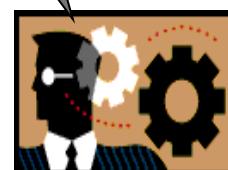
**batch\_runs all** will run all the experiments required for testing all the algorithms in your program.

type of run = "batch\_run" or "single\_run" or "animate\_run"

58

There is a **predefined set of puzzle problems** that will be used in the batch run:

```
string list_of_initialStates[5] = { "120483765","208135467","704851632","536407182","638541720" };
string goalState = { "123456780" };
```



# SEARCH

## batch\_run all

### Running the algorithms

#### Batch run

- This type of run is used for running all the experiments required for all the algorithms.
- Results are tabulated nicely, according to the requirements of the assignment.

Example:

| ALGORITHM            | INIT_STATE | GOAL_STATE | PATH_LENGTH | STATE_EXPANSIONS | MAX_QLENGTH | RUNNING_TIME | DELETIONS_MIDDLE_HEAP | LOCAL_LOOPS_AVOIDED | ATTEMPTED_REEXPANSIONS | PATH                   |
|----------------------|------------|------------|-------------|------------------|-------------|--------------|-----------------------|---------------------|------------------------|------------------------|
| uniform_cost_search  | ,120483765 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| uniform_cost_search  | ,208135467 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| uniform_cost_search  | ,704851632 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| uniform_cost_search  | ,536407182 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| uniform_cost_search  | ,638541720 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_misplacedtiles | ,120483765 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_misplacedtiles | ,208135467 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_misplacedtiles | ,704851632 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_misplacedtiles | ,536407182 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_misplacedtiles | ,638541720 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_manhattan      | ,120483765 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_manhattan      | ,208135467 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_manhattan      | ,704851632 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_manhattan      | ,536407182 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |
| astar_manhattan      | ,638541720 | ,123456780 | ,21         | ,17              | ,115        | ,0.000000    | ,0                    | ,0                  | ,0                     | ,DDRRLLLUUURDLUDURDLUU |

**Extra information only!** - to help with algorithm implementation analysis.

Convert the results from text to Excel file format. (see template.xlsx)



back

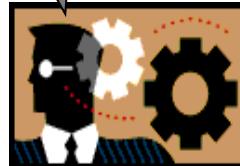
# Tabulation of Experiment Results

Sample only

| ALGORITHM            | INIT_STATE | GOAL_STATE | PATH_LENGTH | STATE_EXPANSIONS | MAX_QLENGTH | RUNNING_TIME | DELETIONS_MIDDLE_HEAP | LOCAL_LOOPS_AVOIDED | ATTEMPTED_REEXPANSIONS | PATH              |
|----------------------|------------|------------|-------------|------------------|-------------|--------------|-----------------------|---------------------|------------------------|-------------------|
| uniform_cost_search  | ,120483765 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| uniform_cost_search  | ,208135467 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| uniform_cost_search  | ,704851632 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| uniform_cost_search  | ,536407182 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| uniform_cost_search  | ,638541720 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_misplacedtiles | ,120483765 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_misplacedtiles | ,208135467 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_misplacedtiles | ,704851632 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_misplacedtiles | ,536407182 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_misplacedtiles | ,638541720 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_manhattan      | ,120483765 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_manhattan      | ,208135467 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_manhattan      | ,704851632 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_manhattan      | ,536407182 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |
| astar_manhattan      | ,638541720 | ,123456780 | ,21         | ,17              | ,115        | ,0.00000     | ,0                    | ,0                  | ,0                     | ,DRLRLUUURDUDRLUU |

- Your assignment will be marked based on:
  - PATH\_LENGTH
  - STATE\_EXPANSIONS
  - MAX\_QLENGTH
  - RUNNING\_TIME
- The **extra** information appearing under the headings DELETIONS\_MIDDLE\_HEAP, LOCAL\_LOOPS\_AVOIDED, and ATTEMPTED\_REEXPANSIONS will **not** be used for marking your assignment. They are only meant to help with algorithm implementation analysis.

Your assignment will be marked based on the results of your experiments.



# Tabulation of Experiment Results

Sample only

Use the format **results\_ID.xlsx** as the name of your Excel file

Example: (e.g., **results\_20298765.xlsx**).

ID number

| A                       | B          | C          | D           | E                | F           | G            | H                     | I                   | J                      | K                       |
|-------------------------|------------|------------|-------------|------------------|-------------|--------------|-----------------------|---------------------|------------------------|-------------------------|
| ALGORITHM               | INIT_STATE | GOAL_STATE | PATH_LENGTH | STATE_EXPANSIONS | MAX_QLENGTH | RUNNING_TIME | DELETIONS_MIDDLE_HEAP | LOCAL_LOOPS_AVOIDED | ATTEMPTED_REEXPANSIONS | PATH                    |
| 1 uniform_cost_search   | 120483765  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 2 uniform_cost_search   | 208135467  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 3 uniform_cost_search   | 704851632  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 4 uniform_cost_search   | 536407182  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 5 uniform_cost_search   | 638541720  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 6 uniform_cost_search   | 120483765  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 7 astar_misplacedtiles  | 120483765  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 8 astar_misplacedtiles  | 208135467  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 9 astar_misplacedtiles  | 704851632  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 10 astar_misplacedtiles | 536407182  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 11 astar_misplacedtiles | 638541720  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 12 astar_manhattan      | 120483765  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 13 astar_manhattan      | 208135467  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 14 astar_manhattan      | 704851632  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 15 astar_manhattan      | 536407182  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 16 astar_manhattan      | 638541720  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 17                      | results    |            |             |                  |             |              |                       |                     |                        |                         |

Results must be fully replicable using your submitted codes.

