

nVazriGeneticAlgorithm.py - nVazir - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

nVazriGeneticAlgorithm.py ×

nVazriGeneticAlgorithm.py > random_pick

You, 2 minutes ago | 2 authors (waqqasiq and others)

1import random

2

3def random_chromosome(size): #making random chromosomes

4return [random.randint(1, n_vazir) for _ in range(n_vazir)]

5

6def fitness(chromosome):

7horizontal_collisions = sum([chromosome.count(vazir)-1 for vazir in chromosome])/2

8diagonal_collisions = 0

9

10n = len(chromosome)

11left_diagonal = [0] * 2*n

12right_diagonal = [0] * 2*n

13for i in range(n):

14left_diagonal[i + chromosome[i] - 1] += 1

15right_diagonal[len(chromosome) - i + chromosome[i] - 2] += 1

16

17diagonal_collisions = 0

18for i in range(2*n-1):

19counter = 0

20if left_diagonal[i] > 1:

21counter += left_diagonal[i]-1

22if right_diagonal[i] > 1:

23counter += right_diagonal[i]-1

24diagonal_collisions += counter / (n-abs(i-n+1))

25

26return int(maxFitness - (horizontal_collisions + diagonal_collisions)) #28-(2+3)=23

27

28def probability(chromosome, fitness):

29return fitness(chromosome) / maxFitness

30

31def random_pick(population, probabilities):

32populationWithProbabilty = zip(population, probabilities)

33total = sum(w for c, w in populationWithProbabilty)

34r = random.uniform(0, total)

35upto = 0

36for c, w in zip(population, probabilities):

waqqasiq, 3 years ago • N-Queen problem/solution coded in python

PROBLEMS

OUTPUT

DEBUG CONSOLE

SQL CONSOLE

TERMINAL

====|====|====|====|====|====|====|====|====|====|
=== Generation 744 ===
i:1 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:2 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:3 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:4 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:5 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:6 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:7 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:8 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:9 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:10 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:11 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:12 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:13 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:14 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:15 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:16 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:17 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:18 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:19 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:20 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12

Maximum Fitness = 13
====|====|====|====|====|====|====|====|====|====|
=== Generation 745 ===
i:1 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:2 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:3 Chromosome = [2, 2, 5, 1, 3, 1], Fitness = 12
i:4 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:5 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:6 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:7 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:8 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:9 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:10 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:11 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:12 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:13 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:14 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:15 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:16 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13

OUTPUT

master

Python 3.8.10 64-bit

Connect

Git Graph

nVazir

Server not selected

waqqasiq, 3 years agoLn 35, Col 13Spaces: 4UTF-8LFPythonkite: not installed

nVazriGeneticAlgorithm.py - nVazir - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

nVazriGeneticAlgorithm.py

nVazriGeneticAlgorithm.py

random_pick

34r = random.uniform(0, total)

35upto = 0waqqasiq, 3 years ago • N-Queen problem/solution coded in python

36for c, w in zip(population, probabilities):

37if upto + w >= r:

38return c

39upto += w

40assert False, "Shouldn't get here"

41

42def reproduce(x, y): #doing cross_over between two chromosomes

43n = len(x)

44c = random.randint(0, n - 1)

45return x[0:c] + y[c:n]

46

47def mutate(x): #randomly changing the value of a random index of a chromosome

48n = len(x)

49c = random.randint(0, n - 1)

50m = random.randint(1, n)

51x[c] = m

52return x

53

54def genetic_vazir(population, fitness, m_p):

55mutation_probability = m_p

56new_population = []

57probabilities = [probability(n, fitness) for n in population]

58for i in range(len(population)):

59x = random_pick(population, probabilities) #best chromosome 1

60y = random_pick(population, probabilities) #best chromosome 2

61child = reproduce(x, y) #creating two new chromosomes from the best 2 chromosomes

62if random.random() < mutation_probability:

63child = mutate(child)

64print('i:{}'.format(i+1), end="\t")

65print_chromosome(child)

66new_population.append(child)

67if fitness(child) == maxFitness: break

68return new_population

69

70def print_chromosome(chrom):

PROBLEMS

OUTPUT

DEBUG CONSOLE

SQL CONSOLE

TERMINAL

====|====|====|====|====|====|====|====|====|====|
=== Generation 744 ===
i:1Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:2Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:3Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:4Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:5Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:6Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:7Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:8Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:9Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:10Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:11Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:12Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:13Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:14Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:15Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:16Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:17Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:18Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:19Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:20Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12

Maximum Fitness = 13
====|====|====|====|====|====|====|====|====|====|
=== Generation 745 ===
i:1Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:2Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:3Chromosome = [2, 2, 5, 1, 3, 1], Fitness = 12
i:4Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:5Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:6Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:7Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:8Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:9Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:10Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:11Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:12Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:13Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:14Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:15Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:16Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13

master

Python 3.8.10 64-bit

0/0

Connect

Git Graph

nVazir

Server not selected

waqqasiq, 3 years ago

Ln 35, Col 13 (1 selected)

