Queen's University
School of Computing

CISC 352– QUIZ #1
Artificial Intelligence

Name: _____

Student Number: _____

This test contains 9 pages (including this cover page) and 13 questions. The total points is 100. Please answer only in the answer boxes provided. You may use the back of the pages as scrap paper. **This is a closed-book test. No computers or calculators are allowed.** Should a question be unclear or ambiguous, you should make a *reasonable* interpretation and state what you have assumed. To be eligible for re-marking, this test must be answered entirely in indelible (unerasable) ink. If erasable ink or pencil is used, then the test will be marked exactly once.

..... For Marker Use Only .....

Distribution of Marks

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	6	
6	6	
7	6	
8	6	
9	9	
10	9	
11	9	
12	9	
13	0	
Total:	100	

This is a **45 minute** test

Do not begin until instructed to do so.

Note: *The following 4 questions correspond to Assignment 1. If you are doing a course project and have chosen to not submit this assignment, please leave a checkmark (or happy face!) next to this sentence, and leave the "assignment questions" blank.*

1. (10 points) Given that the grid is n*n, how would you specify the list of domains for the **binary-not-equal** constraint?

   Enter answer into multiple choice box on right side.
   Correct 1st Choice Box = 6 point(s).
   Correct 2nd Choice Box = 4 point(s).
   If you are confident in your answer, fill in both boxes with your single choice.

   (a) domains = list(range(1, n + 1))

   (b) domains = list(range(0, n + 1))

   (c) domains = list(range(0, n)

   (d) domains = list(range(1, n))

1st Choice	2nd Choice

2. (10 points) In function binary_ne_grid, how many satisfying tuples does each constraint have (assuming the grid is n*n)?
   Hint: recall the method def add_satisfying_tuples(self, tuples) in class Constraint.

   Enter answer into multiple choice box on right side.
   Correct 1st Choice Box = 6 point(s).
   Correct 2nd Choice Box = 4 point(s).
   If you are confident in your answer, fill in both boxes with your single choice.

   (a) $n^2 - 1$

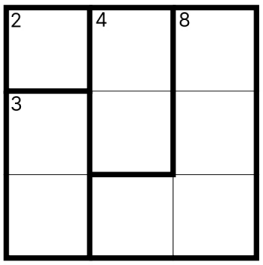
   (b) $n \cdot (n - 1)$

   (c) $n!$

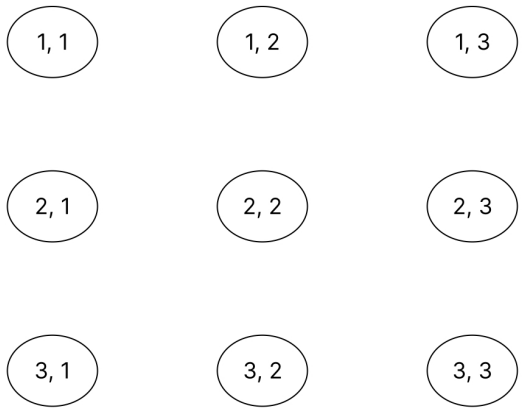
   (d) n^2

1st Choice	2nd Choice

3. (10 points) Given the following 3x3 Cagey grid, create a constraint graph for **Cagey** constraints.



Please draw your edges with the nodes given. Each node here denotes a variable and its coordinates entail its location in the grid.



4. (10 points) Find and describe **one bug** in the implementation of **forward checking**. Where applicable, please provide the line number(s) where the bug is. There is only one bug in this piece of code.

```
1  def prop_FC(csp, newVar=None):
2      '''Do forward checking. That is check constraints with
3         only one uninstantiated variable. Remember to keep
4         track of all pruned variable,value pairs and return '''
5
6       # find constraints
7      if newVar == None:
8          constraints = csp.get_all_cons()
9      else:
10         constraints = csp.get_cons_with_var(newVar)
11
12     # keep track of all pruned variables
13     path = []
14
15     for constraint in constraints:
16         # only one uninstantiated variable
17         if constraint.get_n_unasgn() == 1:
18             unassignedVar= constraint.get_unasgn_vars()[0]
19
20             for val in unassignedVar.cur_domain():
21                 # check if there are still satisfying values
22                 if constraint.check_var_val(unassignedVar, val):
23                     unassignedVar.prune_value(val)
24                     # update path
25                     path.append((unassignedVar, val))
26
27             # check for deadend
28             if unassignedVar.cur_domain_size() == 0:
29                 return False, path
30
31 # path found, return True
32 return True, path
```

Note: *The remaining questions correspond to general content, and every student in the class should complete them.*

For the following four questions, let's consider a scenario with a robotic arm operating in a 3D grid, sized 3x3x3. In this space, each cell can either be empty or contain an object. The robot, which can move in this grid and hold a single object, is capable of performing these actions:

1. Move one unit up, down, left, right, forward, or backward, staying within grid boundaries.
2. Pick up an object from its current cell, provided the cell contains an object and the robot isn't already holding one.
3. Place an object in its current cell, assuming the cell is empty and the robot is holding an object.