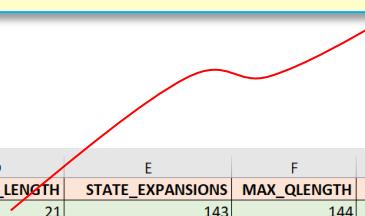
Your assignment will be marked based on the results of your experiments.

The worksheet must be named “results”



# Tabulation of Experiment Results

Enter the required information in the designated cells without altering their position.



|    | A                    | B          | C          | D           | E                | F           | G            | H                     | I                   | J                      | K                       |
|----|----------------------|------------|------------|-------------|------------------|-------------|--------------|-----------------------|---------------------|------------------------|-------------------------|
| 1  | ALGORITHM            | INIT_STATE | GOAL_STATE | PATH_LENGTH | STATE_EXPANSIONS | MAX_QLENGTH | RUNNING_TIME | DELETIONS_MIDDLE_HEAP | LOCAL_LOOPS_AVOIDED | ATTEMPTED_REEXPANSIONS | PATH                    |
| 2  | uniform_cost_search  | 120483765  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 3  | uniform_cost_search  | 208135467  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 4  | uniform_cost_search  | 704851632  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 5  | uniform_cost_search  | 536407182  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 6  | uniform_cost_search  | 638541720  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 7  | astar_misplacedtiles | 120483765  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 8  | astar_misplacedtiles | 208135467  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 9  | astar_misplacedtiles | 704851632  | 123456780  | 21          | 143              | 144         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 10 | astar_misplacedtiles | 536407182  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 11 | astar_misplacedtiles | 638541720  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 12 | astar_manhattan      | 120483765  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 13 | astar_manhattan      | 208135467  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 14 | astar_manhattan      | 704851632  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 15 | astar_manhattan      | 536407182  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 16 | astar_manhattan      | 638541720  | 123456780  | 21          | 123              | 148         | 0            | 0                     | 0                   | 0                      | 0DDRRLLLLUUURDLUDURDLUU |
| 17 |                      |            |            |             |                  |             |              |                       |                     |                        |                         |

This file will be used to accurately analyze your algorithm implementation and assess your assignment.

Failure to perform the required experiments and submit this file will result in a **50% grade deduction**.

## Sample parameter settings

1. search "single\_run" astar\_explist\_manhattan "608435127" "123456780"
2. search "animate\_run" astar\_explist\_manhattan "608435127" "123456780"
3. search "batch\_run" astar\_explist\_manhattan
4. search "batch\_run" astar\_explist\_misplacedtiles
5. search "batch\_run" uc\_explist
6. search "batch\_run" all

S

|   |   |   |
|---|---|---|
| 6 | 2 | 8 |
|   | 3 | 5 |
| 4 | 7 | 1 |

The **states** are the board configurations

- There are  $9!$  states (362,880 states) but not all of them are reachable from a given board position.

S

|   |   |   |
|---|---|---|
| 6 | 2 | 8 |
|   | 3 | 5 |
| 4 | 7 | 1 |

A **move** corresponds to moving the blank '0'-tile

- The successors of a state are to be generated in a **FIXED** order, namely move the blank tile: **Up**, **Right**, **Down**, then **Left**.

