[UNI LOGO]

Course: [COURSE NAME]

[Assigment Name]

Lecturer: [LECURER NAME]

Name: [YOUR NAME]

Contents

1.0 – Background Study

This project involves in creating a fluid simulation in a 2d matric. The 2d matric contains random number from 0 to 9. The program can simulate flow of water by going through the matric according to the condition asked.

[TAMBAH LAGI KALO ADE]

2.0 - Code Explanation

```
sz=int(input("Size: "))
if sz<8 or sz>20:
    exit()
ran=[[]]
path,sp=" ",1
ran=[[random.randint(0,9) for i in range(sz)] for j in range(sz)]
```

This part of the code is same for the three levels

This part takes the size input from user into the “sz” and checks if it’s in between 8 and 20. If not it will exit the program. But if it’s true, that it will create the ran matric with random number from 0 to 9. The “path” variable holes the string that represent the path and the “sp” variable holds the distance used for printing the spacing.

2.1 – flow.py

```
def flow(i,j,flw,sm):
    sm+=flw[i][j]
    flw[i][j]=path
    if i==sz-1 or j==sz-1:
        for i in range(len(flw)):
            for j in flw[i]:
                print(str(j)+sp*" ",end='')
            print()
        print("Sum: "+str(sm)+"\n")
        return
    if flw[i+1][j]>flw[i][j+1]:
        flow(i,j+1,flw,sm)
    elif flw[i+1][j]<flw[i][j+1]:
        flow(i+1,j,flw,sm)
    else:
        flow(i,j+1,copy.deepcopy(flw),copy.deepcopy(sm))
        flow(i+1,j,copy.deepcopy(flw),copy.deepcopy(sm))
for i in ran:
    print(i)
print()
flow(0,0,ran,0)
```

Part of flow.py code

It starts from 0,0 in the matric and it checks if the right value from the current position or the bottom value form the current position is smaller. It will then go to the position of the smaller value. If both of the value are the same then it will deepcopy the matric and do the same for the both path. The whole

process uses a algorithm called “recursive”. It will continue this process until it reaches the right most (j=s-1) value of the matric or the botton most (i=s-1) value of the matric. Then it will print the output.

2.2 – flow2.py

```
if j==0:
    if flw[i+1][j]>flw[i][j+1]:
        flow2(i,j+1,flw,sm)
    elif flw[i+1][j]<flw[i][j+1]:
        flow2(i+1,j,flw,sm)
    else:
        flow2(i,j+1,copy.deepcopy(flw),copy.deepcopy(sm))
        flow2(i+1,j,copy.deepcopy(flw),copy.deepcopy(sm))
elif j==sz-1:
    if flw[i+1][j]>flw[i][j-1]:
        flow2(i,j-1,flw,sm)
    elif flw[i+1][j]<flw[i][j-1]:
        flow2(i+1,j,flw,sm)
    else:
        flow2(i,j-1,copy.deepcopy(flw),copy.deepcopy(sm))
        flow2(i+1,j,copy.deepcopy(flw),copy.deepcopy(sm))
else:
    if flw[i][j-1]<flw[i+1][j] and flw[i][j-1]<flw[i][j+1]:
        flow2(i,j-1,flw,sm)
    elif flw[i+1][j]<flw[i][j-1] and flw[i+1][j]<flw[i][j+1]:
        flow2(i+1,j,flw,sm)
    elif flw[i][j+1]<flw[i][j-1] and flw[i][j+1]<flw[i+1][j]:
        flow2(i,j+1,flw,sm)
    else:
        if flw[i][j-1]==flw[i+1][j] and flw[i][j-1]==flw[i][j+1]:
            flow2(i,j-1,copy.deepcopy(flw),copy.deepcopy(sm))
            flow2(i+1,j,copy.deepcopy(flw),copy.deepcopy(sm))
            flow2(i,j+1,copy.deepcopy(flw),copy.deepcopy(sm))
        elif flw[i][j-1]==flw[i+1][j]:
            flow2(i,j-1,copy.deepcopy(flw),copy.deepcopy(sm))
            flow2(i+1,j,copy.deepcopy(flw),copy.deepcopy(sm))
        elif flw[i][j-1]==flw[i][j+1]:
            flow2(i,j-1,copy.deepcopy(flw),copy.deepcopy(sm))
            flow2(i,j+1,copy.deepcopy(flw),copy.deepcopy(sm))
        elif flw[i+1][j]==flw[i][j+1]:
            flow2(i+1,j,copy.deepcopy(flw),copy.deepcopy(sm))
            flow2(i,j+1,copy.deepcopy(flw),copy.deepcopy(sm))
```

