National University

of Computer & Emerging Sciences-Islamabad
Chiniot-Faisalabad Campus

Dr. Hashim Yasin,

Department of Computer Science.

AI 2002: Artificial Intelligence

Assignment No. 3

Instructions:

1. Submit your assignment with a **report** within the due date and time. Late submission will result in a deduction of marks.
2. Mention your names and roll numbers clearly on your document.
3. Name your zip or folder/file, that you want to submit, according to the following format: **AI_A3_RollNo**
4. Try to solve each task of the assignment on your own in a group of a **maximum of two persons**.
5. No excuse or resubmission is permissible.
6. For programming questions, you can use any language for implementation.

Question No. 1: [Beyond Classical Search]

Implement the **Simulated Annealing Search Algorithm** in order to solve the **N-Queen** problem, where the number of queens may vary between $5 < \mathbf{N} < 10$, while the user provides the following information before starting,

- The number of iterations at run time and
- The number of queens (N).

0	0	0	1	0	0
0	1	0	0	0	0
0	0	0	0	0	1
1	0	0	0	0	0
0	0	0	0	1	0
0	0	1	0	0	0

Hints:

- The grid will be generated according to the number of the queens, i.e., if the N = 6, the 6×6 grid will be generated.

- In a 6×6 grid, the presence of a queen can be shown with 1 and 0 for empty cells as in the Figure given below.
- Neighbors can be generated by randomly moving any one of the Queen inside that column.

Question No. 2: [Adversial Search]

Build a **tic-tac-toe game** and design **rational agents** who play the game using **MINMAX** algorithm with **alpha-beta pruning**. Your program will show the following simulations:

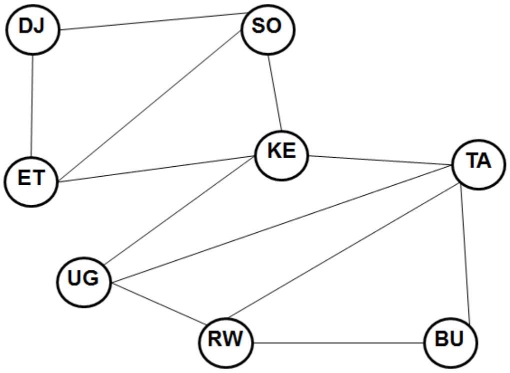
- How the game is played between two agents named MAX and MIN.
- What is the **score of each player** at each move?
- How the values of the **alpha and beta** are updated continuously and at what stage the alpha/beta cut-offs have been observed.

Hints:

- Rational agents may be considered as two computer-based agents. More precisely, you may create two instances of your agent in order to represent the opponents.
- Maintain the utilities of the opponents separately.

Question No. 3: [CSP]

Implement the constraint satisfaction problem of the following map coloring problem. The state space is represented as,



Variables: DJ, SO, ET, KE, UG , TA, RW, BU

Domains: Di = red; green; blue

Constraints: adjacent regions must have different colours, e.g.,

- $DJ \neq SO$
- $(DJ; SO) \in [(red;\ green);\ (red;\ blue);\ (green;\ red); (green;\ blue)\ \ldots\ ]$