Spaces: 4

UTF-8

LF

Python

kite: not installed

nVazriGeneticAlgorithm.py - nVazir - Visual Studio Code

File Edit Selection View Go Run Terminal Help

nVazriGeneticAlgorithm.py ×

nVazriGeneticAlgorithm.py > random_pick

```
70 def print_chromosome(chrom):
71     print("Chromosome = {}, Fitness = {}".format(str(chrom), fitness(chrom)))
72
73
74 if __name__ == "__main__":
75     n_vazir = int(input("Tedad Vazir: ")) # N V = 6
76     n_population = int(input("Jamiat har generation: ")) # N Population = 20
77     mutation_probability = float(input("Mutation Probability (n<1 ==> 0.03): ")) # Mutation Probabil
78     maxFitness = (n_vazir*(n_vazir-1))/2 # maxFitness = 6*5/2 = 15
79     population = [random_chromosome(n_vazir) for _ in range(n_population)]
80
81     generation = 1
82     while not maxFitness in [fitness(chrom) for chrom in population]:
83         print("=== Generation {} ===".format(generation))
84         population = genetic_vazir(population, fitness, mutation_probability)
85         print("")
86         print("Maximum Fitness = {}".format(max([fitness(n) for n in population])))
87         generation += 1
88         print("=====|=====|=====|=====|=====|=====|=====|=====|=====|")
89     chrom_out = []
90     print("Solved in Generation {}".format(generation-1))
91     for chrom in population:
92         if fitness(chrom) == maxFitness:
93             print("");
94             print("One of the solutions: ")
95             chrom_out = chrom
96             print_chromosome(chrom)
97
98     board = []
99
100     for x in range(n_vazir):
101         board.append(["x"] * n_vazir)
102
103     for i in range(n_vazir):
104         board[n_vazir-chrom_out[i]][i]="V"
105
106
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

SQL CONSOLE

TERMINAL

```
=====|=====|=====|=====|=====|=====|=====|=====|=====|
=== Generation 744 ===
i:1 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:2 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:3 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:4 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:5 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:6 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:7 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:8 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:9 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:10 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:11 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:12 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:13 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:14 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:15 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:16 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:17 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:18 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:19 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:20 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12

Maximum Fitness = 13
=====|=====|=====|=====|=====|=====|=====|=====|=====|
=== Generation 745 ===
i:1 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:2 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:3 Chromosome = [2, 2, 5, 1, 3, 1], Fitness = 12
i:4 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:5 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:6 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:7 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:8 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:9 Chromosome = [2, 4, 5, 1, 1, 5], Fitness = 12
i:10 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:11 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:12 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
i:13 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:14 Chromosome = [2, 4, 5, 1, 1, 1], Fitness = 11
i:15 Chromosome = [2, 4, 5, 1, 3, 1], Fitness = 13
i:16 Chromosome = [2, 4, 5, 1, 3, 5], Fitness = 13
```

waqqasiq, 3 years ago Ln 35, Col 13 (1 selected) Spaces: 4 UTF-8 LF Python kite: not installed

nVazriGeneticAlgorithm.py - nVazir - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

nVazriGeneticAlgorithm.py ×

nVazriGeneticAlgorithm.py > random_pick

```
80
81 generation = 1
82 while not maxFitness in [fitness(chrom) for chrom in population]:
83     print("=== Generation {} ===".format(generation))
84     population = genetic_vazir(population, fitness, mutation_probability)
85     print("")
86     print("Maximum Fitness = {}".format(max([fitness(n) for n in population])))
87     generation += 1
88     print("|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|")
89 chrom_out = []
90 print("Solved in Generation {}!".format(generation-1))
91 for chrom in population:
92     if fitness(chrom) == maxFitness:
93         print("");
94         print("One of the solutions: ")
95         chrom_out = chrom
96         print_chromosome(chrom)
97
98 board = []
99
100 for x in range(n_vazir):
101     board.append(["x"] * n_vazir)
102
103 for i in range(n_vazir):
104     board[n_vazir-chrom_out[i]][i]="V"
105
106
107 def print_board(board):
108     for row in board:
109         print (" ".join(row))
110
111 print()
112 print_board(board)
113 print()
114 print("Forked from {https://github.com/waqqasiq/n-queen-problem-using-genetic-algorithm}")
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