Part of flow2.py code

It starts from 0,0 in the matric and it checks if the right value, left value, and the bottom value from the current position is smaller. It will then go to the position of the smaller value. If the two or three of the smaller value are the same then it will deepcopy the matric and do the same for those path. The whole process uses a algorithm called “recursive”. It will continue this process until it reaches the botton most (i=s-1) value of the matric. Then it will print the output.

2.3 – flow3.py

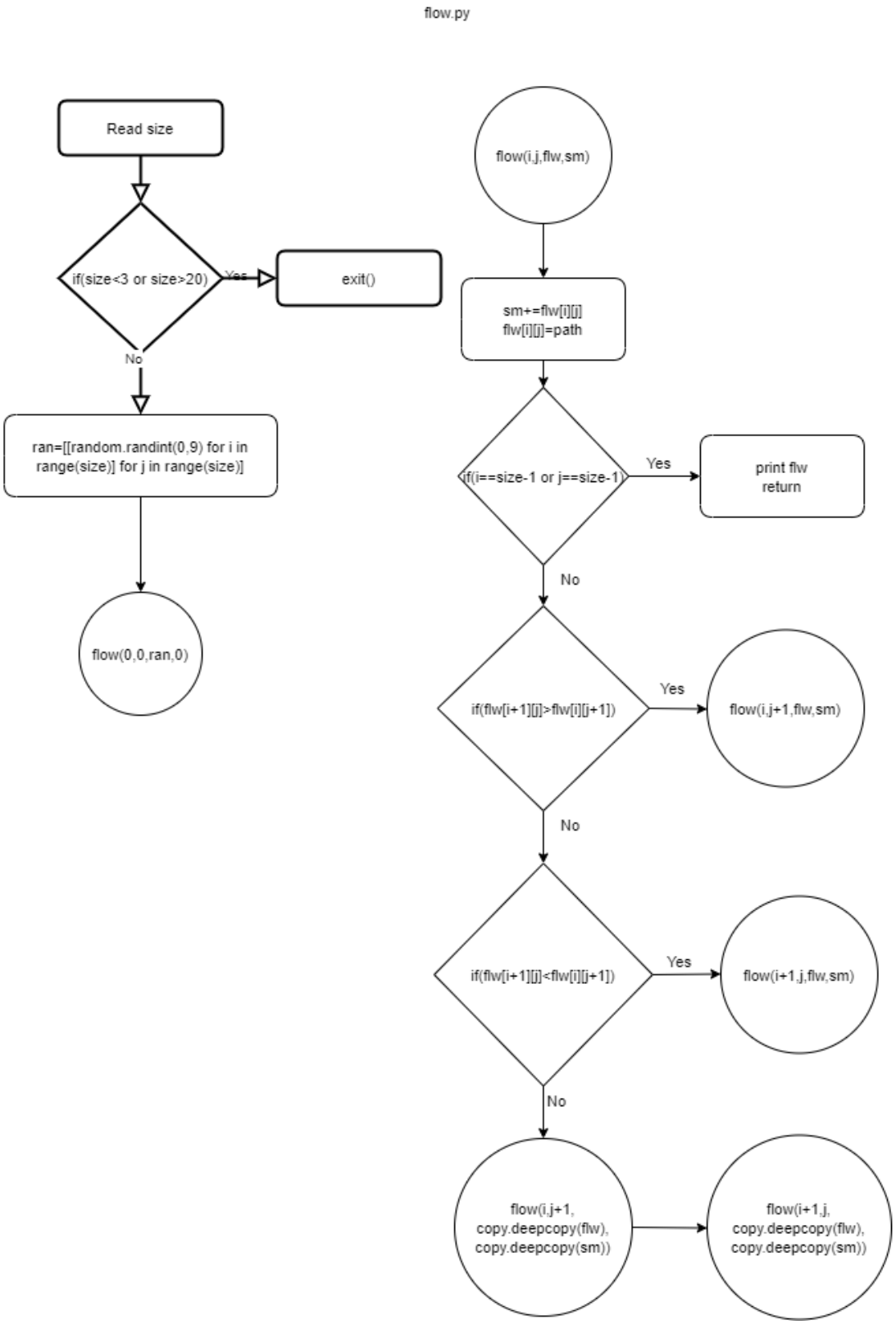
```
sz=int(input("Size: "))
if sz<8 or sz>20:
    exit()
print("Enter Starting Point, X and Y ")
x=int(input("X: "))-1
y=int(input("Y: "))-1
path,sps=" ",1
ran=[[random.randint(0,9) for i in range(sz)] for j in range(sz)]
```

