# PROJECT ASSIGNMENT 4

**Issue Date : 10.12.2024 - Tuesday**
**Recitation Date : 10.12.2024 - Tuesday (20:00) (held on Zoom)**
**Due Date : 22.12.2024 - Sunday (23:00)**
**Advisor :** R.A. Görkem AKYILDIZ
**Programming Language :** Python 3.9.18

## 1 Introduction

Humankind is a social-being indeed, even though this fact, most of the humankind are also tend to be selfish which makes social life harder. According to this sad fact, lawmakers adopt "Your right ends where the right of others begin." as a principle, so that people get afraid of being selfish not only because of conscience but also because of fear of being judged but as you cannot judge someone without any evidence, humankind still can be selfish if they have no conscience and there is no evidence of that selfishness. The sinkholes at the Konya are one of the dramatic result of this selfishness. Sinkholes mostly occur due to unconscious water consumption. Konya has an arid climate which means there supposed to grow agricultural products with low water consumption but people find that they can grow agricultural products with high water consumption with help of groundwater which sounded good for a while but nature has a balance and we must protect it, the balance that people ruined in this story is, they used groundwater more than it get feed which resulted with an empty space at the underground. Gravity did its job and created such sinkholes in Konya due to this dramatic scenario and a small part of the result can be seen at the image above.

Aim of this project is creating a program that calculates shortest possible route from one end of the field with sinkholes to another end.

## 2 Problem Definition

The problem here is there is a person at the one side of the field who tries to reach to the opposite side of the field, it is not important that which entrance s/he uses and which exit s/he reaches, the important thing is, s/he must start from left end and and end at the right end. S/he cannot pass over the sinkholes, s/he can only move in horizontal and vertical axis. Each move has a cost and there are three types of costs: $Cost_1$ is stepping over a part of the field that has no sinkhole one hop away. $Cost_2$ is stepping over a part of the field that has sinkhole in diagonal axis (one hop away). $Cost_3$ is stepping over a part of the field that has sinkhole in horizontal or vertical axis (one hop away). For the sake of converting continuous world into discrete one, the field can be thought as cells and costs can be though as $Cost_1$ there is no sinkhole in any neighbors cells (including the diagonal ones), $Cost_2$ there is no sinkhole in horizontal or vertical neighbors but there is at least one sinkhole in diagonal neighbors. $Cost_3$ is there is at least one sinkhole in horizontal or vertical neighbors.

## 3 Definition of Input

First line of the input contains costs in alphabetical order ($Cost_1$, $Cost_2$, and $Cost_3$), they are separated with space character. Following lines represent bird's-eye view of the field itself, zeros represent sinkholes where ones represent ground. Each line represents each row of the field, each cell is separated with space character between each other.

## 4 Definition of Output

If there is no possible route, output must be "There is no possible route!".

If there is at least one possible route, first line of the output must be "Cost of the route: <COST_OF_THE_ROUTE>" where <COST_OF_THE_ROUTE> is cost of the route that is found. The following lines must represent bird's-eye view of the field itself as in the input file but route must be marked with the character "X".

## 5 Rules Must to Obey

In this project, there are some rules that must be obeyed, otherwise your grade may be as low as zero even if your solution is correct:

- The solution must contain recursion as necessary.
- While trying the routes the uppermost cell must be selected at the left side of the field and then lower ones must be tried.
- When having a new step, firstly right, then upper, then lower, and at the last left neighbor must be tried.
- The shortest route must be given as output, if there are more then one shortest routes, then the first one must be provided, the first one will be selected according to last two rules above.

## 6 Restrictions

- Your code must be able to execute on our department's developer server (dev.cs.hacettepe.edu.tr).
- You must obey given submit hierarchy and get score (1 point) from the submit system.
- **You must benefit from recursion.**
- Your code must be clean, do not forget that main function is just a driver function that means it is just for making your code fragments run, not for using them as a main container, create functions in necessary situations but use them as required.
- You must use comments for this project and you must give brief information about the challenging parts of your code. Do not over comment as it is against clean code approach. Design your comments so that they make your code fully understandable and not excessive for others. You can check guides of Python namely PEP-8 and PEP-257 for further information.
- You can benefit from Internet sources for inspiration but do not use any code that does not belong to you.
- You can discuss high-level (design) problems with your friends but do not share any code or implementation with anybody.
- Do not miss the submission deadline.
- Source code readability is a great of importance. Thus, write READABLE SOURCE CODE, comments, and clear MAIN function. This expectation will be graded as "clean code".
- Use UNDERSTANDABLE names to your variables, classes, and functions regardless of the length. The names of functions, attributes and classes should obey Python naming convention. This expectation will be graded as "coding standards".
- You can ask your questions through course's Piazza group, and you are supposed to be aware of everything discussed in the Piazza group. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- Submit system for this homework will be opened a few days before deadline, so please be patient.

## 7 Execution and Test

Your code must be executed under **Python 3.9.18** at **dev.cs.hacettepe.edu.tr**. If your code does not run at department's developer server during the testing stage, then you will be graded as 0 for the code part even if it works on your own machine.

Sample run command is as follows:

- python3 route_finder.py input.txt output.txt

## 8 Grading

| Task | Point |
|---|---|
| **Correct Output** | 80** |
| **Clean Code & Comment** | 20* |
| **Total** | 100 |

*** The score of the clean code & comment part will be multiplied by your overall score (excluding clean code & comment part) and divided by the maximum score that can be taken from these parts. Say that you got 60 from all parts excluding clean code & comment part and 10 from clean code & comment part, your score for clean code & comment part is going to be 10*(60/80) which is 7.5 and your overall score will be 60+7.5=67.5.**

**** It does not mean that you can get full credit if your output exactly matches, in addition to having correct output, you must obey all the given rules to get full credit, otherwise you may get a grade which is as low as zero.**

**Note that you must score one at the submit system, otherwise 20% of your grade will be deducted, moreover, usage of global variables are forbidden and you must implement a main function as advised otherwise 20% of your grade will be deducted! There may also be other point deductions if you do not obey the given rules, such as if you do not use recursion as necessary.**

**Also note that you must give the desired outputs to get full credit from the parts, otherwise you may get a grade which is as low as zero for the parts that is not giving the desired output even if your implementation is correct!**

**Last but not least, if your code just prints out "There is no possible route!" you will not gain any points for the cases with no possible route, your code must generate outputs for the inputs with possible routes to get point from the inputs with no possible routes.**

## 9 Submit Format

File hierarchy must be zipped before submitted (Not .rar, only not compressed as .zip files because the system just supports .zip files).

```
- b<StudentID>.zip
    - route_finder.py
```

## 10 Late Policy

You have two days for late submission. You will lose 10 points from maximum evaluation score for each day (your submitted study will be evaluated over 90 and 80 for each late submission day). You must submit your solution in at the most two days later than submission date, otherwise it will not be evaluated. Please do not e-mail to me even if you miss the deadline for a few seconds due to your own fault as it would be unfair for your friends, e-mail submissions will not be considered if you do not have a valid issue.