5. (6 points) What is the total state space size?

Enter answer into multiple choice box on right side.
Correct 1st Choice Box = 4 point(s).
Correct 2nd Choice Box = 2 point(s).
If you are confident in your answer, fill in both boxes with your single choice.

(a) $27^2 \times 27$

(b) $27^2 \times 27 \times 2$

(c) $2^{27} \times 27 \times 2$

(d) $2^{27} \times 27$

1st Choice	2nd Choice

6. (6 points) What is the maximum branching factor?

Enter answer into multiple choice box on right side.
Correct 1st Choice Box = 4 point(s).
Correct 2nd Choice Box = 2 point(s).
If you are confident in your answer, fill in both boxes with your single choice.

(a) 7

(b) 9

(c) 8

(d) 6

1st Choice	2nd Choice

Consider a modification of the above scenario where the robot can now hold two objects.

7. (6 points) What is the total state space size?

Enter answer into multiple choice box on right side.
Correct 1st Choice Box = 4 point(s).
Correct 2nd Choice Box = 2 point(s).
If you are confident in your answer, fill in both boxes with your single choice.

(a) $2^{27} \times 27 \times 2$

(b) $2^{27} \times 27 \times 5$

(c) $2^{27} \times 27 \times 3$

(d) $27^2 \times 27 \times 3$

1st Choice	2nd Choice

8. (6 points) What is the maximum branching factor?

Enter answer into multiple choice box on right side.
Correct 1st Choice Box = 4 point(s).
Correct 2nd Choice Box = 2 point(s).
If you are confident in your answer, fill in both boxes with your single choice.

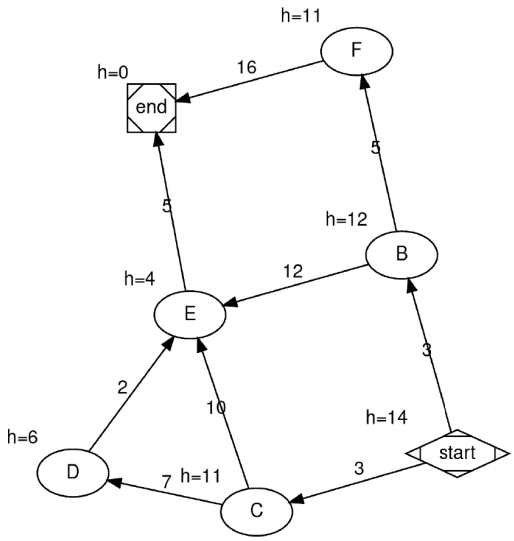
(a) 6

(b) 8

(c) 9

(d) 7

1st Choice	2nd Choice



9. (9 points) Given the graph above, list the sequence of nodes expanded from a **Breadth First Search** tree search of the graph. Report the found path and its cost from start to end. Please break any ties by first selecting the node with the letter that comes FIRST in alphabetical order. If relevant, the "end node should always be selected first to break a tie.

10. (9 points) Given the graph above, list the sequence of nodes expanded from a **Greedy** tree search of the graph. Report the found path and its cost from start to end. Please break any ties by first selecting the node with the letter that comes FIRST in alphabetical order. If relevant, the "end" node should always be selected first to break a tie.

11. (9 points) Given the graph above, list the sequence of nodes expanded from a **A*** tree search of the graph. Report the found path and its cost from start to end. Please break any ties by first selecting the node with the letter that comes FIRST in alphabetical order. If relevant, the "end" node should always be selected first to break a tie.

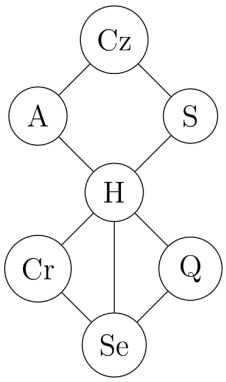
12. (9 points) The following is a constraint graph for a map coloring problem with variable domains {r, g, b}. The values remaining in each domain are given below.