Part of flow3.py code

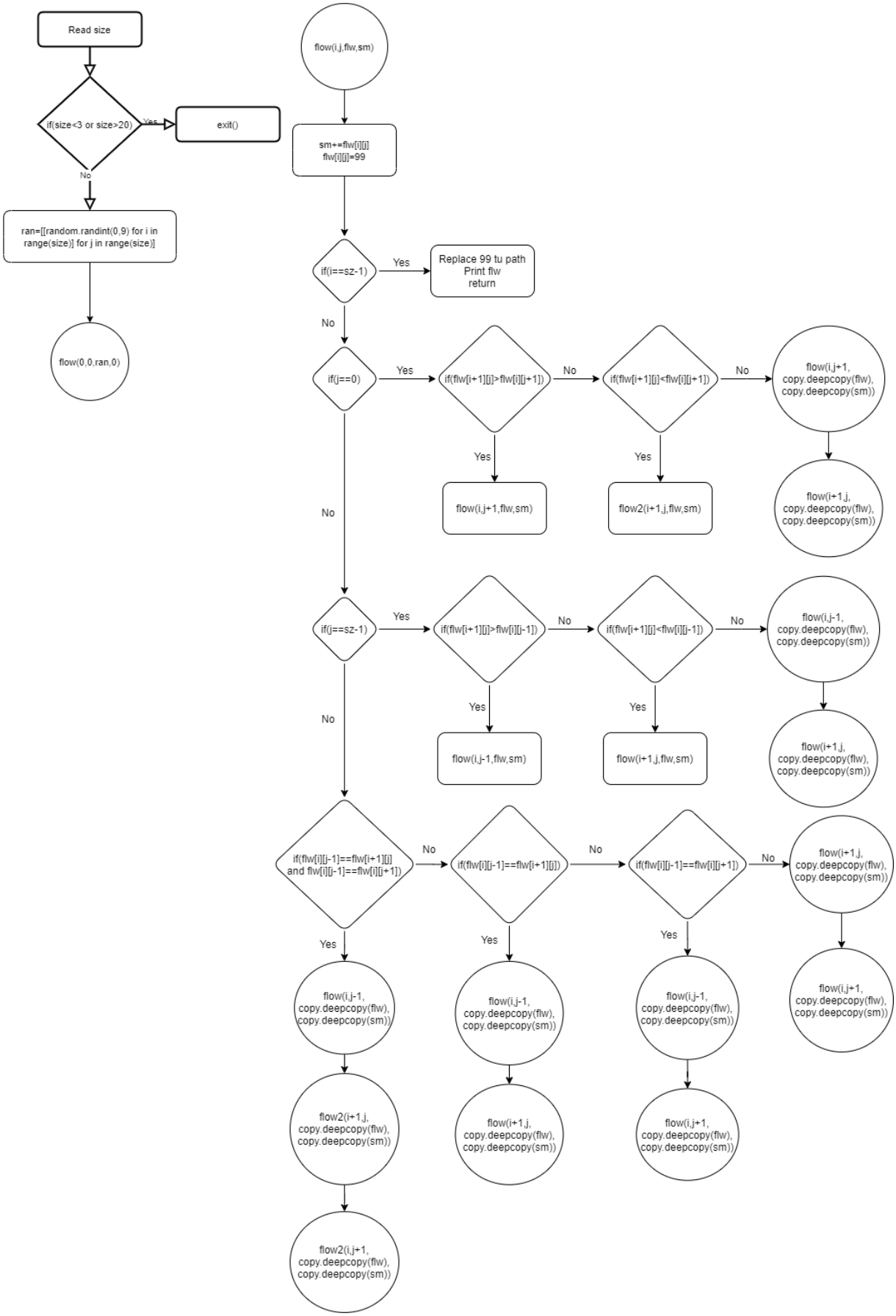
It's the same as the flow2.py file above but this one asks the user where they want to start the flow. The "x" and "y" variable holds the value of the position the user want to start the flow. User can enter from 1 to s for both x and y.