Up

|   |   |   |
|---|---|---|
|   | 2 | 8 |
| 6 | 3 | 5 |
| 4 | 7 | 1 |

Right

|   |   |   |
|---|---|---|
| 6 | 2 | 8 |
| 3 |   | 5 |
| 4 | 7 | 1 |

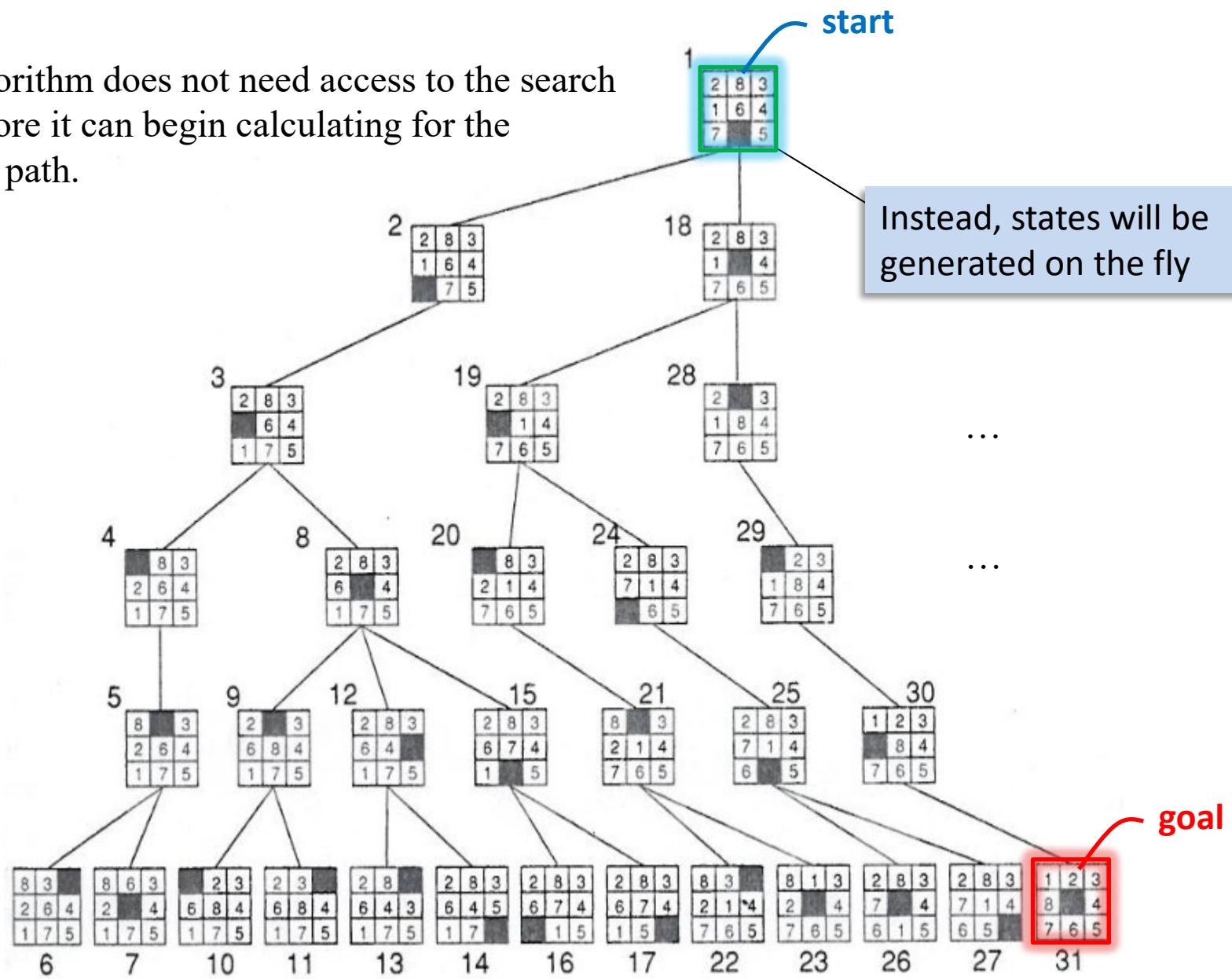
Down

|   |   |   |
|---|---|---|
| 6 | 2 | 8 |
| 4 | 3 | 5 |
|   | 7 | 1 |

Left



The algorithm does not need access to the search tree before it can begin calculating for the shortest path.



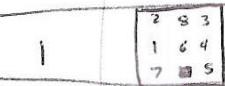
Example: Using Depth-limited Depth-First Search

DFS (depth bound = 5)

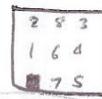
(CLOSED LIST)

Step

8



$d=1$



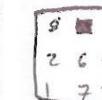
$d=2$



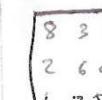
$d=3$



$d=4$



$d=5$

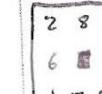
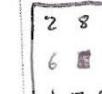


no more expansions!

7



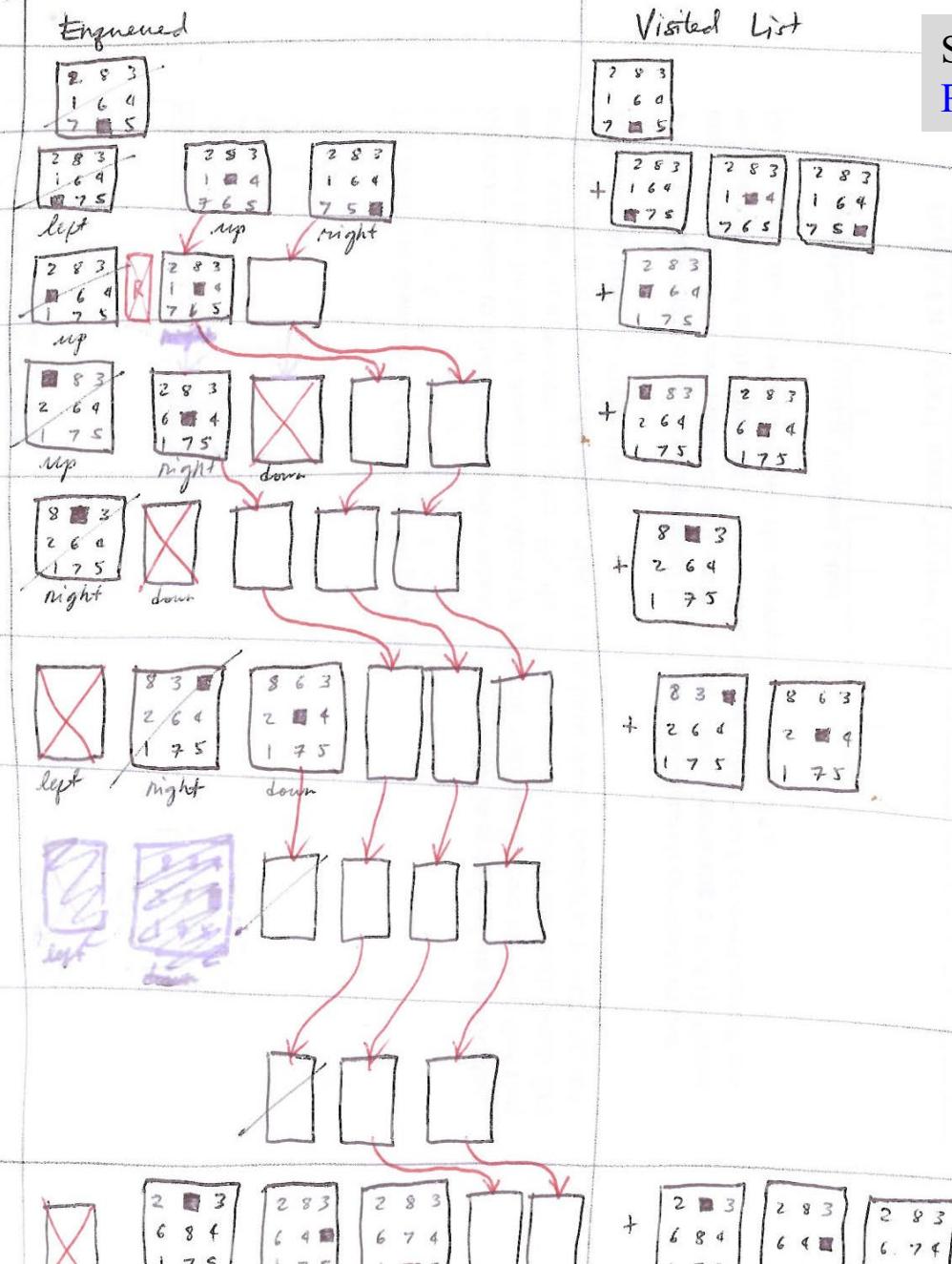
$d=5$ !



