

Task 2 - Image Processing and Path Planning

Please find the following in the Experiment folder:

- **task2sets:** this folder comprises of 7 test images containing randomly generated 14x14 grids having “1”s and “0”s for checking the effectiveness of your code
- **digits:** this folder comprises of image templates for digit “1” and “0” as a resource
- **outputs:** this folder will store the output images after you run your code
- **imgLib.py** and **task2-main.py:** these are python files for you to modify
NOTE: Please open and read the comments in these files carefully and follow the instructions as given in the comments. Do NOT edit the code wherever mentioned in the comments.
- **grid_map_solution, route_length_solution** and **route_path_solution:** these files are provided for you to verify your solution

Given:

- A 14x14 image comprising of randomly generated 14x14 grids having “1”s and “0”s as shown in Figure 1
- The axes are shown in Figure 1 for reference. Please note that:
 - top-left cell in the grid corresponds to the coordinate (1,1)
 - top-right cell in the grid corresponds to coordinate (14,1)
 - bottom-left cell in the grid corresponds to coordinate (1,14)
 - bottom-right cell in the grid corresponds to coordinate (14,14)
- Each grid in the image has a **Start Row{(1,14) to (14,14)}** and **Destination Row{(1,1) to (14,1)}** as shown in Figure 1
- Each “1” in the Start Row is a valid Starting Point and each “1” in the Destination Row is a valid Ending point
- The “1”s in the grid correspond to **valid cells** and can be visited
- The “0”s in the grid correspond to **invalid cells** and cannot be visited
- Valid moves within the grid are:
 - Up
 - Down
 - Left
 - Right
 - Diagonal
- **Valid Route:** A valid route is any valid path in the grid that starts from a Starting Point and ends at the Ending Point
- A valid cell can be visited only once for each Valid Route.

