CSIS, BITS Pilani K. K. Birla Goa Campus

Artificial Intelligence (CS F407)

Programming Assignment 1
Total Marks: 15
Submission Deadline: 9 PM on 01/10/2022 (Saturday)

Each student must individually do this programming assignment. Your program must be written in Python and should run (without errors) on Python 3.6 or later.
Any form of plagiarism will result in 0 marks being awarded to everyone involved. There will be *no* differentiation between minor and major plagiarism.
Note that the deadline is **9 PM** and not midnight. Five marks per day will be deducted for submissions after the deadline. It will be your responsibility to submit the assignment well in advance and avoid unforeseen problems like power failures etc.

Question 1 (15 marks)

Vertex colouring problem: You are given a randomly generated undirected graph (V, E) where the vertex set V always contains 50 vertices. The edge set E contains edges that are randomly selected from all possible edges in the graph. You can assume that there are no self-loop edges in the graph. The fifty vertices are numbered from 0 to 49. The problem is to colour each vertex with any of the three colours — Red, Green and Blue — such that no two adjacent vertices have the same color. A state contains information about colour assignments for all the fifty vertices. The fitness function value for a state is the number vertices that are coloured such that no two adjacent vertices have the same colour. The goal state (if exists) will be the solution for the vertex colouring problem and will have a fitness function value of 50.

a) Implement the genetic algorithm for finding a state with the best fitness function value. First, implement the version of the genetic algorithm given in the textbook. Find how the algorithm performs when the number of edges in the graph increases. Run the algorithm for edge set sizes : 100, 200, 300, 400 and 500. (Use the *CreateGraphWithRandomEdges* function (see ROLLNO_NAME.py file) for generating a random graph with a given number of edges.) Plot the best fitness function value obtained for the genetic algorithm (with a population size 100 and for 50 generations) for graphs having the following number of edges : 100, 200, 300, 400 and 500. You should try at least 10 randomly generated graphs for any given edge set size. Also, plot how the best fitness function value changes across the 50 generations for the graphs with 100, 200, 300, 400 and 500 edges. Include the plot in the report.

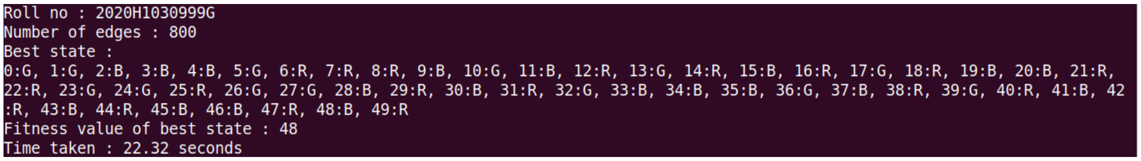
b) Think of ways in which the genetic algorithm can be improved. You can try varying the population size and number of generations as well. In the report, mention the different modifications that you have tried and what effect it had on the best state that could be found and the number of generations it took to find the best state. Include graphs in the report so that the report is easier to read. The graphs must be easy to understand and have proper labels and captions. Use 1.5 line spacing.

The improved algorithm that you submit for evaluation must terminate within 45 seconds. This means that you will have to check the time elapsed inside your program and terminate if the time elapsed crosses 45 seconds[1]. You can also terminate your algorithm if the best fitness function value does not change over several generations or the fitness value of 50 is reached.

Use the *CreateGraphWithRandomEdges* function (see ROLLNO_NAME.py file) to generate graphs with different number of edges. 3 test cases have also been given for you to run your code and see the results. You can vary the number of edges $|E|$, but the number of vertices $|V|$ must be 50.

Instructions for submission

- You must submit a single program file with the name "ROLLXYZ_FIRSTNAME.py". Your program should include only the improved version of the genetic algorithm. The program that you submit must read the edges from the 100.csv file given in the zipped test cases folder using the *ReadGraphfromCSVfile*() function (see ROLLNO_NAME.py file). As mentioned earlier, the program must terminate within 45 seconds. The output of the program that you submit should be as shown below:



```
Roll no : 2020H1030999G
Number of edges : 800
Best state :
0:G, 1:G, 2:B, 3:B, 4:B, 5:G, 6:R, 7:R, 8:R, 9:B, 10:G, 11:B, 12:R, 13:G, 14:R, 15:B, 16:R, 17:G, 18:R, 19:B, 20:B, 21:R,
22:R, 23:G, 24:G, 25:R, 26:G, 27:G, 28:B, 29:R, 30:B, 31:R, 32:G, 33:B, 34:B, 35:B, 36:G, 37:B, 38:R, 39:G, 40:R, 41:B, 42
:R, 43:B, 44:R, 45:B, 46:B, 47:R, 48:B, 49:R
Fitness value of best state : 48
Time taken : 22.32 seconds
```

- Your report must be named "ROLLXYZ_FIRSTNAME.pdf". Make sure to take a screenshot of the output of the all the test cases given to you and paste it in the report.

- Please use only capital letters for both the file names. Eg. 2020H1030999G_ADARSH.py and 2020H1030999G_ADARSH.pdf.

- Submit **only** the two files mentioned above. **Don't** zip the files. The assignment submission will be through quanta.

- 7.5 marks will be for the report and 7.5 marks will be for the output of your submitted program.

[1] This can be easily done in Python using very few lines of code.