## Example: Depth-limited Depth-First Search

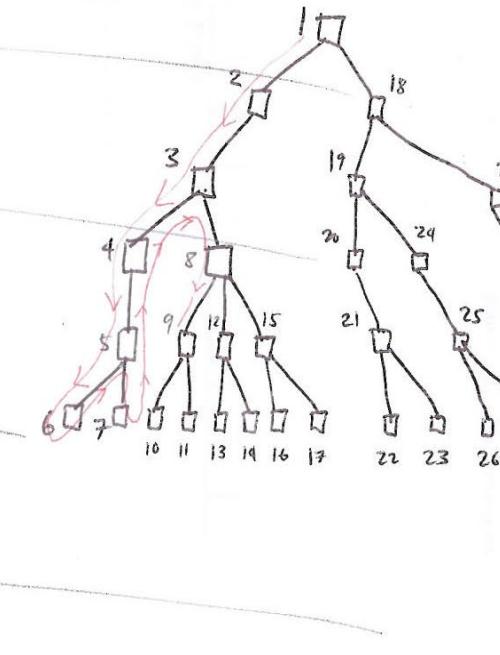
8 PUZZLE

Successor generator: Left, Up, Right, Down



Store each state as a triple:

- STATE
- PARENT
- LENGTH OF PATH



## class **Puzzle** (Optional – you may choose not to use this)

In the start-up codes, I am providing a class named **Puzzle** to give the class **some clue** on how to approach the problem using an object-oriented programming style.

```
class Puzzle{

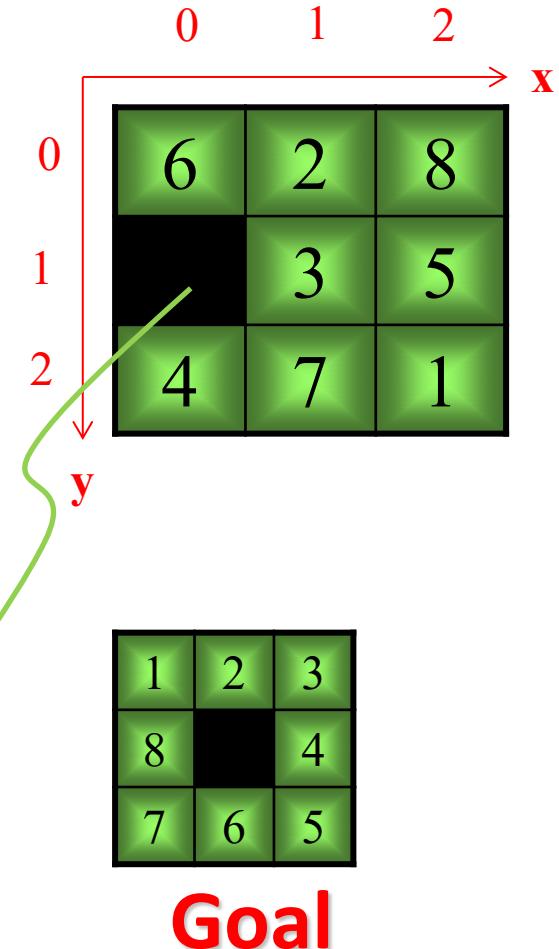
private:

 string path; //moves generated (e.g. URDL..)
 int pathLength;
 int hCost;
 int fCost; } //used by A*
 int depth; //may be used by PDS

 int goalBoard[3][3]; //goal board configuration

 int x0, y0; //coordinates of the blank or 0-tile

 int board[3][3]; //current board configuration
```



Feel free to add more variables and functions that you may require.

## class **Puzzle** (Optional – you may choose not to use this)

The class **Puzzle** is not complete, if you intend to use it in your assignment, you will have to fill-in the missing functions.

```
void Puzzle::updateFCost(){
 //fCost = ?
}
```

Note that it is not required to use the class **Puzzle** when implementing the different algorithms - you may design your own class from scratch.

