Chapter 3 Homework

Due date: Nov. 16, 2017

Program Exercises

1. Suppose that *maze*[*x*][*y*][z] is a 3-dimentional maze of the size $X \times Y \times Z$, where *x, y, z*∈{0, 1}. 1 denotes barrier and 0 denotes open path. Write a function, *path*, to search a path for a maze. The entry is at (0, 0, 0), while the exit is at (*X*-1, *Y*-1, *Z*-1)

   Requirements:

   a. An array *move*[*dir*].*vert* and an array *move*[*dir*].*horiz* can be used to indicate the direction of the next move. (***hint***: the size of each *move* array could be 26)

   b. Print out the path (all the locations along a path) if there is one. If there is no available path, print out the string "There is no available path."

2. Program 3.15 in the textbook has illustrated a function to convert from infix expression to postfix notation. Please rewrite the function so that it works with the following operations: &&, !!, <<, >>, <=, !=, <, >, and >=.

3. We wish to implement *n* stacks over a one-dimensional array of the size *m*. The space of the array is equally allocated to *n* stacks. If one of *n* stacks is full, write a function *stackFull* to hand this situation.

   Requirements:

   a. You can use *mealloc* only if the array is fully.

   b. If stack *i* is full, find the smallest *j* (*j*>*i*) such that there is an available space, and include this space to *i*.

   c. If there is not such *j* (*j*>*i*), find the largest *j* (*j*<*i*) such that there is an available space, and include this space to *i*.

   d. Complete the entire program, including initialize the array and *n* stacks, and allow users to push an item to the *i* stack. (Allow user to input *m* and *n*, and then allow user to keep pushing items to any stack)

   e. When the *i* stack is full and you find a space from *j*, print out the the string "The *i* stack is full, but we find a space from the *j* stack, and now the size of *j* is *CURRENT_SIZE_J,* and the size of *i* is *CURRENT_SIZE_I*". If the *i* stack is full and the array is also full, then print out the the string "The *i* stack is full and the array is also full, but we create a new space for *i*, and now the size of *i* is *CURRENT_SIZE_I* "