DEAD END DETECTORS

21B01A1263 Jujjavarapu.Radha Rashmitha
21B01A12D6 Pasumarthi.Praneetha
21B01A54D2 Yadam Mohana Sai Siri Chandana
21B01A0201 Adabala Sri Ramya
21B01A1264 Juvva Sripranya

March 25,2023

Introduction

Problem Statement

while locations and streets information are provided, placing dead end signs at dead end streets and removing the redundant dead end signs

Approach

- The method used in this code is to create a matrix using locations and initializing the matrix with zeros.
- Zeroes are replaced with ones where ever street is present.
- We then traverse the matrix to check each street if it is leading to dead end and append to a list
- Redundant dead ends are then removed and output is displayed.

Packages

- No packages are used

Challenges

- Our biggest challenge was formatting the code.
- Breaking the approach into different parts such that each function does its designated task required more work
- TEAM WORK and communication is most challenging part

Statistics

- Number of line of code:86
- Functions:
  - input_command_line()
  - connections()
  - detect_dead_ends()
  - is_dead_end()
  - remove_unwanted_dead_ends()

Demo of the project:

- Dead end detector project takes input about the locations and streets through command line. input_command_line() function takes input into list of lists and calls respective functions for further tasks.
- List of streets is sent to connections() function which establishes connections between different locations by 0 or 1 in matrix at respective positions
- connections_matrix is traversed in detect_dead_ends_() functions . If it finds 1 in any position that street is sent to is_dead_end() function to check whether it leads to dead end. If that function returns true,that street is added to dead_ends list.
- is_dead_end function checks whether it leads to dead end through continuous recursions.
- Finally redundant dead ends are removed from list of dead ends through remove_unwanted_dead_ends() function and display the output.

Screenshots:

WISE_PROJECT (1).py

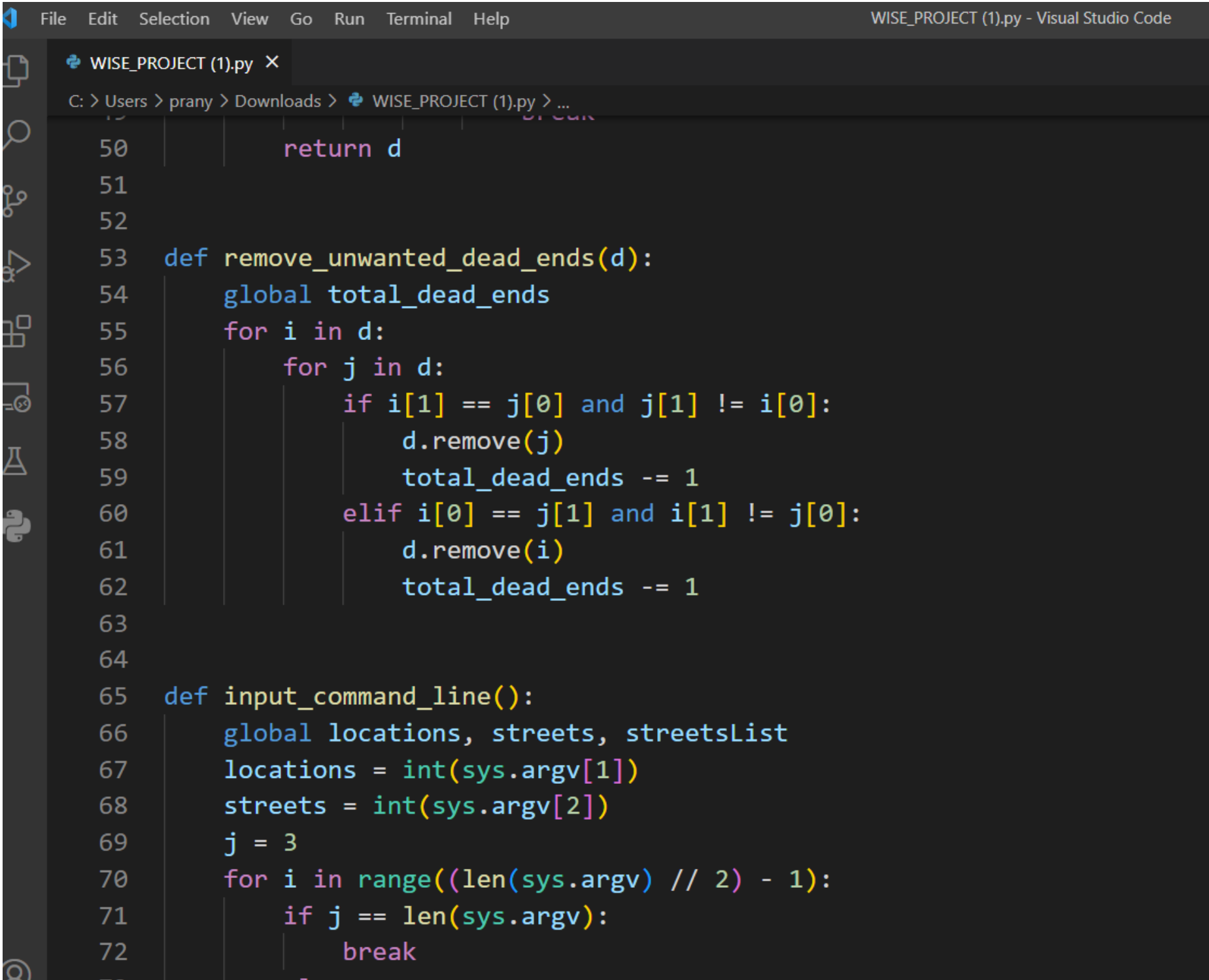
C: > Users > prany > Downloads > WISE_PROJECT (1).py > ...

```python
import sys

total_dead_ends = 0
dead_ends_list = []
locations = 0
streets = 0
streetsList = []
connections_matrix = []


def connections(l, s):
    global connections_matrix
    connections_matrix = [[0 for i in range(l + 1)] for j in range(l + 1)]
    for street in s:
        v = street[0]
        w = street[1]
        connections_matrix[v][w] = connections_matrix[w][v] = 1


def detect_dead_ends(c):
    for i in range(1, locations + 1):
        for j in range(1, locations + 1):
            if connections_matrix[i][j] == 1:
                if is_dead_end(i, [], [i, j]):
                    global total_dead_ends
                    total_dead_ends += 1
```