This class is only optional. You may choose not to use it in your program.

```
//Heuristic function implementation
int Puzzle::h(heuristicFunction hFunction){

 int sum=0;
 int h=0;
 int numOfMisplacedTiles=0;

 switch(hFunction){
 case misplacedTiles:
 //place your implementation here
 //h = ??? //numOfMisplacedTiles;
 break;
 case manhattanDistance:
 //place your implementation here
 //h = ??? //sum of manhattan distance;
 break;
 };
 return h;
}
```



```
Puzzle(const Puzzle &p); //constructor; creates a copy of a given puzzle
Puzzle(string const elements, string const goal); //constructor; generates 2D-
array versions corresponding to an initial board configuration and a goal
configuration.
```

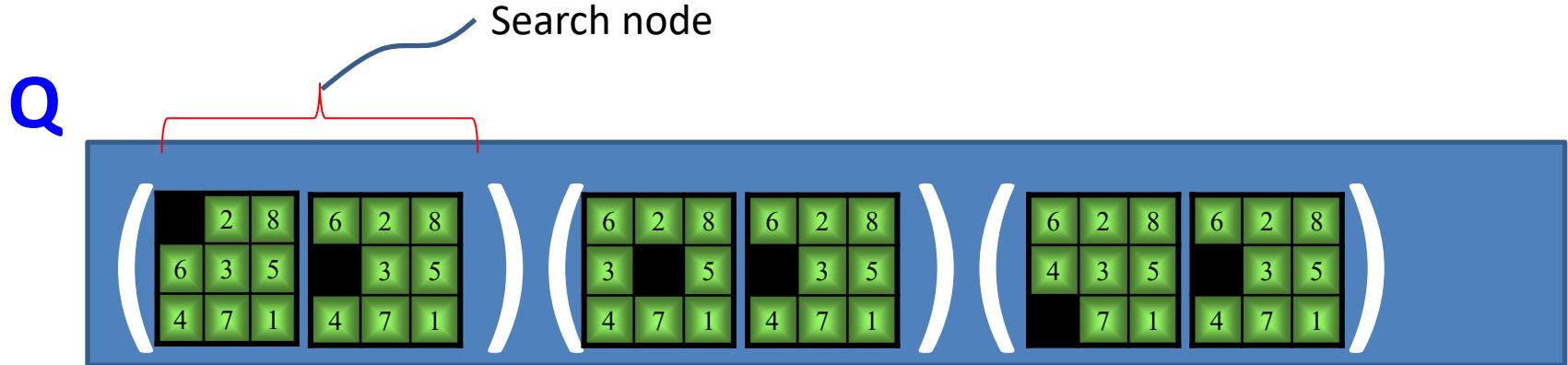
```
void printBoard(); //prints the content of a puzzle
int h(heuristicFunction hFunction); //computes for the heuristic value of a state
void updateFCost(); //computes for the f-cost of a search node
bool goalMatch(); //returns true if the current board configuration matches the
goal state; otherwise, false
```

```
bool canMoveLeft();
bool canMoveRight();
bool canMoveUp();
bool canMoveDown();
```

//checks to see if the blank **0-tile** could  
be moved in a specific direction (codes  
provided)

```
Puzzle * moveUp();
Puzzle * moveRight();
Puzzle * moveDown();
Puzzle * moveLeft();
```

//creates an instance of a Puzzle (including  
memory allocation), representing the result of  
the move taken. It also keeps track of the  
sequence of moves taken (or the partial path).  
The function returns a Puzzle object. (codes  
provided)



- a **Q** is a container for the partial paths (**search nodes**) being considered for expansion.
- depending on the algorithm you are implementing, you may be needing a **Queue**, **Stack** or a **Priority Queue** to implement this **Q** container.

When extending a node, moveUp(), moveRight(), moveDown(), moveLeft() comes handy.

Within a search node, there shouldn't be any repeating nodes/no local loops allowed (i.e. you may have to add variables and test functions to do this, depending on the algorithm you are implementing).

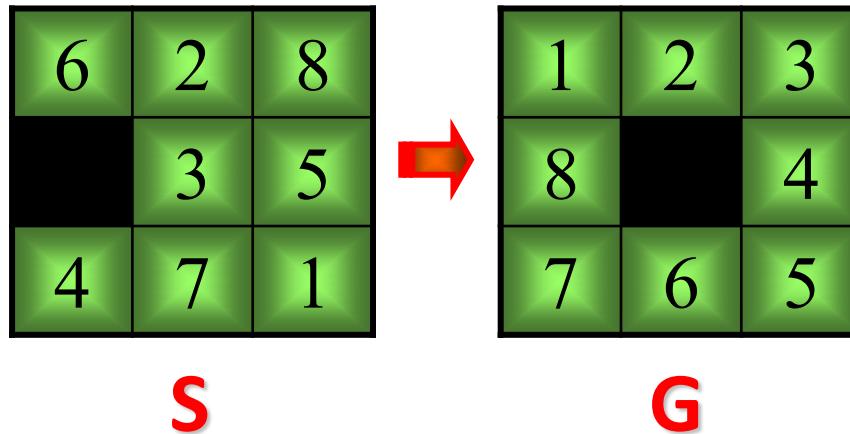
# UC and the A\* Algorithm

- Implement the Q container using the **heap data structure implementation** available in the C++ Standard Template Library (STL).
- You can convert a **vector** into a **heap** using: `make_heap()`, `push_heap()`, `pop_heap()`, etc.
- Use a custom-built comparator in conjunction with all these heap functions.

