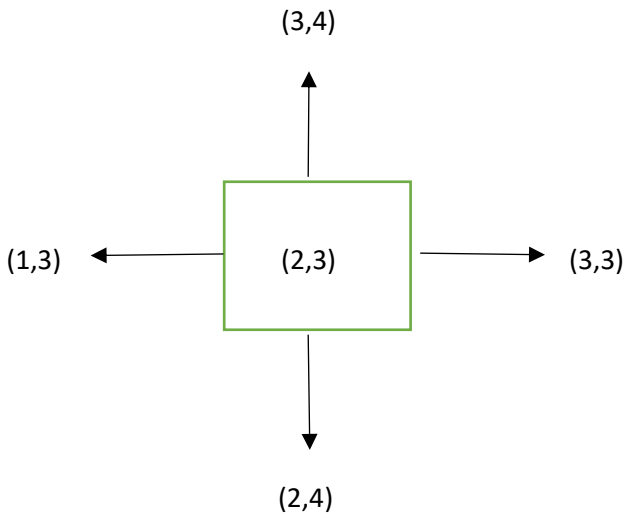
Foundations of Intelligent Systems | **Name : Pillai Rajkumar Lenin**
LAB-1 Report | **Section-01**
Prof : Jansen Orfan | **Course: CSCI-630**

1. Consideration of neighbor pixels:-

In this program the neighbor of a pixel is considered to be the 4 neighbors with either of them present in either in one of the two horizontal directions or one of the two vertical direction from current pixel.

Consider the below diagram showing the neighbor of pixel (2,3) :



The neighbor pixels for (2,3) are: (3,4) , (3,3) , (1,3) and (2,4)

A pixels neighbors are stored in dictionary named neighbordictionary in program with the current pixel as key and neighbor pixels as value.

2. Evaluation of cost between pixels:-

The cost function which is considered in this program is time which means minimum time to travel neighboring pixel of current pixel. The function used in the program is f(n) = g(n) + h(n).

Here the time g(n) is calculated using the formula:

G(n) = $\sqrt{((x2-x1)*10.29)^2+((y2-y1)*7.55)^2+((z2-z1))^2}$ / speed through that terrain

The above equation uses Euclidean distance to calculate distance between current and neighbor pixels

Here x_1 corresponds to the x co-ordinate of current pixel and x_2 corresponds to x coordinate of neighboring pixel and y_1 corresponds to the y co-ordinate of current pixel and y_2 corresponds to y coordinate of neighboring pixel

z_1 corresponds to the elevation of current pixel and z_2 corresponds to elevation of neighboring pixel

The numbers 10.24 and 7.55 are the pixel size in longitude(X) and latitude(Y) respectively.

The speed value associated with different terrains is given in below table which are assumed to be as follows:

Terrain type	***Speed (m/s)***
Open land	240
Rough meadow	150
Easy movement forest	60 (fall) and 120(other seasons)
Slow run forest	80
Walk forest	20
Impassible vegetation	1
Lake/Swamp/Marsh	1
Paved road	350
Footpath	20
Out of bounds	1
Ice	170
Mud	1

Here the time h(n) which is heuristic function is calculated using the formula:

H (n) = $\sqrt{((x2-x1)*10.29)^2+((y2-y1)*7.55)^2+((z2-z1))^2}$ / (Fastest speed among the available terrains)

Here x_1 corresponds to the x co-ordinate of current pixel and x_2 corresponds to x coordinate of goal pixel and y_1 corresponds to the y co-ordinate of current pixel and y_2 corresponds to y coordinate of goal pixel

z_1 corresponds to the elevation of current pixel and z_2 corresponds to elevation of goal pixel

Here the constants 10.29 and 7.55 have the same meaning as above. Fastest speed among the available terrains is 350 m/s which is paved road . The elevation factor $(z2-z1)^2$ in the above equations and type of terrain speed takes care of the amount of speed to travel in pixels which would be slower of uphill and higher for downhill.

The above heuristic function is admissible as it uses Euclidean distance between current pixel and goal pixel and divides it with the fastest speed in the map to estimate the time to reach goal node from current node and thus will provide a optimum solution to reach goal pixel from start pixel. It may expand the nodes which are not optimal but not all nodes as it constraints itself by using a Euclidean distance between pixels and then dividing be the fastest speed to get the time to travel to neighbor nodes.

3. Algorithm in different seasons:-

In different seasons the algorithm runs as follows :

Summer: In summer the given image terrain.png is considered to be the input image and then set of points are read from file white.txt or brown.txt or red.txt. Then astarsearch() function is used in program which uses heapq as priority queue to expand relevant nodes from start to the goal pixel using the f(n) formula described above.The optimal path is sent to the showImage function in the program which displays a new image with optimal path between set of given points.

Winter: In winter, first all the pixels that are next to water are found using the waterPixelsneighbors() function in program. This list contains all pixels that have at least one water pixel as neighbors and then from this list the pixels which are water pixels are removed by comparing their RGBvalue from RGBARRAY which stores the RGB value of image. Thus, we obtain all the pixels that are next to water. Now ,this land pixels which are next to water are passed to the BFS_Winter function which finds all pixels that are 7 pixels away from land towards water which are safe to walk and this list is sent to called function to change corresponding pixel to ice and in BFS function the land pixels are initialized in the queue and expanded till a depth of 7 then a new image is generated using draw function of python image library and astarsearchfunction() is used to find the optimal path.

Fall: The algorithm works in same manner as in summer season with just one change which is the speed through walk forest is reduced from (120 m/s which is for all other seasons) to 60m/s in the typeOfterrain() function in the program.

Spring: In spring the image terrain.png is taken as input image and then all pixels in the image that are water pixels are found and stored in a list. Then this list is passed to the BFS_mud () function which starts at a water pixels from the list and then find the neighbors of that pixel along with calculating the elevation till it reaches a non-water pixel and if the difference in elevatoion is greather than 1m then the pixel is considered to be water pixel and added to the list of a mudpixels.This list mudpixels is passed to the showImage() image which converts the pixels in the mudpixels list to color brown and then this Image is used to perform astarsearch and find the optimal path between given set of points.In the program, it is considered that going through the mud terrain is not possible.

For Human readable output,the function showImage() takes the Imagefilename and path list as argument and draws a red point for each pixel which is in list using draw function of python Image library.

4. Different data structures used in program

The different datastructures used in program are:

Gndictionary:-Stores the cost so far to reach the pixel from start pixel

Hndictionary:- Stores the estimated cost to reach goal pixel from current pixel

Fndictionary:- Stores the g(n)+h(n) value for each pixel.

Neighbourdictionary:- Stores maximum of 4 possible neighbors for current pixel.

hprimen: The Array used to store the h(n) value for each pixel in image.

ElevationArray: The array which contains elevation corresponding to pixels in image

RGBArray: Array containing RGB value of image

The functions used in program are:-

def summer(): Finds the optimal path in summer.

def winter(): Finds the optimal path in winter.

def spring(): Finds the optimal path in spring.

def fall():Finds the optimal path in fall.

def calculatedistance(): Calculates the g(n) or f(n) value based on the operationtype passed to the function.

def elevation(): Reads the elevation value from file and stores in the elevationArray.

def waterPixelsneighbors(): Finds all landedges near to water.

The Total path length is finally displayed on the terminal