ECE 479/579
Spring 2018

Homework #3
Due: March 30, 2018 (11:59pm) Please note: no late submission will be accepted.

Consider a sliding block puzzle with the following initial configuration:

B	B	B	W	W	W	E

There are three black tiles (B), three white tiles (W), and an empty cell (E). The puzzle has the following moves:

a. A tile may move into an adjacent empty cell with unit cost.
b. A tile may hop over at most two other tiles into an empty cell with a cost equal to the number of tiles hopped over.

Initial	Next	Cost
BBBWWEW	BBBW**EW**W	1
BBBWWEW	BBB**E**W**W**W	1
BBBWWEW	BB**E**WW**B**W	2

The goal of the puzzle is to have all of the white tiles to the left of all of the black tiles (regardless of the position of the blank cell).

You are to:

1. Specify a heuristic function, h, for this problem and show the search tree produced by the algorithm A* using this heuristic function (show the explicit portion of the search tree).
2. Determine and show that your function never overestimates the true cost of reaching a goal or otherwise.
3. Implement a program in C, C++ or Java to solve the sliding block puzzle using your heuristic function.

The input to the program should be given interactively in your console as follows (the example below illustrates the initial state given above):

Enter initial state: EWBWBWB

The Output Format in your console:

EWBWBWB : G(n) = 0 : h(n) = 3
WWBEBWB : G(n) = 2 : h(n) = 2
WWBWBEB : G(n) = 3 : h(n) = 1
WWEWBBB : G(n) = 5 : h(n) = 0
Total cost = 5
Number of states expanded = 11

Submission Requirements

File Format and Naming Rules

- All assignments must be submitted as a **single ZIP (.zip) (RAR is not accepted)** using the naming convention: hw03_NetID.zip. NetID is your UA NetID.
  Ex. My email: kspeng@email.arizona.edu,
  My submission filename: hw03_kspeng.zip.

- Include a **hw03_NetID.pdf** that answers the two written parts.
  1. Specify a heuristic function, h, in formula.
  2. Show that your heuristic function never overestimates.

- Include a README.txt in your submission that specifies:
  - your programming language.
    Ex. Programming: C++
    * If you use C++11, please specify C++11. Ex. Programming: C++
  - your compiling and running procedure:
    Ex. g++ hw03_ NetID.cpp
    ./a.out input.txt

- The submitted archive must use the following directory structure:

```
hw03_NetID
|____\ hw03_NetID.pdf
|____\ README.txt
|____\ src
      |____\ hw03_NetID.c, hw03_NetID.cpp, or hw03_NetID.java
      |____\ any other files related to your work.
```

- The result must be **printed in the console**.

- The source file should be named as following: **hw03_NetID.xxx**

- Do not include any build files, packages, or .exe in the directory.

***** If you do not follow the format and naming rules, we can't recognize your works and you will not get any credit.**

Compiling Rules

- All assignments will be compiled and tested using the ECE server - ece3/computer (ece3.ece.arizona.edu / compute.ece.arizona.edu), on which the CMake tools and java compile tools are already available. You can access the ece3/compute server using any secure shell (SSH) client. All students have your own ID and PASSWORD to access ece3 server. You are strongly encouraged to test your programs using the ece3 server before submission.
    - You can use ssh in command window or putty (download here) to connect to ece3 server.
    - ID and PASSWORD are you current NetID and PASSWORD.
- **Your code must be able to compile and run on ECE server. Otherwise, you will not get any credit in the programming part.**

Grading Rubrics:

0. Follow the naming, output format, and file structure rules.	10%
1. Heuristic function Design	20%
2. Show that your heuristic function never overestimates.	20%
3. Programming Part	
3.1. Be able to compile/run on ece3 (**Must be no error and warning.**)	10%
3.2. Show the correct actions, states, with your heuristic values.	20%
3.3. Show the optimal actions.	20%
*** You have to pass 0. and 3.1. to get the credits of 3.2. and 3.3.**	