

# CS 581 Spring 2024 Written Assignment #01

Due: **Saturday, February 10, 2024, 11:59 PM CST**  
Points: **30**

## Instructions:

1. Use this document template to report your answers. Name the complete document as follows:

LastName\_FirstName\_CS581\_WA01.doc or pdf

**ONLY PDF or MS Word file formats will be accepted.**

2. Submit the final document to Blackboard Assignments section before the due date.  
No late submissions will be accepted.

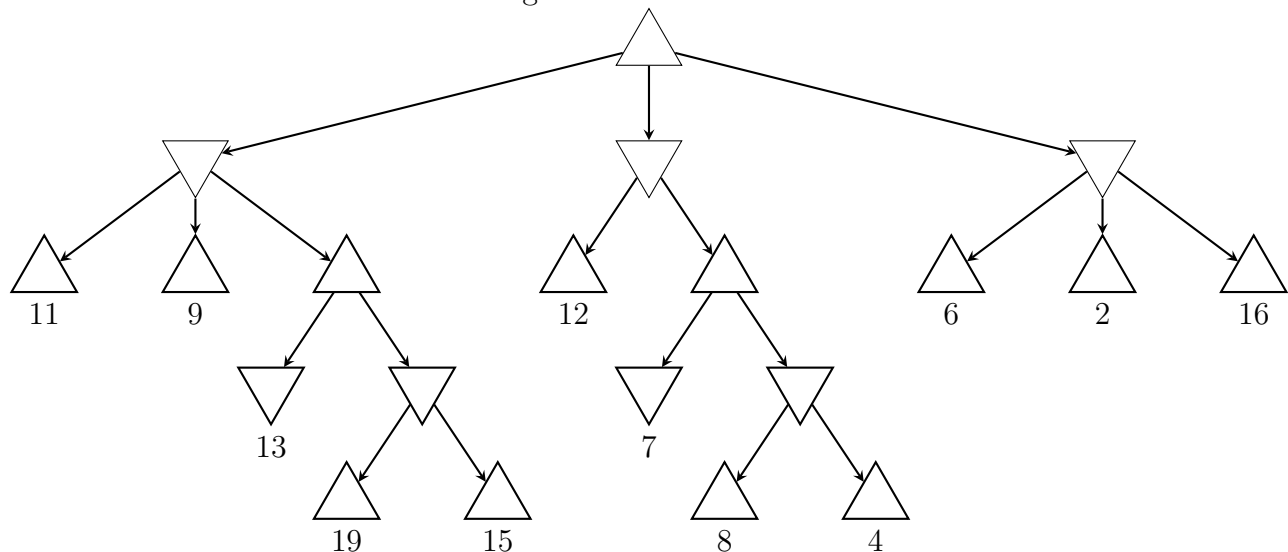
## Objectives:

1. (10 points) Demonstrate your understanding of MiniMax search algorithm,
2. (10 points) Demonstrate your understanding of A\* search algorithm,
3. (10 points) Demonstrate your understanding of a basic Genetic algorithm.

