

# Tutorial 4:

## Adversarial Search

Name	Student no.

For this tutorial, you can discuss the questions in small groups (up to 4 students). Individually submit the answers to each of the 6 Questions.

The objective of the tutorial is to represent the adversarial problem tic-tac-toe and define a heuristic and an informed search algorithm that can choose an appropriate move.

### Tic-tac-toe:

Two players take it in turns to mark a box on a three-by-three grid with a naught or a cross. The first player to get three-in-a-row wins.

X	O	
	X	
		O

### Question 1

Define a suitable data structure (representation) for the game

Use this representation in the following questions

### Question 2

Define a terminal test for the game (if the game is in this state, is the game over?)

### Question 3

Define a successor function for the game (what states can be reached from the current state?)

### Question 4

Define a heuristic function for the game (evaluation function: how good is this state?)

### Question 5

Write pseudo-code for a two-ply minimax search for one player of the game (from a given current state, the algorithm will look ahead two moves to choose the best current move to make)

- The algorithm will conduct a depth-first search to a depth of three
- At non-terminal nodes it will assign values of  $-\infty$  to maximising nodes (program's turn) and  $+\infty$  to minimising nodes (opponent's turn)
- At terminal nodes (depth=3) it will use the heuristic function to assign a value
- It will then conduct minimax to find maximum or minimum values for parent nodes (each child is compared to the parent to find the min or max)
- The algorithm will return the next move that should be made

### **Question 6**

Discuss how alpha-beta pruning would affect your minimax search of this problem  
Include space and time complexity in the discussion