WISE_PROJECT (1).py

C: > Users > prany > Downloads > WISE_PROJECT (1).py > ...

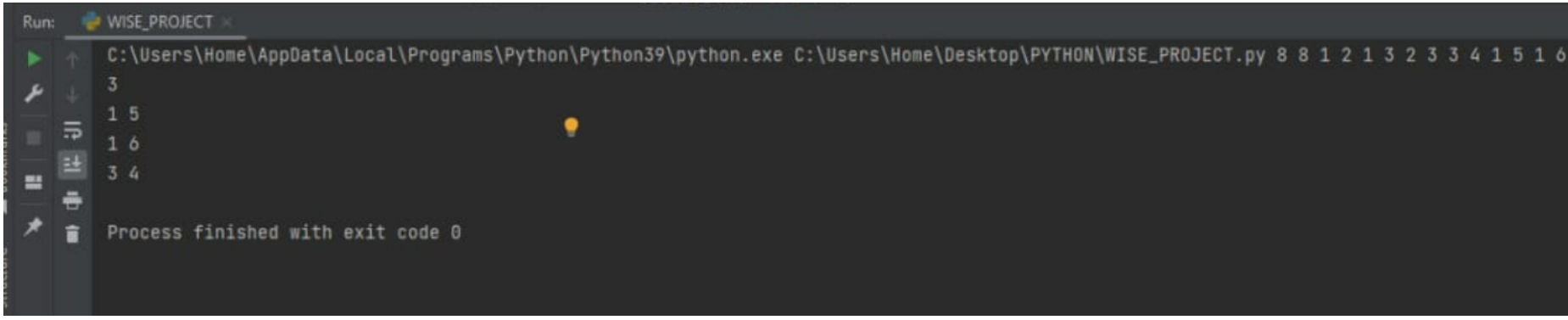
```
                    total_dead_ends += 1
                    dead_ends_list.append([i, j])


def is_dead_end(v, traversed, l):
    x = l[0]
    y = l[1]
    s = 0
    for i in range(1, locations + 1):
        s = s + connections_matrix[y][i]
    if s == 1:
        return True
    else:
        d = True
        for i in range(1, locations + 1):
            if connections_matrix[y][i] == 1 and i != x:
                if i == v:
                    d = False
                    break
                elif [y, i] not in traversed:
                    traversed.append([y, i])
                    d = is_dead_end(v, traversed, [y, i])
                    if not d:
                        break
        return d
```


File Edit Selection View Go Run Terminal Help
WISE_PROJECT (1).py - Visual Studio Code
WISE_PROJECT (1).py
C: > Users > prany > Downloads > WISE_PROJECT (1).py > ...
50 return d
51
52
53 def remove_unwanted_dead_ends(d):
54 global total_dead_ends
55 for i in d:
56 for j in d:
57 if i[1] == j[0] and j[1] != i[0]:
58 d.remove(j)
59 total_dead_ends -= 1
60 elif i[0] == j[1] and i[1] != j[0]:
61 d.remove(i)
62 total_dead_ends -= 1
63
64
65 def input_command_line():
66 global locations, streets, streetsList
67 locations = int(sys.argv[1])
68 streets = int(sys.argv[2])
69 j = 3
70 for i in range((len(sys.argv) // 2) - 1):
71 if j == len(sys.argv):
72 break

WISE_PROJECT (1).py

C: > Users > prany > Downloads > WISE_PROJECT (1).py > is_dead_end

```
72                  break
73              else:
74                  streetsList.append([int(sys.argv[j]), int(sys.argv[j + 1])])
75                  j += 2
76
77      connections(locations, streetsList)
78      detect_dead_ends(connections_matrix)
79      remove_unwanted_dead_ends(dead_ends_list)
80      print(total_dead_ends)
81      for i in dead_ends_list:
82          print(i[0], end=' ')
83          print(i[1])
84
85
86  input_command_line()
87
```


Run:
WISE_PROJECT
C:\Users\Home\AppData\Local\Programs\Python\Python39\python.exe C:\Users\Home\Desktop\PYTHON\WISE_PROJECT.py 8 8 1 2 1 3 2 3 3 4 1 5 1 6
3
1 5
1 6
3 4
Process finished with exit code 0

Thank you!!