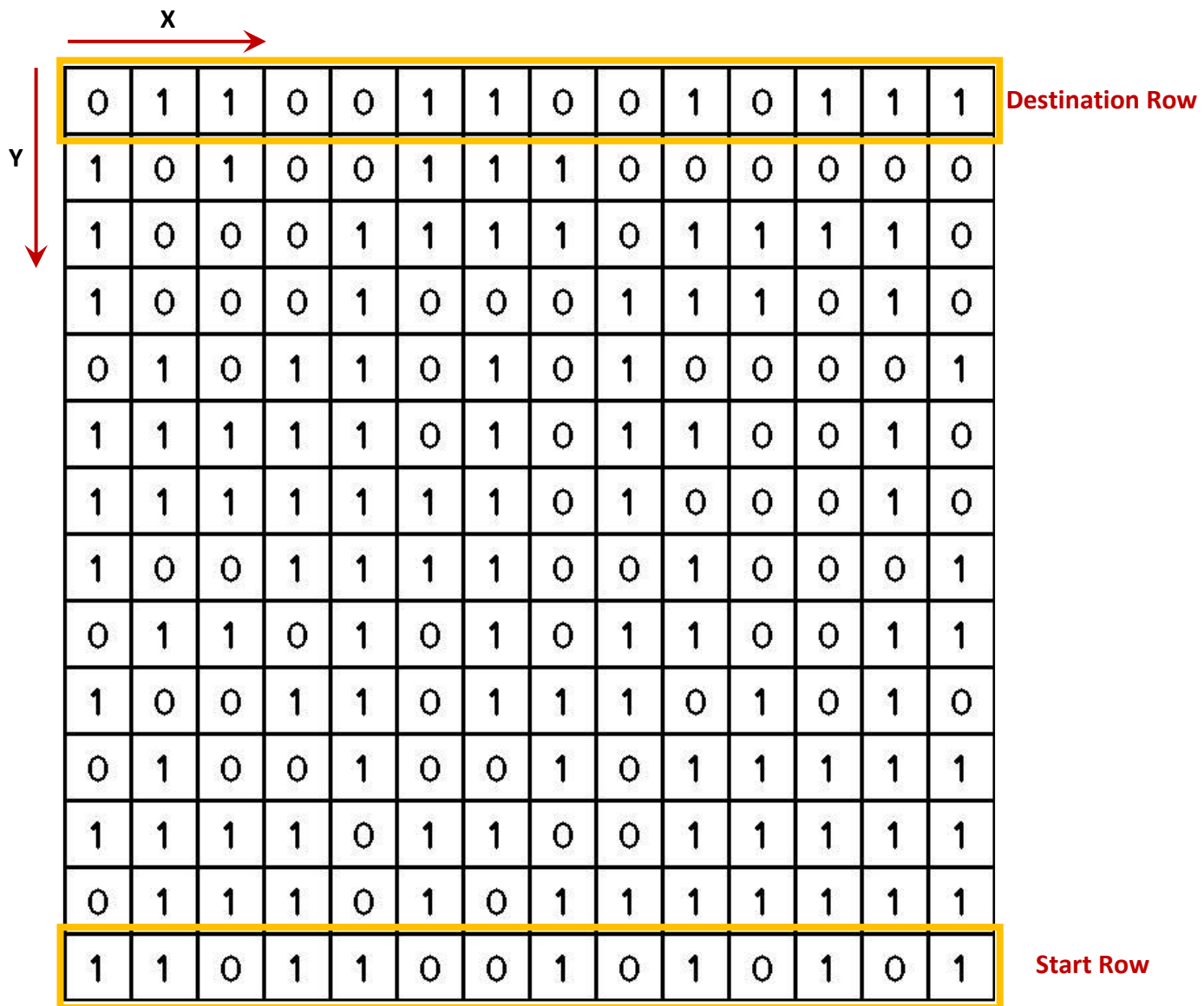


Figure 1: Image

Problem Statement:

To read the test images in the **task2sets** folder and find the shortest Valid Route within the grid by modifying python files **task2-main.py** and **imgLib.py**

Output format:

1. If a shortest Valid Route is found, the output of your code should:
 - Print the whole grid map of 1s and 0s as an array
 - Print the **route length**
 - Print the **route path**
 - Show the shortest Valid Route (in Blue colour) in the output image and store it in the Output folder.

Consider the test image as shown in the Figure 2. The shortest Valid Route marked in blue and the sample output are shown in Figure 3

0	1	1	0	0	1	1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	0	1	1	1	0	1	0
0	1	0	1	1	0	1	0	1	0	0	0	0	1
1	1	1	1	1	0	1	0	1	1	0	0	1	0
1	1	1	1	1	1	1	0	1	0	0	0	1	0
1	0	0	1	1	1	1	0	0	1	0	0	0	1
0	1	1	0	1	0	1	0	1	1	0	0	1	1
1	0	0	1	1	0	1	1	1	0	1	0	1	0
0	1	0	0	1	0	0	1	0	1	1	1	1	1
1	1	1	1	0	1	1	0	0	1	1	1	1	1
0	1	1	1	0	1	0	1	1	1	1	1	1	1
1	1	0	1	1	0	0	1	0	1	0	1	0	1

Figure 2: Test Image 1

Output format for Test Image 1 in Figure 2:

0	1	1	0	0	1	1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	0	1	1	1	0	1	0
0	1	0	1	1	0	1	0	1	0	0	0	0	1
1	1	1	1	1	0	1	0	1	1	0	0	1	0
1	1	1	1	1	1	1	0	1	0	0	0	1	0
1	0	0	1	1	1	1	0	0	1	0	0	0	1
0	1	1	0	1	0	1	0	1	1	0	0	1	1
1	0	0	1	1	0	1	1	1	0	1	0	1	0
0	1	0	0	1	0	0	1	0	1	1	1	1	1
1	1	1	1	0	1	1	0	0	1	1	1	1	1
0	1	1	1	0	1	0	1	1	1	1	1	1	1
1	1	0	1	1	0	0	1	0	1	0	1	0	1

Figure 3: Test Image 1

[[0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1], [1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0], [1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0], [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0], [1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0], [1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0], [0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1], [1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0], [0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1], [1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1], [0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1], [1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1]]

route length 13

route path [(1,14), (2,13), (1,12), (2,11), (1,10), (2,9), (1,8), (1,7), (1,6), (2,5), (1,4), (1,3), (1,2), (2,1)]

2. If NO Valid Route is found, the output of your code should:
 - Print the whole grid map of 1s and 0s as an array.
 - Print the string **"No Path Found"**

For the test image as shown in the Figure 4, the sample output is as shown.

0	1	1	0	0	1	1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	0	1	1	1	0	1	0
0	1	0	1	1	0	1	0	1	0	0	0	0	1
1	1	1	1	1	0	1	0	1	1	0	0	1	0
1	1	1	1	1	1	1	0	1	0	0	0	1	0
1	0	0	1	1	1	1	0	0	1	0	0	0	1
0	1	1	0	1	0	1	0	1	1	0	0	1	1
1	0	0	1	1	0	1	1	1	0	1	0	1	0
0	1	0	0	1	0	0	1	0	1	1	1	1	1
1	1	1	1	0	1	1	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	1	0	1	0	1	0	1

Figure 4: Test Image 2

Output format for Test Image 2 in Figure 4:

```
[[0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1], [1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0], [1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0], [1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0], [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1], [1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0], [1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0], [1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0], [0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1], [1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0], [0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1], [1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0], [1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1]]
```

No Path Found

3. If there are **multiple Valid Routes** and
 - a. a shortest Valid Route is found out of these multiple Valid Routes, return this shortest Valid Route according to Output format as explained in point 1 above.
 - b. two or more Valid Routes have the same route length, return any one of these shortest Valid Routes according to Output format as explained in point 1 above.