# SEARCH

## Admissible Heuristics

**8 Puzzle:** Move tiles to reach goal. Think of a move as moving an “empty” tile.



Alternative underestimates of “distance” (number of moves) to goal:

1. Number of **misplaced tiles** (7 in the example above).
2. Sum of **Manhattan Distance** of tile to its goal location (17 in example above).

Manhattan Distance between  $(x_1, y_1)$  and  $(x_2, y_2)$  is  $|x_1 - x_2|$  and  $|y_1 - y_2|$ . Each move can only decrease the distance of exactly one tile.

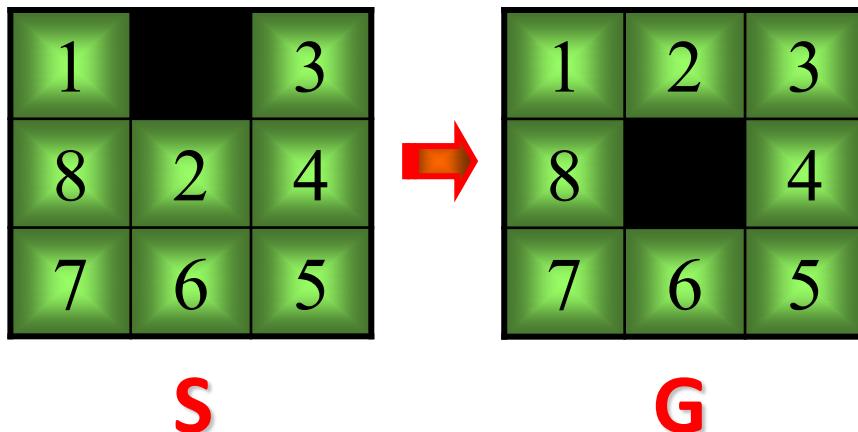
The second of these is much better at predicting actual number of moves.

# SEARCH

## Admissible Heuristics

**8 Puzzle:** Move tiles to reach goal. Think of a move as moving an “empty” tile.

### 1. Number of misplaced tiles



Are we supposed to include the blank tile in the counting of misplaced tiles?

No.

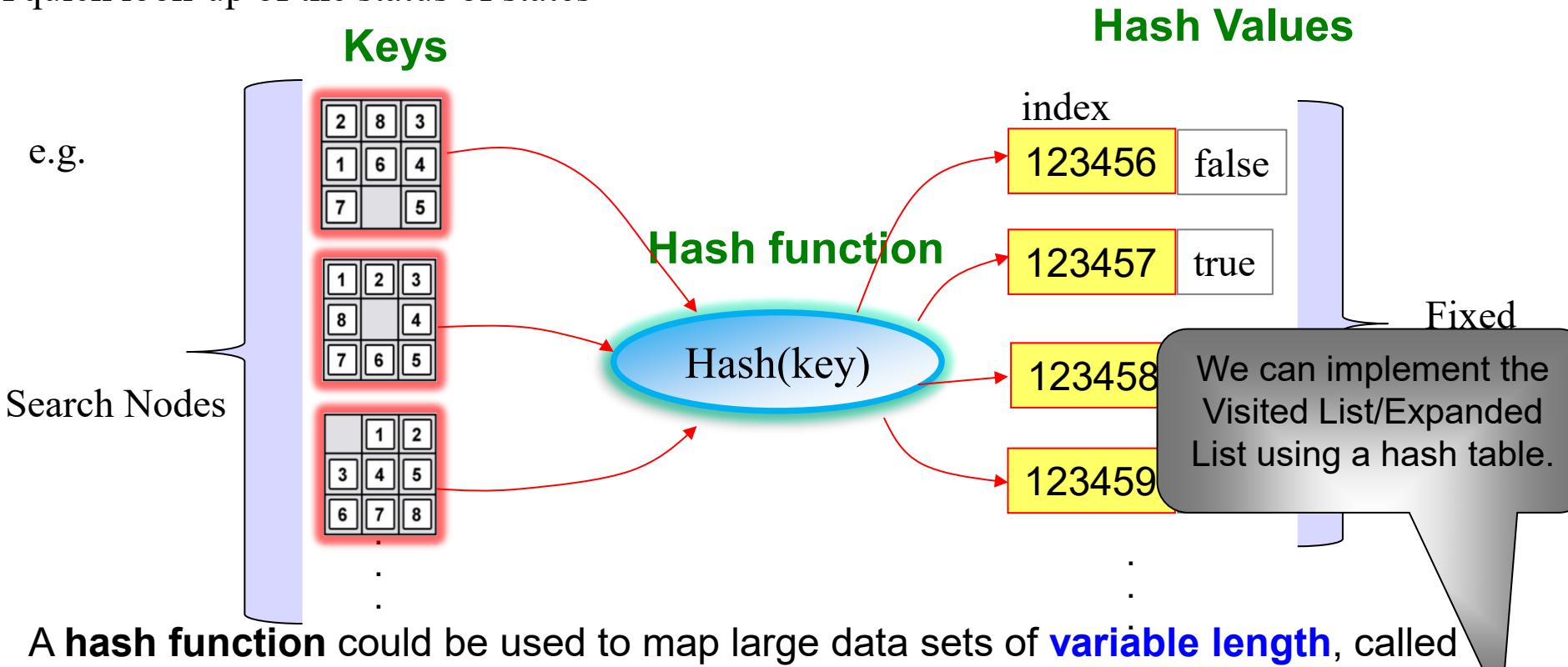
What will be the consequence if we include it? Examine the case above.