    Given a partial assignment Q = b, S = r:

    1. Which variable would be selected next using the **maximum degree heuristic**?
    2. What color will this variable be assigned?
    3. What are the remaining values in each variable domain after the assignment and forward checking is applied?

    Note: Please break any ties by first selecting the variable or color with a name that comes FIRST in alphabetical order.

Variable	Cz	S	A	H	Cr	Se	Q
Domain	g,b	r	r,g,b	g	r,g,b	r,g	b



13. (5 bonus points) Give an example of a utility function for an intelligent agent using a problem setting not given in the class lectures and describe how an agent blindly following that utility function can have unintended consequences. Please fit your answer inside the box.

Queen's University
School of Computing

CISC 352– QUIZ #1
Artificial Intelligence

Name: _____

Student Number: _____

This test contains 9 pages (including this cover page) and 13 questions. The total points is 100. Please answer only in the answer boxes provided. You may use the back of the pages as scrap paper. **This is a closed-book test. No computers or calculators are allowed.** Should a question be unclear or ambiguous, you should make a *reasonable* interpretation and state what you have assumed. To be eligible for re-marking, this test must be answered entirely in indelible (unerasable) ink. If erasable ink or pencil is used, then the test will be marked exactly once.

..... For Marker Use Only .....

Distribution of Marks

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	6	
6	6	
7	6	
8	6	
9	9	
10	9	
11	9	
12	9	
13	0	
Total:	100	

This is a **45 minute** test

Do not begin until instructed to do so.

Note: *The following 4 questions correspond to Assignment 1. If you are doing a course project and have chosen to not submit this assignment, please leave a checkmark (or happy face!) next to this sentence, and leave the "assignment questions" blank.*

1. (10 points) Given that the grid is n*n, how would you specify the list of domains for the **nary-all-different** constraint?

   Enter answer into multiple choice box on right side.
   Correct 1st Choice Box = 6 point(s).
   Correct 2nd Choice Box = 4 point(s).
   If you are confident in your answer, fill in both boxes with your single choice.

   (a) domains = list(range(1, n))

   (b) domains = list(range(0, n + 1))

   (c) domains = list(range(0, n)

   (d) domains = list(range(1, n + 1))

1st Choice	2nd Choice

2. (10 points) In function nary_ad_grid, how many satisfying tuples does each constraint have (assuming the grid is n*n)?
   Hint: recall the method def add_satisfying_tuples(self, tuples) in class Constraint.

   Enter answer into multiple choice box on right side.
   Correct 1st Choice Box = 6 point(s).
   Correct 2nd Choice Box = 4 point(s).
   If you are confident in your answer, fill in both boxes with your single choice.

   (a) $n!$

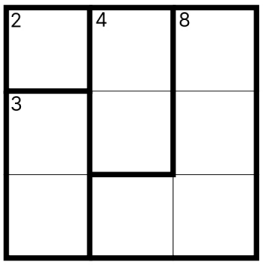
   (b) n^2

   (c) $n^2 - 1$

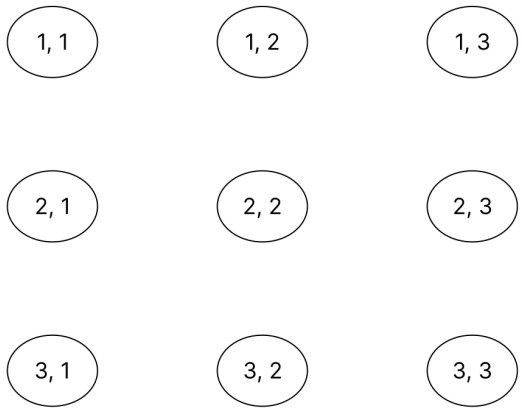
   (d) $n \cdot (n - 1)$

1st Choice	2nd Choice

3. (10 points) Given the following 3x3 Cagey grid, create a constraint graph for **Cagey** constraints.



Please draw your edges with the nodes given. Each node here denotes a variable and its coordinates entail its location in the grid.