3.0 – Flow Charts

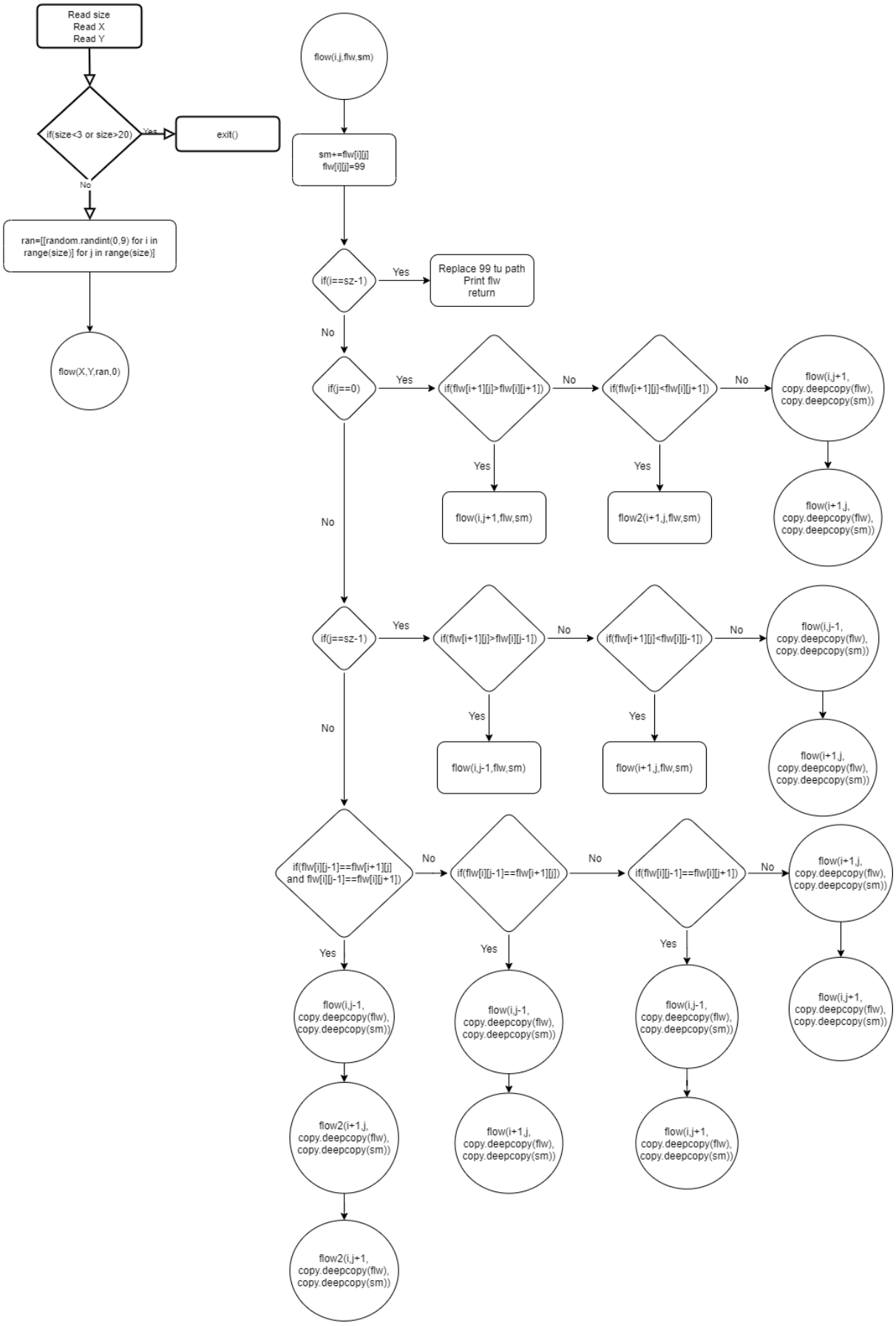
3.1 – flow.py



3.2 – flow2.py


Read size
if(size<3 or size>20)
Yes
exit()
No
ran=[[random.randint(0,9) for i in range(size)] for j in range(size)]
flow(0,0,ran,0)
flow(i,j,flw,sm)
sm+=flw[i][j]
flw[i][j]=99
if(i==sz-1)
Yes
Replace 99 tu path
Print flw
return
No
if(j==0)
Yes
if(flw[i+1][j]>flw[i][j+1])
No
if(flw[i+1][j]<flw[i][j+1])
No
flow(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j+1,flw,sm)
Yes
flow2(i+1,j,flw,sm)
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
No
if(j==sz-1)
Yes
if(flw[i+1][j]>flw[i][j-1])
No
if(flw[i+1][j]<flw[i][j-1])
No
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,flw,sm)
Yes
flow(i+1,j,flw,sm)
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
No
if(flw[i][j-1]==flw[i+1][j]
and flw[i][j-1]==flw[i][j+1])
No
if(flw[i][j-1]==flw[i+1][j])
No
if(flw[i][j-1]==flw[i][j+1])
No
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow2(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow2(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))

3.3 – flow3.py


Read size
Read X
Read Y
if(size<3 or size>20)
Yes
exit()
No
ran=[[random.randint(0,9) for i in range(size)] for j in range(size)]
flow(X,Y,ran,0)
flow(i,j,flw,sm)
sm+=flw[i][j]
flw[i][j]=99
if(i==sz-1)
Yes
Replace 99 tu path
Print flw
return
No
if(j==0)
Yes
if(flw[i+1][j]>flw[i][j+1])
No
if(flw[i+1][j]<flw[i][j+1])
No