The heuristic will no longer be admissible. A\* will miss finding the optimal solution if the heuristic is not admissible.

# SEARCH

## Hash function

A quick look-up of the status of states



A **hash function** could be used to map large data sets of **variable length**, called **keys** to **hash values** that are of a **fixed length**. The hash values may serve as indexes to an associative array.

However, hash values may have duplicates; therefore, may cause collision between keys.

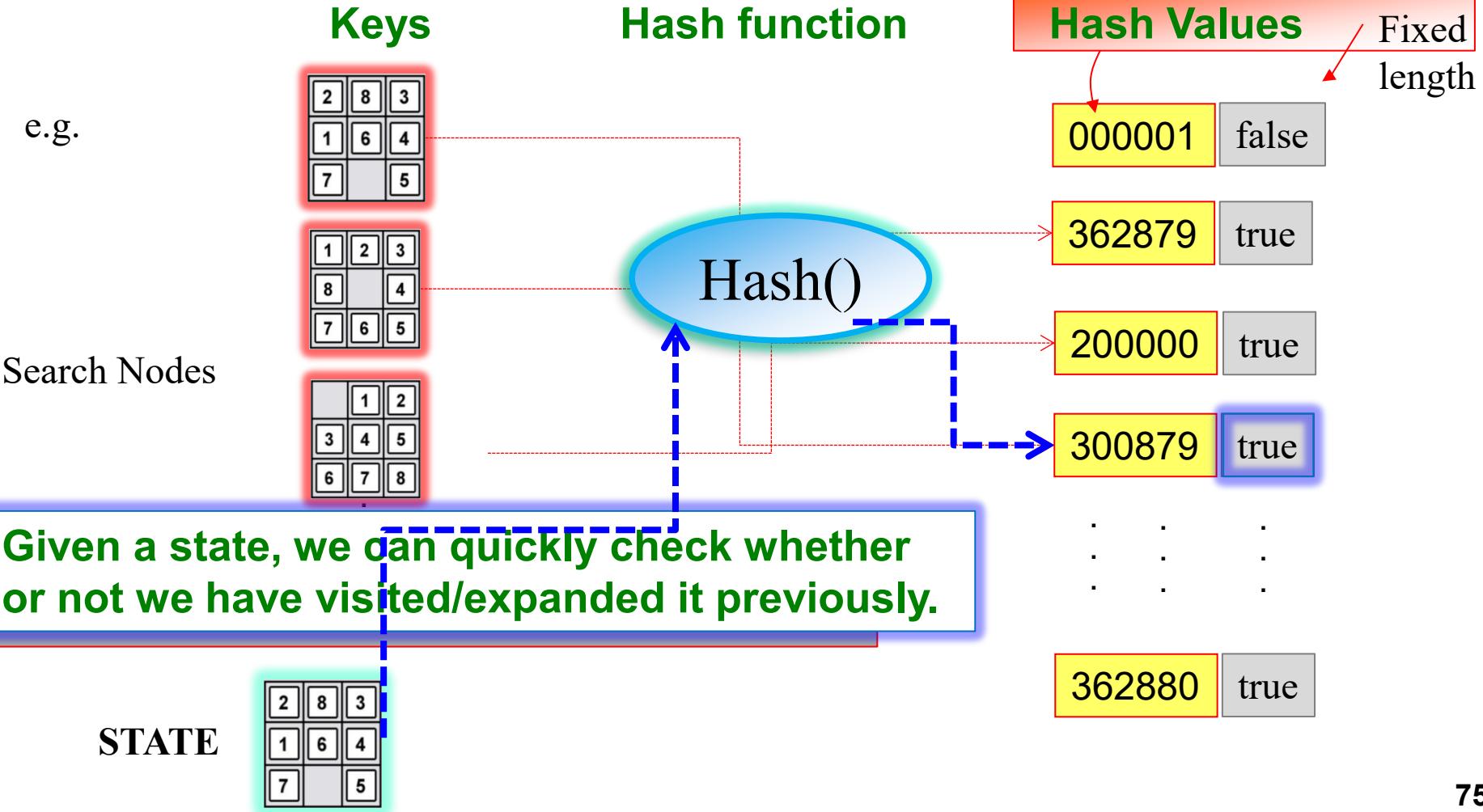


# SEARCH

## Hash function

## Visited List / Expanded List

A quick look-up of the status of states



# **Running and Documenting the Experiments**

**How to save the experiment results to an  
Excel worksheet?**

# **Checklist**

**Please accomplish the following checklist  
to allow for accurate marking of your  
assignment.**

Download the file named **checklist.xlsx** from our Stream website.