SQL CONSOLE

TERMINAL

PYTHON +

```
|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|
=== Generation 744 ===
i:1   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:2   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:3   Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:4   Chromosome = [2, 4, 5, 1, 1, 1],   Fitness = 11
i:5   Chromosome = [2, 4, 5, 1, 1, 1],   Fitness = 11
i:6   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:7   Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:8   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:9   Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:10  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:11  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:12  Chromosome = [2, 4, 5, 1, 1, 1],   Fitness = 11
i:13  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:14  Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:15  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:16  Chromosome = [2, 4, 5, 1, 1, 5],   Fitness = 12
i:17  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:18  Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:19  Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:20  Chromosome = [2, 4, 5, 1, 1, 5],   Fitness = 12

Maximum Fitness = 13
=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|
=== Generation 745 ===
i:1   Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:2   Chromosome = [2, 4, 5, 1, 1, 5],   Fitness = 12
i:3   Chromosome = [2, 2, 5, 1, 3, 1],   Fitness = 12
i:4   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:5   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:6   Chromosome = [2, 4, 5, 1, 1, 1],   Fitness = 11
i:7   Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:8   Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:9   Chromosome = [2, 4, 5, 1, 1, 5],   Fitness = 12
i:10  Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:11  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:12  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
i:13  Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:14  Chromosome = [2, 4, 5, 1, 1, 1],   Fitness = 11
i:15  Chromosome = [2, 4, 5, 1, 3, 1],   Fitness = 13
i:16  Chromosome = [2, 4, 5, 1, 3, 5],   Fitness = 13
```

OUTPUT

master

Python 3.8.10 64-bit

0x00

Connect

Git Graph

nVazir

Server not selected

waqqasiq, 3 years ago

Ln 35, Col 13 (1 selected)

Spaces: 4

UTF-8

LF

Python

kite: not installed

ActivitiesVisual Studio Codemem23%net1kB3k59%

13 Dec 15:59دوشنبه ۲۲ آذر

nVazriGeneticAlgorithm.py - nVazir - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

nVazriGeneticAlgorithm.py ×

nVazriGeneticAlgorithm.py > random_pick

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

```
generation = 1
while not maxFitness in [fitness(chrom) for chrom in population]:
    print("=== Generation {} ===".format(generation))
    population = genetic_vazir(population, fitness, mutation_probability)
    print("")
    print("Maximum Fitness = {}".format(max([fitness(n) for n in population])))
    generation += 1
    print("|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|")
chrom_out = []
print("Solved in Generation {}!".format(generation-1))
for chrom in population:
    if fitness(chrom) == maxFitness:
        print("");
        print("One of the solutions: ")
        chrom_out = chrom
        print_chromosome(chrom)

board = []

for x in range(n_vazir):
    board.append(["x"] * n_vazir)

for i in range(n_vazir):
    board[n_vazir-chrom_out[i]][i]="V"

def print_board(board):
    for row in board:
        print (" ".join(row))

print()
print_board(board)
print()
print("Forked from {https://github.com/waqqasiq/n-queen-problem-using-genetic-algorithm}")
```

PROBLEMSOUTPUTDEBUG CONSOLESQL CONSOLE

TERMINAL

i:12

Chromosome = [2, 4, 5, 1, 3, 5],

Fitness = 13

i:13

Chromosome = [2, 4, 5, 1, 3, 1],

Fitness = 13

i:14

Chromosome = [2, 4, 5, 1, 1, 1],

Fitness = 11

i:15

Chromosome = [2, 4, 5, 1, 3, 1],

Fitness = 13

i:16

Chromosome = [2, 4, 5, 1, 3, 5],

Fitness = 13

i:17

Chromosome = [2, 4, 5, 1, 3, 5],

Fitness = 13

i:18

Chromosome = [2, 4, 5, 1, 1, 1],

Fitness = 11

i:19

Chromosome = [2, 4, 5, 1, 3, 5],

Fitness = 13

i:20

Chromosome = [2, 4, 5, 1, 3, 1],

Fitness = 13

Maximum Fitness = 13

|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|

=== Generation 746 ===|

i:1

Chromosome = [2, 2, 5, 1, 3, 5],

Fitness = 12

i:2

Chromosome = [2, 4, 5, 1, 1, 1],

Fitness = 11

i:3

Chromosome = [2, 4, 5, 1, 3, 5],

Fitness = 13

i:4

Chromosome = [2, 4, 5, 1, 1, 5],

Fitness = 12

i:5

Chromosome = [2, 4, 5, 1, 3, 1],

Fitness = 13

i:6

Chromosome = [2, 4, 5, 1, 3, 1],

Fitness = 13

i:7

Chromosome = [2, 4, 5, 1, 1, 5],

Fitness = 12

i:8

Chromosome = [2, 4, 5, 1, 1, 1],

Fitness = 11

i:9

Chromosome = [2, 4, 6, 1, 3, 5],

Fitness = 15

Maximum Fitness = 15

|=====|=====|=====|=====|=====|=====|=====|=====|=====|=====|

Solved in Generation 746!

One of the solutions:

Chromosome = [2, 4, 6, 1, 3, 5], Fitness = 15

x x V x x x

x x x x x V

x V x x x x

x x x x V x

V x x x x x

x x x V x x

Forked from {https://github.com/waqqasiq/n-queen-problem-using-genetic-algorithm}

zakaria@legionrig:~/Documents/University/Semister 04/Artificial-Intelligence/HomeWork/nVazir\$

OUTPUT

masterPython 3.8.10 64-bit⊗⊕⚠0ConnectGit GraphnVazirServer not selected

waqqasiq, 3 years agoLn 35, Col 13 (1 selected)Spaces: 4UTF-8LFPythonkite: not installed