## Problem 1 [5 pts]:

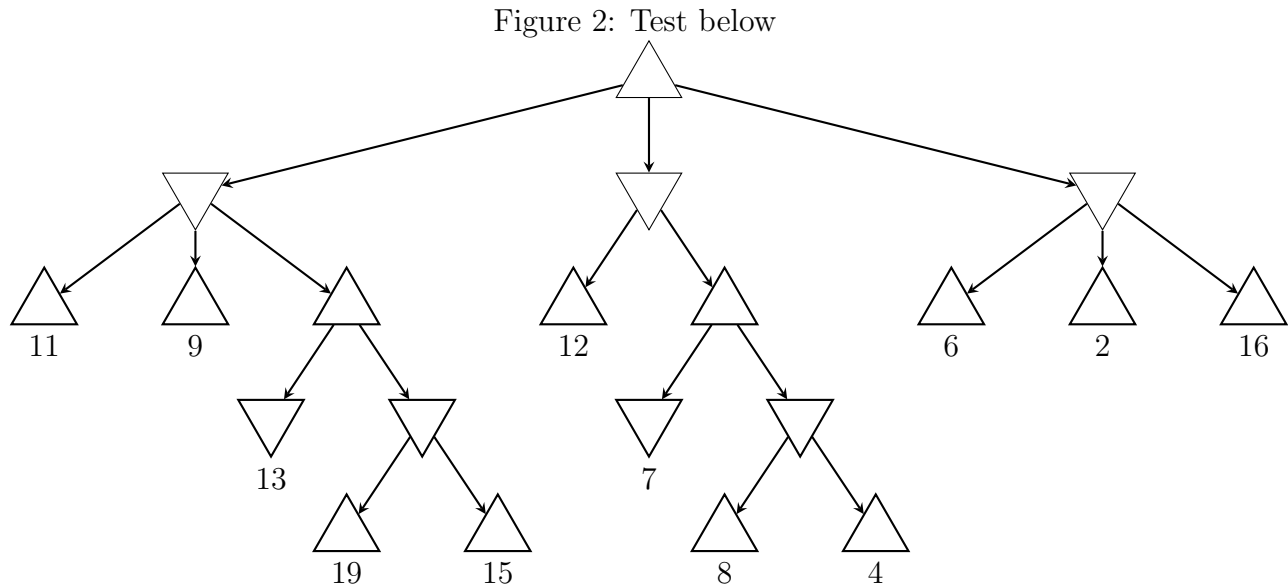
Apply MiniMax algorithm on the following game tree. What is the maximum utility that MAX can achieve, assuming MIN plays optimally?

Figure 1: Test below



## Problem 2 [5 pts]:

This is the same game as Problem 1. Hand trace the alpha-beta search. Show the updated bounds on the nodes. Clearly mark which branches are pruned, if any.



## Problem 3 [10 pts]:

Consider the following problem **state space** (undirected and weighted) graph (fig. 3) representing a map with cities (vertices) and roads (maps).

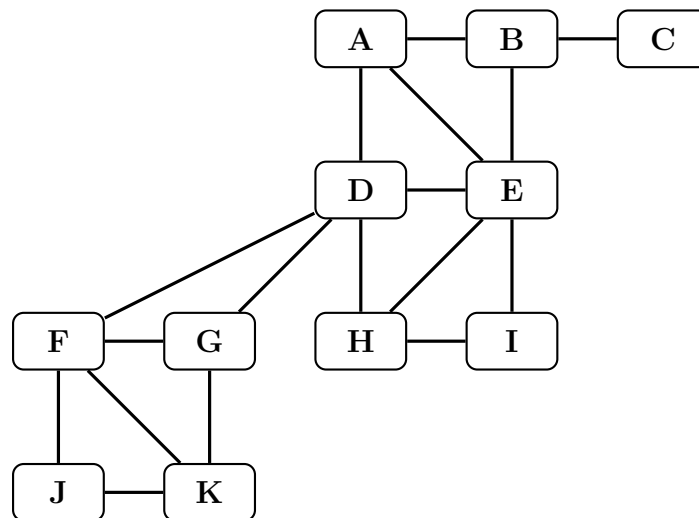


Figure 3: Problem state space (“cities and roads”).

The table (Table ??) below provides adjacency matrices for the state space graph above

(Driving distances) and a corresponding (but not shown) straight-line distances graph.

