# CS380 HW Assignment 3

## Christopher Geib

## Due: Feb. 13th 11:59pm

### Part 1: Written Problems (40 points)

(This is problem 3.26 of the 3rd Edition of the textbook). Consider the unbounded version of the regular 2D grid shown in Figure 3.9. The start state is at the origin (0,0), and the goal state is at (x,y).

1. What is the branching factor b in this state space?
2. How many distinct states are there at depth k (for k larger than 0).
3. What is the maximum number of nodes expanded by breadth-first tree search?
4. What is the maximum number of nodes expanded by breadth-first graph search?
5. Is h = |u - x| + |v - y| an admissible heuristic for a state at (u,v)? Explain.
6. How many nodes are expanded by A* graph search using h?
7. Does h remain admissible if some links are removed?
8. Does h remain admissible if some links are added between nonadjacent states?

### Part 2: Programming Assignment (60 points)

Using the code you wrote for assignment 1 and assignment 2, write a function that solves a given sliding bricks puzzle using A*. As a heuristic, use the Manhattan distance between the master brick and the goal. Notice that the search space is a graph, so you will have to keep track of all the states visited so far, and make sure your algorithm does not get stuck in loops.

When the solution is found, it should be printed to screen. Print the list of moves required to solve the state, and the final state of the puzzle, for example:

```
(2,left)
(4,down)
(3,right)
(2,up)
(2,up)
5,4,
1,2,2,1,1,
1,0,0,3,1,
1,0,0,4,1,
1,1,1,1,1,
```

Together with the source code, turn in (in a plain text file called 'output-part2.txt') the output that your program generates for the following levels: SBP-level0.txt, SBP-level1.txt, SBP-level2.txt, SBP-level3.txt, SBP-bricks-level1.txt, SBP-bricks-level2.txt, SBP-bricks-level3.txt, SBP-bricks-level4.txt, SBP-bricks-level5.txt, SBP-bricks-level6.txt, SBP-bricks-level7.txt. Also, report how many nodes are explored, how much time does the search take, and the length of the solution found

**Part 3: Extra Credit (10 points)**

Devise a better admissible heuristic. Notice that the performance of A* strongly depends on the heuristic you use. Try to design the best heuristic you can, and turn in an explanation of the heuristic in a readme text file (please recall that we do NOT accept Microsoft Word files, use plain text files).

**What to Submit**

All homework for this course must be submitted electronically using Bb Vista. Do not e-mail your assignment to a TA or Instructor! If you are having difficulty with your Bb Vista account, you are responsible for resolving these problems with a TA, an Instructor, or someone from IRT, before the assignment it due. It is suggested you complete your work early so that a TA can help you if you have difficulty with this process.

For this assignment, you must submit:

- A PDF document with your answers to the "Written problems to be turned in" (Do NOT submit a Microsoft Word, OpenOffice document, or any other format that is not a PDF, you will lose points if you do so!).
- Your C/C++/Java/Lisp,/Python/Javascript/... source code, written documentation for your program, and results of your testing.
- Strongly recommended. We strongly recommend using a compression utility so that you can compress your files into a single file (with a .zip extension) and just upload it, rather than go through the tedious and error-prone process of uploading each file separately.

**Academic Honesty**

You must compose all program and written material yourself, including answers to book questions. All material taken from outside sources must be appropriately cited. If you need assistance with this aspect of the assignment, see a consultant during consulting hours.