

ECE: 579 Artificial Intelligence

HW Assignment #3

David Akre

1. Specify a heuristic function $h(n)$ in formula:

The heuristic function $h(n)$ used in this programming assignment is as follows: $h(n)$ = the number of misplaced tiles. Thus when selecting which node to visit from all expanded nodes in the A* algorithm the following formula applies: $\text{Cost} = G(n) + h(n)$, where $G(n)$ is the cost to get to the next node summed with the heuristic measure of the misplaced nodes to the next node.

2. Show that your heuristic function never overestimates:

The heuristic function never overestimates because every action that results in a smaller amount of misplaced tiles will be taken (e.g. there is an if/else control strategy condition in the code that will not allow future actions to take place if $h(n)$ results in a larger value than the previous or current $h(n)$). Additionally, A* will choose the total cost summed with the $h(n)$ which is lowest as the next path to be explored. Hence, the cost of the path chosen will be the minima of all other states expanded. Thus, the $h(n)$ function will never overestimate the cost of getting to the solution state.

The following is an output example of the code proving this (e.g. build and execution on the ece3 server at the University of Arizona):

```
dakre@compute:~/artificial_intelligence$ cd hw03_dakre/
dakre@compute:~/artificial_intelligence/hw03_dakre$ ls
README.txt  src
dakre@compute:~/artificial_intelligence/hw03_dakre$ cd src/
dakre@compute:~/artificial_intelligence/hw03_dakre/src$ ls
CMakeLists.txt  hw03_dakre.cpp  hw03_dakre.h
dakre@compute:~/artificial_intelligence/hw03_dakre/src$ cd ../
dakre@compute:~/artificial_intelligence/hw03_dakre$ mkdir build
dakre@compute:~/artificial_intelligence/hw03_dakre$ cd build/
dakre@compute:~/artificial_intelligence/hw03_dakre/build$ cmake ../src
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /bin/cc
-- Check for working C compiler: /bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /bin/c++
-- Check for working CXX compiler: /bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
CMake Warning (dev) in CMakeLists.txt:
  No cmake_minimum_required command is present.  A line of code such as

    cmake_minimum_required(VERSION 3.4)

  should be added at the top of the file.  The version specified may be lower
  if you wish to support older CMake versions for this project.  For more
  information run "cmake --help-policy CMP0000".
This warning is for project developers.  Use -Wno-dev to suppress it.

-- Configuring done
-- Generating done
-- Build files have been written to: /home/dakre/artificial_intelligence/hw03_dakre/build
dakre@compute:~/artificial_intelligence/hw03_dakre/build$ make
Scanning dependencies of target hw03_dakre
[ 50%] Building CXX object CMakeFiles/hw03_dakre.dir/hw03_dakre.o
[100%] Linking CXX executable hw03_dakre
[100%] Built target hw03_dakre
dakre@compute:~/artificial_intelligence/hw03_dakre/build$ ./hw03_dakre
Enter initial state: BBBWWWE
BBBWWWE : G(n) = 0 : h(n) = 3
BBBWEW : G(n) = 1 : h(n) = 3
BBBWEW : G(n) = 1 : h(n) = 3
BBEWBWW : G(n) = 2 : h(n) = 2
BBEWBWW : G(n) = 3 : h(n) = 2
BBEWBWW : G(n) = 3 : h(n) = 2
BBWEBWW : G(n) = 3 : h(n) = 2
EBBWBWW : G(n) = 3 : h(n) = 2
BBWWBEW : G(n) = 4 : h(n) = 2
```

```

BBWBWEW : G(n) = 4 : h(n) = 2
BBWEWBW : G(n) = 4 : h(n) = 2
BEBWBWB : G(n) = 4 : h(n) = 2
BBWWEBW : G(n) = 4 : h(n) = 2
EBBWBWB : G(n) = 4 : h(n) = 2
BWBEBWW : G(n) = 4 : h(n) = 2
BEWBBWW : G(n) = 4 : h(n) = 2
BBWBWBE : G(n) = 5 : h(n) = 2
EBWBBWW : G(n) = 5 : h(n) = 2
WBBEBWW : G(n) = 5 : h(n) = 2
BEWBWBW : G(n) = 5 : h(n) = 2
BWBEBBW : G(n) = 5 : h(n) = 2
BEWWBBW : G(n) = 6 : h(n) = 1
BWEWBBW : G(n) = 7 : h(n) = 1
EBWWBBW : G(n) = 7 : h(n) = 1
BWWEBBW : G(n) = 7 : h(n) = 1
EWBWBWW : G(n) = 8 : h(n) = 1
WBEWBBW : G(n) = 8 : h(n) = 1
WBWBBBW : G(n) = 9 : h(n) = 1
BWWWBWE : G(n) = 9 : h(n) = 1
EWWBWW : G(n) = 9 : h(n) = 1
WEBWBWW : G(n) = 9 : h(n) = 1
WWBEBBW : G(n) = 10 : h(n) = 1
WEWBWW : G(n) = 10 : h(n) = 1
BWWWBEB : G(n) = 10 : h(n) = 1
BWWWEBB : G(n) = 10 : h(n) = 1
WWBWWBW : G(n) = 10 : h(n) = 1
WBWWBWE : G(n) = 11 : h(n) = 1
BWWBWBW : G(n) = 11 : h(n) = 1
BWWEWBB : G(n) = 11 : h(n) = 1
BWEWWBB : G(n) = 11 : h(n) = 1
WWBWBWE : G(n) = 12 : h(n) = 1
BWEWBWB : G(n) = 12 : h(n) = 1
BEWWBBB : G(n) = 12 : h(n) = 1
WBWWBEB : G(n) = 12 : h(n) = 1
WBWWEBB : G(n) = 12 : h(n) = 1
BEWBWBW : G(n) = 12 : h(n) = 1
EWBWWBB : G(n) = 12 : h(n) = 1
EWWBWBW : G(n) = 13 : h(n) = 1
EWWBWBW : G(n) = 13 : h(n) = 1
WWBWBEB : G(n) = 13 : h(n) = 1
WWBWEBB : G(n) = 13 : h(n) = 1
EWBWBWB : G(n) = 13 : h(n) = 1
EBWWBBB : G(n) = 13 : h(n) = 1
WBWEBWB : G(n) = 13 : h(n) = 1
WBWEWBB : G(n) = 13 : h(n) = 1
WBEWWBB : G(n) = 13 : h(n) = 1
WEWWBBB : G(n) = 14 : h(n) = 0
Total cost = 14
Number of states expanded = 106
dakre@compute:~/artificial_intelligence/hw03_dakre/build$ date
Sat Mar 24 12:11:09 MST 2018
dakre@compute:~/artificial_intelligence/hw03_dakre/build$ hostname
compute.engr.arizona.edu

```

As one can see $h(n)$ never overestimates the total cost to get to the solution state.