Judelle Clareese E. Gaza

CMSC 21 – 1 || Lec 6 - 7

1.

```
/* ==================== LINE REVISIONS ========================== */
/*1A*/ bool pathway[8] = {[0]true, [2]true};
/*1B*/ bool pathway[8] = {true, false, true};
```

2. My code has 2 functions: The Main Function and the NetworkNavigation. Both Functions are void functions, so they don't have to return a value. The NetworkNavigation function is a recursion and takes 3 arguments. First, I will discuss the Main Function. The flow of the Main Function goes as follows: **(1) Declaring and Assigning Variables => (2) Printing of the Adjacency Matrix => (3) User Input => (4) Network Navigation.**

========== MAIN FUNCTION ==========

(1) Declaring and Assigning Variables

First, I declared a macro (ARRAY = 8), for the 2d size of the array. I only used one because it is an 8x8 multidimensional array. Next, in the main function I have a starting point variable (startpoint) to store the user input, the 8x8 multidimensional array using the ARRAY variable (road_networks), and a character array to store the nodes/stations (point_stations).

(2) Printing of the Adjacency Matrix

Next I print out the Matrix using 2 for loops. One normal for loop to print out the first row: A B [C] [D] E F G H. Point C and D are special cases because these are charging stations so I have to create a condition for this. The Second For loop is a nested loop for printing out the point/node, along with its Boolean values. First, we print out the points from point_stations array, then the inner for loop prints out the bool values from the road_networks array, ex: A 1 1 0 0 0 1 0 0. All for loops iterate 8 times, this is why I only used one macro ARRAY.

(3) User Input

A simple while loop is used to get the User Input. This code block asks user for the starting point from A to H and checks if the user input is valid [0-7]. The loop breaks if valid, but if user input is invalid then the loop loops until the user gives a valid input.

(4) ========== NetworkNavigation ==========

Next, we call the recursive function, NetworkNavigation. This function takes 3 arguments: the current point, the adjacency matrix, and the points/nodes (startpoint, road_networks, point_stations). First, we evaluate the current point we have. If it is 2 or 3, then it means that the user is already at C or D, which are the charging stations. Else if the current point is at the direct path to C or D then the next 2 conditions are evaluated, we know it is in the direct path if the current point corresponds to 1 in either points C or D in the matrix. If it is in the direct path, then the next recursion will print out that the user has arrived to either charging station C or D. If the conditions above are not met, then it means that the user has to go through more nodes to arrive at charging station C or D. We do a for loop and a recursive call to change the value of the current point (with the current point incrementing but will remain less than 8) and print out the points it arrived to in each recursion. Eventually, the program will run until the user arrives at charging station C or D. Once it arrives to any of the charging stations the recursive call breaks.

FULL CODE

```c
// Judelle Gaza - CMSC 21 - Lec 6-7

#include <stdio.h>
#define ARRAY 8       // Macro for defining the size of the Array; the multidimensional array is 8 x 8 so one variable is enough

// ================================ RECURSIVE FUNCTION FOR NETWORK NAVIGATION ================================
/* This function is a recursion and has 4 conditions */
void NetworkNavigation(int currentpoint, int network[ARRAY][ARRAY], char stations[ARRAY]){
    printf("%c \n", stations[currentpoint]);                                    // Prints out the current point
    /* If the current point of the user is 2 or 3, then the user is at the Charging Station C or D */
    if (currentpoint == 2 || currentpoint == 3){
        printf("You have arrived at charging station %c\n",stations[currentpoint]);}

    /* If network[currentpoint][2] is == 1, then the user is at a direct path to Charging Station C */
    else if (network[currentpoint][2]){
        printf("Now at point ");
        NetworkNavigation(2, network, stations);}

    /* If network[currentpoint][3] is == 1, then the user is at a direct path to Charging Station C */
    else if (network[currentpoint][3]){
        printf("Now at point ");
        NetworkNavigation(3, network, stations);}

    /* If the user is not at C or D, and not in the direct path to C or D then the user has to navigate through other points,
       and start a recursive call until the user arrives at charging station C or D */
    else{
        for (int nextpoint = 0; nextpoint < ARRAY; nextpoint++) {
            if ((nextpoint != currentpoint) && (network[currentpoint][nextpoint])) {
                printf("Now at point ");
                NetworkNavigation(nextpoint, network, stations);
                break;}
        }
    }
}
// ================================================ MAIN FUNCTION ================================================
void main(){
    int startpoint;         // Variable for user input starting point
    // ------------------------- THE ARRAY OF ROAD NETWORKS ------------------------------
    int road_networks[ARRAY][ARRAY] = {
    //       A  B [C][D] E  F  G  H
            {1, 1, 0, 0, 0, 1, 0, 0},   //A
            {1, 1, 1, 0, 0, 0, 0, 0},   //B
            {0, 1, 1, 0, 1, 1, 0, 0},   //C
            {0, 0, 0, 1, 1, 0, 0, 0},   //D
            {0, 0, 0, 1, 1, 0, 0, 0},   //E
            {1, 0, 1, 0, 0, 1, 0, 0},   //F
            {1, 0, 0, 1, 0, 0, 1, 0},   //G
            {0, 0, 0, 0, 0, 1, 0, 1},}; //H

    // ------------------------- ARRAY OF THE STATIONS AND POINTS -------------------------
    char point_stations[ARRAY] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    /*======================== PRINTING OUT THE ADJACENCY MATRIX ============================*/
    /* Used 2 for Loops ~~ The first for loop is for printing out the first row which are the station points
       The second for loop is a nested loop for printing out the rest of the rows*/
    printf("     ");                                    //Add Spacing
    for (int j = 0; j < ARRAY; j++ ){                   //Iterate 8 times
        if (j == 2 || j == 3){                          //The special cases for letters C and D which are the charging stations- use "[]"
            printf("      [%c]", point_stations[j]);}
        else{                                           //Printing the points [A,B,E,F,G,H]
            printf("%8c ", point_stations[j]);}}
    printf("\n");
    /* Nested For loop that prints out the rest of the rows along with the road networks */
    for (int j = 0; j < ARRAY; j++ ){
        if (j == 2 || j == 3){                          //Prints out C and D with "[]"
            printf("[%c] ", point_stations[j]);}
        else{                                           //Printing the points [A,B,E,F,G,H]
            printf(" %-3c", point_stations[j]);}
        for (int k = 0; k< ARRAY; k++){                 //Prints out the road networks values that corresponds with the stations
            printf("%9d", road_networks[j][k]);}
        printf("\n");
    }
    /*============= While Loop For Multiple User Input Given That User Input Is Invalid ====================
        Asks user for the starting point, checks if the user input is valid [0-7], breaks if valid, loops if  */
    while(1) {
        printf("\n========== Which point are you located? ==========\n0 - A\n1 - B\n2 - C\n3 - D\n4 - E\n5 - F\n6 - G\n7 - H\nInput [0-7]:");
        scanf("%d", &startpoint);
        if (startpoint < 0 || startpoint > 7){
            printf("Input Invalid. Try Again.\n");}
        else{break;}
    }
    /*============= CALLING RECURSIVE FUNCTION TO NAVIGATE THROUGH POINTS AND TO ARRIVE AT A CHARGING STATION ============
        The function below takes the starting point of the user, and the road networks multidimensional array, as well as the point stations
        Since the function is recursive it terminates ones the conditions are met -> that is if the user arrives at a charging station*/
    printf("You are at point ");
    NetworkNavigation(startpoint,road_networks,point_stations);
}
```