flow(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j+1,flw,sm)
Yes
flow2(i+1,j,flw,sm)
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
No
if(j==sz-1)
Yes
if(flw[i+1][j]>flw[i][j-1])
No
if(flw[i+1][j]<flw[i][j-1])
No
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,flw,sm)
Yes
flow(i+1,j,flw,sm)
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
No
if(flw[i][j-1]==flw[i+1][j]
and flw[i][j-1]==flw[i][j+1])
No
if(flw[i][j-1]==flw[i+1][j])
No
if(flw[i][j-1]==flw[i][j+1])
No
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
Yes
flow(i,j-1,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow2(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow(i+1,j,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))
flow2(i,j+1,
copy.deepcopy(flw),
copy.deepcopy(sm))

4.0 – Discussion

Some part of the program has “deepcopy” and what this does is that it copy’s the data of a variable and uses the copied one. The deepcopy is used when the program finds a path that has two or three same value. This is used to prevent the program from overwriting the paths before. This ensures the path to continue from the last “deepcopied” part and give us all the possible outputs.

[TAMBAH LAGI IF TAK OK]

5.0 – Conclusion

This project is able to simulate all the possible path of fluid flow in a 2d random matric. It uses a recursive algorithm and deep copy through out the code. This project is fully created by [NAMA KAU].

[TAMBAH LAGI TAK OK}