CSS324: Artificial Intelligence
Midterm Mock Exam

curated by The Peanuts

Name.............ID.............Section.............Seat No.............

Conditions: Semi-closed Book

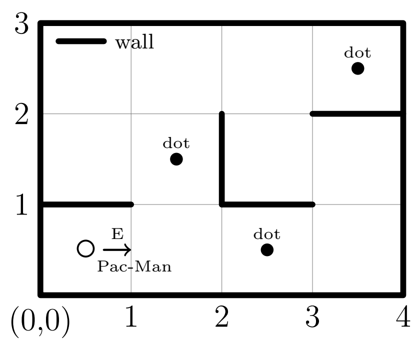
Directions:

1. This exam has 11 pages (including this page).
2. You may bring **one A4 cheat sheet**, written on both sides. If your handwriting is smaller than 2pt font, congratulations, you've basically invented microchips.
3. Calculators are NOT allowed.
4. Cheating is strictly prohibited. We have invisible drones watching (probably)
5. Draw search trees and diagrams where requested. Stick figures are acceptable only if they're funny.
6. Have fun!

*For solution, **click here**.*
Currently working on that.

Question 1

Your goal is to navigate Pac-Man through the following maze to collect all the dots. Each location in the maze is denoted by a coordinate (x, y). Pac-Man is initially placed at $(0, 0)$ and is facing east (right direction). The maze contains dots at coordinates $(1, 1)$, $(2, 0)$, and $(3, 2)$. Pac-Man must collect all dots to win the game.



Pac-Man accepts only four commands: (1) move forward 1 block, (2) rotate 90° clockwise, (3) rotate 180° clockwise, and (4) rotate 270° clockwise. Pac-Man cannot move through walls or outside the maze boundaries.
Formulate this problem so that it can work with the search algorithms studied in class without any modification.

(a) Define how to represent a state and write the initial state.

(b) Explain how to generate a set of successors from a given state.

(c) What is the depth of the shallowest goal of this problem? Explain your reasoning for how you obtained this value.

(d) Is this search problem more suitable for breadth-first search or depth-first search? Justify your answer.

Question 2

Taylor Swift is planning her Eras Tour and needs to visit cities $\{A, B, C, D, E, F\}$ exactly once, starting and ending at city A. However, she has special requirements: she must perform at least 2 consecutive concerts in cities that have populations over 1 million (cities B, D, and F). The travel costs between cities are given in the following matrix:

	A	B	C	D	E	F
A	0	15	20	25	30	35
B	15	0	10	18	22	28
C	20	10	0	12	15	25
D	25	18	12	0	8	16
E	30	22	15	8	0	20
F	35	28	25	16	20	0

(a) Formulate this as a search problem by defining:

(a) State representation

(b) Initial state

(c) Goal test

(d) Actions and successor function

(e) Step cost

(b) Draw the search tree when conducting Uniform-Cost Graph Search on this problem. Expand at least 8 nodes and show the frontier contents after each expansion.

(c) Design an admissible heuristic function for this problem that takes into account the constraint about consecutive large city concerts.

Question 3

The following code attempts to implement A* Search, but ***contains several errors***. Identify and correct all the mistakes to make it work properly.