Data in matrices represents action cost and heuristic function values, respectively.

a) Driving distances												
	H	K	E	J	C	B	G	D	F	I	A	
H	0	0	102	0	0	0	0	114	0	87	0	
K	0	0	0	64	0	0	112	0	129	0	0	
E	102	0	0	0	0	68	0	170	0	50	180	
J	0	64	0	0	0	0	0	0	112	0	0	
C	0	0	0	0	0	164	0	0	0	0	0	
B	0	0	68	0	164	0	0	0	0	0	116	
G	0	112	0	0	0	0	0	205	127	0	0	
D	114	0	170	0	0	0	205	0	293	0	158	
F	0	129	0	112	0	0	127	293	0	0	0	
I	87	0	50	0	0	0	0	0	0	0	0	
A	0	0	180	0	0	116	0	158	0	0	0	
b) Straight-line distances												
	H	K	E	J	C	B	G	D	F	I	A	
H	0	234	93	278	229	116	151	82	242	66	172	
K	234	0	322	53	463	348	84	258	105	284	384	
E	93	322	0	368	149	63	242	139	335	41	152	
J	278	53	368	0	505	390	126	291	91	332	417	
C	229	463	149	505	0	116	380	230	458	191	138	
B	116	348	63	390	116	0	265	119	342	96	89	
G	151	84	242	126	380	265	0	176	113	206	301	
D	82	258	139	291	230	119	176	0	231	133	126	
F	242	105	335	91	458	342	113	231	0	305	353	
I	66	284	41	332	191	96	206	133	305	0	178	
A	172	384	152	417	138	89	301	126	353	178	0	

Table 1: Adjacency matrices for the problem.

**Your task:** Apply the A\* Search algorithm to the problem with following initial/goal states:

**initial state (IS):**  $F$

**goal state (GS):**  $I$

Show how the tree search develops:

- assume that **root node (corresponding to  $F$ )** was already dequeued from the **frontier** (see updated Reached structure below) and is ready to be expanded,
- show the search tree after first **TWO (2)** expansions,
- show changes in the frontier and reached/visited structures  
**BEFORE AND AFTER EVERY NODE EXPANSION**

Frontier structure [front $\leftarrow$ rear]												
Parent												
State												
$f()$												

Reached / visited												
Parent	-											
State	$F$											
distance from $IS$	0											

Show your work below (make sure it is legible)

Search Tree diagrams + structures

## Problem 4 [10 pts]:

You are solving an optimization problem using a basic Genetic Algorithm Algorithm parameters are:

- Individual representation:
  - binary with 16th bits,
  - first 8 bits correspond to a base<sub>2</sub> (binary) encoding of base<sub>10</sub> variable (“gene”)  $X$  value,
  - second 8 bits correspond to a base<sub>2</sub> (binary) encoding of base<sub>10</sub> variable (“gene”)  $Y$  value,
- Population size  $N = 6$ ,
- Fitness function:

$$f(X, Y) = -(X^2 + Y^2) + 28000$$

- Selection mechanism:
  - Order individuals according to their fitness in **descending order** (in case of ties: the individual that was first in unordered population goes first here as well)
  - offspring is created by pairing two subsequent parents with “wraparound”:
    - \*  $parent_1 + parent_2 \rightarrow child_1$ ,
    - \*  $parent_2 + parent_3 \rightarrow child_2$ ,
    - \*  $\dots$
    - \*  $parent_{N-1} + parent_N \rightarrow child_{N-1}$ ,

$$* \text{parent}_N + \text{parent}_1 \rightarrow \text{child}_N$$

- Probability of crossover  $P_c = 1$ ,
- Crossover mechanism: 2-point crossover with crossover points after the 4th and 12th bit (counting from the left),
- Probability of mutation  $P(m) = 0$ .

Your initial population is shown below:

Generation 1																
Individual 1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Individual 2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Individual 3	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
Individual 4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
Individual 5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Individual 6	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Generation 1 Evaluation				
Individual	$X$	$Y$	Fitness	Fitness Ratio [%]
Individual 1				
Individual 2				
Individual 3				
Individual 4				
Individual 5				
Individual 6				

Now, apply the Genetic Algorithm specified above. Stop after Generation 4 is created and evaluated:

- populate and show Generation and Generation Evaluation tables every time a new generation is created,
- generate  $Fitness = f(Generation)$  plot. It is enough to plot best individual of the generation's fitness.