1. **(20 points)** Please write a C/C++ program to find a min-cost path connecting two points on a grid with one blockage and one free-island. For example, see the 7x7 grid below, with row and column indices from 1 to 7, starting from bottom and left, respectively. You need to find a path from S(6, 3), which is row 6 and column 3, to T(3,6), which is row 3 and column 6. The routing cost from grid cell to its neighbor cell is 1. There is one blockage indicated by the orange rectangle, where routing wires cannot enter. It is specified by (4,3), its lower left grid cell and (5,7), its upper right cell. There is one free island, where routing inside has 0 cost, like the green rectangle. It is specified by (1,2) and (2,6). A wire from an empty cell to a neighbor green cell has cost 1.

Use such command to compile my code. (Only one cpp file, hsu1725.cpp)
g++ -std=c++11 **Path**/hsu1725.cpp -o **Path**/hsu1725.out
For example, my Path: /home/grads/w/**NetID**/ece687

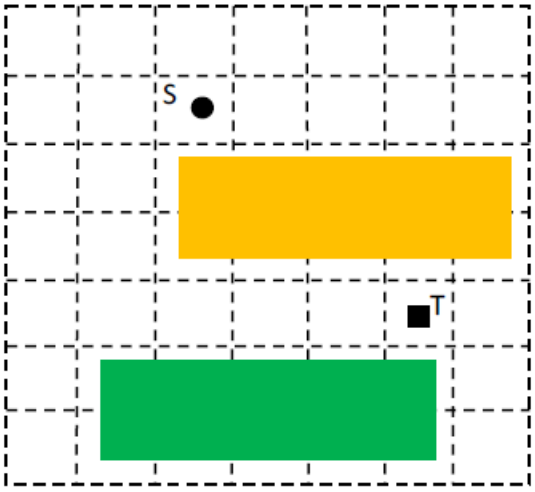
How to test, put hsu1725.out and test file in the same folder.
The filename input by command line.
./hsu1725.out **Your_Test_FileName**
For example, my Test_FileName: test_input1

[Following command I used for the following Figure]
g++ -std=c++11 /home/grads/w/**NetID**/ece687/hsu1725.cpp
-o /home/grads/w/**NetID**/ece687/hsu1725.out
./hsu1725.out test_input1
./hsu1725.out test_input2

Below is the simulation for my program on ECE server hera:

```
[redacted]@hera3 ~> (02:05:58 11/04/17)
:: cd ece687/

[redacted]@hera3 ~/ece687> (02:06:02 11/04/17)
:: ls
hsu1725.cpp  hsu1725.out  hw10  hw3  hw4  test_input1  test_input2  test_input3

[redacted]@hera3 ~/ece687> (02:06:03 11/04/17)
:: g++ -std=c++11 /home/grads/w/[redacted]/ece687/hsu1725.cpp -o /home/grads/w/
[redacted]/ece687/hsu1725.out

[redacted]@hera3 ~/ece687> (02:06:13 11/04/17)
:: ./hsu1725.out test_input1
Path cost from (6,3) to (3,6): 6
PATH: (6,2) (5,2) (4,2) (3,2) (2,2) (2,3) (2,4) (2,5) (2,6) (3,6)

[redacted]@hera3 ~/ece687> (02:06:25 11/04/17)
:: ./hsu1725.out test_input2
Path cost from (6,3) to (3,6): 8
PATH: (6,2) (6,1) (5,1) (4,1) (3,1) (2,1) (2,2) (2,3) (2,4) (2,5) (2,6) (3,6)

[redacted]@hera3 ~/ece687> (02:06:31 11/04/17)
:: ./hsu1725.out test_input3
There is no path from source to target


[redacted]@hera3 ~/ece687> (02:06:35 11/04/17)
::
```

[Example] Test Pattern (Yellow part is blockage, and green part is free island):
test_input1



Output:

```
Path cost from (6,3) to (3,6): 6
PATH: (6,2) (5,2) (4,2) (3,2) (2,2) (2,3) (2,4) (2,5) (2,6) (3,6)
```

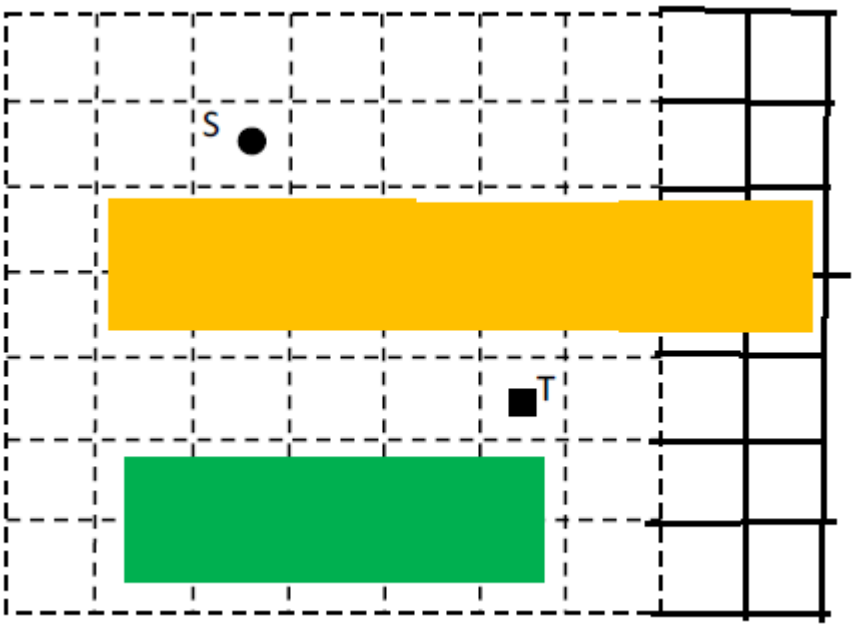
Output:

- Total path cost of your solution.
- The path in term of grid cells on the path. For the example above, the output path format is like:
  (6,2) (5,2) (4,2)(3,2)(2,2)(2,3)(2,4)(2,5)(2,6)(3,6)

For the example above, the input file is like:

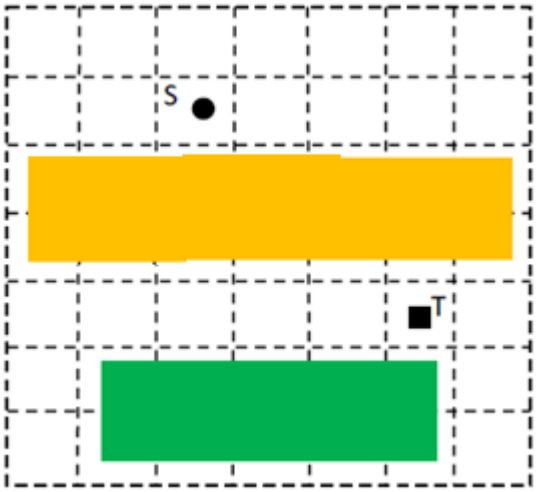
7 7
6 3
3 6
4 3 5 7
1 2 2 6

test_input2:




```
Path cost from (6,3) to (3,6): 8
PATH: (6,2) (6,1) (5,1) (4,1) (3,1) (2,1) (2,2) (2,3) (2,4) (2,5) (2,6) (3,6)
```

test_input3:




```
There is no path from source to target
```