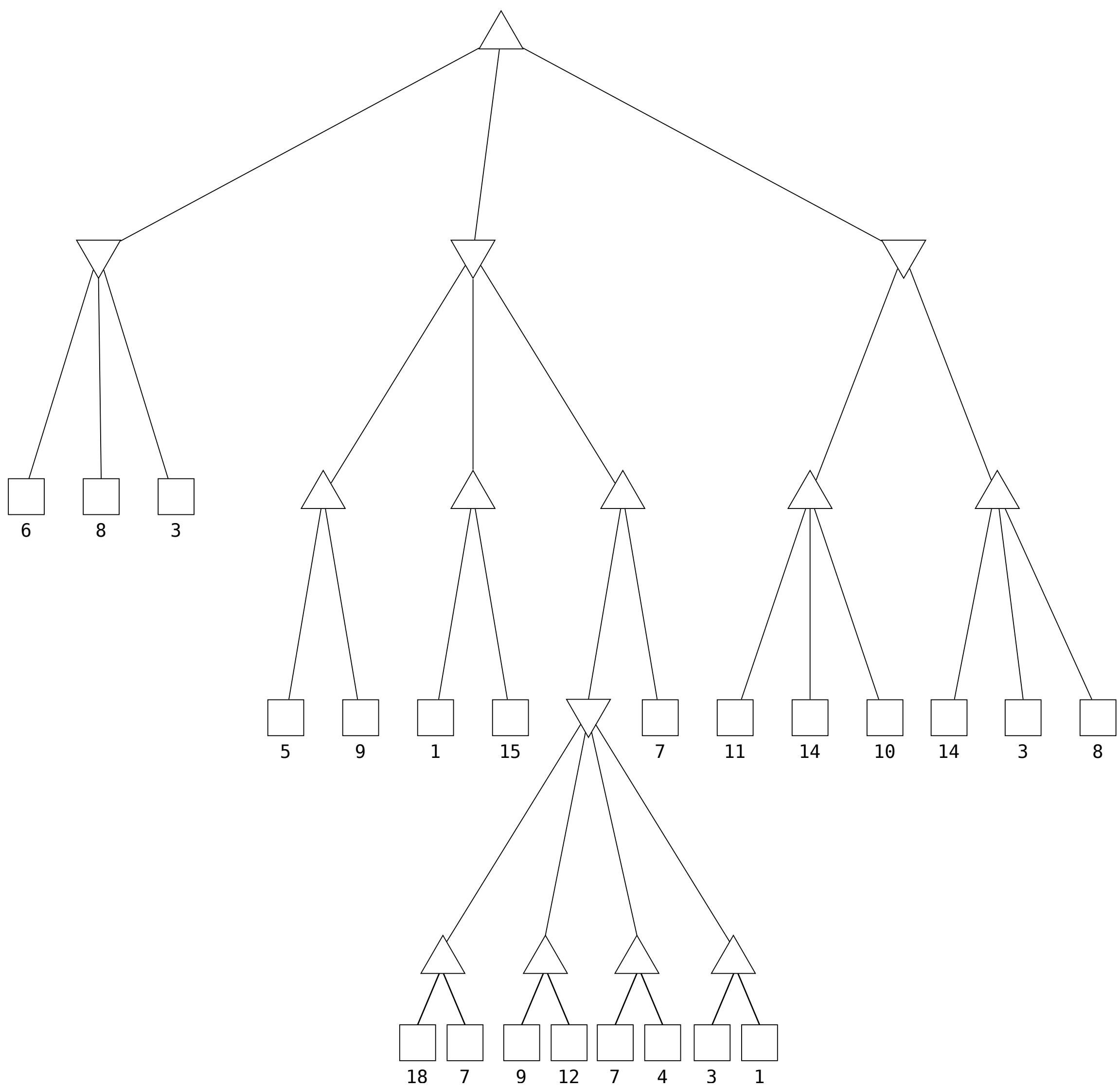
```python
from heapq import heappush, heappop

def astar_search(initial_state: State, heuristic_func) -> Node:
    initial_node = Node(initial_state, None, 0, 0)

    # Priority queue with path cost only
    frontier = [(0, initial_node)]
    explored = set()

    while frontier:
        current_cost, node = heappop(frontier)

        if node.state.is_goal():
            return node

        if node.state in explored:
            continue

        explored.add(node.state)

        for child_state, step_cost in node.state.successors():
            child_cost = node.path_cost + step_cost
            child_node = Node(child_state, node, child_cost, node.depth+1)

            # Add to frontier with only path cost
            heappush(frontier, (child_cost, child_node))

    return None
```

(a) List all the errors in the above code and provide the corrected version.

(b) Explain the key difference between this corrected A* implementation and Uniform-Cost Search.

Question 4

Consider the following game tree, use alpha-beta pruning algorithm to cross out unnecessary subtrees. Assume that the tree is evaluated by depth-first search algorithm from left to right.



Question 5

At S2T University, there's a computer science department where students, professors, and courses interact in various ways. Use the following predicates:

- $Student(x)$ - "x is a student"
- $Professor(x)$ - "x is a professor"
- $Course(x)$ - "x is a course"
- $Enrolled(x, y)$ - "student x is enrolled in course y"
- $Teaches(x, y)$ - "professor x teaches course y"
- $Smart(x)$ - "x is smart"
- $Difficult(x)$ - "course x is difficult"
- $Likes(x, y)$ - "x likes y"

(a) Translate the following English sentences into First-Order Logic:

(1) "Every student is enrolled in at least one course"

(2) "Some professors teach only difficult courses"

(3) "No student likes all difficult courses"

(4) "There exists a professor who is liked by all students in at least one course they teach"

(b) Translate the following FOL sentences into plain English:

(1) $\forall x \exists y (Student(x) \rightarrow (Course(y) \wedge Enrolled(x, y) \wedge Difficult(y)))$

(2) $\exists x \forall y \forall z (Professor(x) \wedge Course(y) \wedge Student(z) \wedge Teaches(x, y) \wedge Enrolled(z, y) \rightarrow Smart(z))$

(3) $\forall x \exists y \exists z (Student(x) \wedge Professor(y) \wedge Course(z) \wedge Enrolled(x, z) \wedge Teaches(y, z) \wedge \neg Likes(x, y))$

(c) Consider the statement: “Every smart student is enrolled in exactly two courses.” Write this in FOL using appropriate quantifiers and equality. Then, write the negation of this statement in both FOL and plain English.