# Checklist.xlsx

| CHECKLIST |                                                                                                    |                                                                                                                                                                                                                                                                            |         |           |           |                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-----------|-----------|------------------------------------------------------------------------------------------------------|
|           |                                                                                                    | Description                                                                                                                                                                                                                                                                |         |           |           | Instructions                                                                                         |
| 1         | <b>STUDENT(S)</b><br><b>STUDENT_MEMBER_1</b><br><b>STUDENT_MEMBER_2</b><br><b>STUDENT_MEMBER_3</b> | Group members/Individual details                                                                                                                                                                                                                                           | Surname | Firstname | ID number | Instructions                                                                                         |
| 2         | OS                                                                                                 |                                                                                                                                                                                                                                                                            |         |           |           | Write names and IDs of all members in a group (maximum of 3 members in a group) or single individual |
| 3         | COMPILER                                                                                           |                                                                                                                                                                                                                                                                            |         |           |           |                                                                                                      |
| 4         | IDE                                                                                                |                                                                                                                                                                                                                                                                            |         |           |           |                                                                                                      |
| Item      | Description                                                                                        | Instructions                                                                                                                                                                                                                                                               |         |           |           |                                                                                                      |
| 2         | Operating System(s) used for testing your FTP server codes                                         | Indicate the OS used for testing your FTP server. Note that your cross-platform FTP server must be able to run on Windows 10. If you have a different OS other than Windows 10 or Windows 11, you may use Massey's Lab machine (via VPN) to test your codes in Windows 11. |         |           |           |                                                                                                      |
| 3         | Compiler used for developing the assignment                                                        | g++ v.15.1.0 (or later), g++ v.8.1 (Massey's Lab machine), or clang 15.0 (or later) is acceptable. Your program will be compiled and marked using one of them.                                                                                                             |         |           |           |                                                                                                      |
| 4         | IDE used for developing the assignment                                                             | Provide the IDE name and version (e.g. SublimeText 4, Visual Studio Code v.1.76.2, CLion 2022.1.3)                                                                                                                                                                         |         |           |           |                                                                                                      |

# Checklist.xlsx

|    |                                                                                  |                                                                                                                                                      |                   |                                                                                                                                                                                                                         |
|----|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5  | <b>MAKEFILE</b>                                                                  | <b>makefile:</b> Have you included the makefile required to build the executable (i.e. compile and link) for your codes?                             | Yes/No            | Indicate 'Yes' or 'No'. This file is required for marking, even if you have used Visual Studio Code, Clion, etc. to develop your codes. The <b>makefile</b> should suffice for building the executable from your codes. |
| 6  | <b>SOURCE_CODES</b>                                                              | <b>Source Codes (*.cpp, *.h):</b> Have you included all your source code files?                                                                      | Yes/No            | Indicate 'Yes' or 'No'. This is very important, as your program will be recompiled. If the program does not compile you will receive a failing mark.                                                                    |
| 7  | <b>SUPPORT_FILES</b>                                                             | <b>Other Support Files:</b> Have you submitted other supporting files required to compile your codes and test it? i.e. *.json, *.sln, .vcxproj, etc. | Yes/No            | Indicate 'Yes' or 'No'. Make sure that you submit all the supporting files if you developed your codes using CLion or Visual Studio. This will only be used as a back-up approach to building your codes.               |
| 7  | <b>NAMES_OF_SUPPORT_FILES</b>                                                    | Names of Support Files.                                                                                                                              |                   | If you are submitting other supporting files, please specify their filenames (e.g. tasks.json, *.sln, etc.)                                                                                                             |
| 8  | <b>ALGORITHMS</b>                                                                | <b>Required Functionalities</b>                                                                                                                      |                   | <b>Instructions</b>                                                                                                                                                                                                     |
|    | <b>uc_explist</b>                                                                | <b>Uniform Cost Search with Strict Expanded List</b> implementation                                                                                  | full/partial/none | Indicate 'full', if you have completed the implementation of the required algorithm, 'partial', if you are only submitting a partial implementation, or 'none', if not accomplished.                                    |
|    | <b>astar_explist_manhattan</b>                                                   | <b>A* Search with Strict Expanded List (<math>h=\text{Sum of Manhattan distance}</math>)</b> implementation                                          | full/partial/none |                                                                                                                                                                                                                         |
|    | <b>astar_explist_misplacedtiles</b>                                              | <b>A* Search with Strict Expanded List (<math>h=\text{Misplaced Tiles}</math>)</b> implementation                                                    | full/partial/none |                                                                                                                                                                                                                         |
| 9  | <b>Is your program able to run using the <b>batch_run</b> all mode provided?</b> | <b>batch_run</b> must execute all the required algorithms and solve all given start-state combinations.                                              | Yes/No            | Indicate 'Yes' or 'No'.                                                                                                                                                                                                 |
| 10 | <b>Experiment Results in Excel Worksheet</b>                                     | <b>Note:</b> Name your Excel file using the following format: <b>results_ID.xlsx</b> ; where ID is your ID number.                                   | Yes/No            | Indicate 'Yes' or 'No'. You will lose 50% of your grade if you fail to perform the required experiments and submit this Excel file.                                                                                     |

|          |           |                     |                                                                                                         |        |                                                                              |  |
|----------|-----------|---------------------|---------------------------------------------------------------------------------------------------------|--------|------------------------------------------------------------------------------|--|
| 22<br>25 | <b>11</b> | <b>Gen AI Usage</b> | Did you use Generative AI in producing your assignment?                                                 | Yes/No | Indicate 'Yes' or 'No'. You may use AI for planning and debugging your code. |  |
|          |           |                     | If yes, specify the AI tool used                                                                        |        | Indicate all the AI tools used (e.g. chatGPT 4o, etc.)                       |  |
|          |           |                     | If yes, submit a separate document detailing: the prompts used and the AI-generated responses received. |        | Provide the file name of the submitted document.                             |  |

Fill-up this table and submit as part of your assignment documentation.