4. (10 points) Find and describe **one bug** in the implementation of **forward checking**. Where applicable, please provide the line number(s) where the bug is. There is only one bug in this piece of code.

```
1  def prop_FC(csp, newVar=None):
2      '''Do forward checking. That is check constraints with
3         only one uninstantiated variable. Remember to keep
4         track of all pruned variable,value pairs and return '''
5
6       # find constraints
7      if newVar == None:
8          constraints = csp.get_all_cons()
9      else:
10         constraints = csp.get_cons_with_var(newVar)
11
12     # keep track of all pruned variables
13     path = []
14
15     for constraint in constraints:
16         # only one uninstantiated variable
17         if constraint.get_n_unasgn() == 1:
18             unassignedVar= constraint.get_unasgn_vars()[0]
19
20             for val in unassignedVar.cur_domain():
21                 # check if there are still satisfying values
22                 if constraint.check_var_val(unassignedVar, val):
23                     unassignedVar.prune_value(val)
24                     # update path
25                     path.append((unassignedVar, val))
26
27             # check for deadend
28             if unassignedVar.cur_domain_size() == 0:
29                 return False, path
30
31 # path found, return True
32 return True, path
```

Note: *The remaining questions correspond to general content, and every student in the class should complete them.*

For the following four questions, imagine a search and rescue robot navigating a 5x5 maze. Each cell of the maze can be one of three types: open space, obstacle, or one of the target locations. The robot can perform the following actions:

1. Move one cell north, south, east, or west if the adjacent cell is open space.
2. Scan the adjacent cells for a target location (all adjacent cells are simultaneously scanned with a single action).

The initial state of the maze is known, with the robot starting in the bottom-left corner and target locations are randomly placed. There can be multiple obstacles and targets. The objective is for the robot to find the shortest path to a target location.

5. (6 points) What is the total state space size?

Enter answer into multiple choice box on right side.
Correct 1st Choice Box = 4 point(s).
Correct 2nd Choice Box = 2 point(s).
If you are confident in your answer, fill in both boxes with your single choice.

(a) 3^{25}

(b) $3^{25} \times 25$

(c) 25^3

(d) $2^{25} \times 25$

1st Choice	2nd Choice

6. (6 points) What is the maximum branching factor?

Enter answer into multiple choice box on right side.
Correct 1st Choice Box = 4 point(s).
Correct 2nd Choice Box = 2 point(s).
If you are confident in your answer, fill in both boxes with your single choice.

(a) 5

(b) 9

(c) 8

(d) 4

1st Choice	2nd Choice

Consider the simplified version of the domain where there are no obstacles and we just care about pathfinding for the robot itself.

7. (6 points) What is the total state space size?

   Enter answer into multiple choice box on right side.
   Correct 1st Choice Box = 4 point(s).
   Correct 2nd Choice Box = 2 point(s).
   If you are confident in your answer, fill in both boxes with your single choice.

   (a) 3^{25}

   (b) $3^{25} \times 25$

   (c) 25

   (d) 25^3

1st Choice	2nd Choice

8. (6 points) What is the maximum branching factor?

   Enter answer into multiple choice box on right side.
   Correct 1st Choice Box = 4 point(s).
   Correct 2nd Choice Box = 2 point(s).
   If you are confident in your answer, fill in both boxes with your single choice.

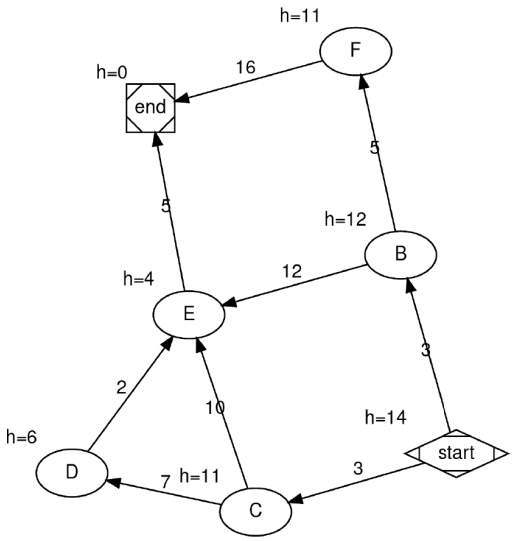
   (a) 4

   (b) 8

   (c) 5

   (d) 9

1st Choice	2nd Choice



9. (9 points) Given the graph above, list the sequence of nodes expanded from a **Breadth First Search** tree search of the graph. Report the found path and its cost from start to end. Please break any ties by first selecting the node with the letter that comes FIRST in alphabetical order. If relevant, the "end node should always be selected first to break a tie.

10. (9 points) Given the graph above, list the sequence of nodes expanded from a **Uniform Cost Search** tree search of the graph. Report the found path and its cost from start to end. Please break any ties by first selecting the node with the letter that comes FIRST in alphabetical order. If relevant, the "end node should always be selected first to break a tie.

11. (9 points) Given the graph above, list the sequence of nodes expanded from a **A*** tree search of the graph. Report the found path and its cost from start to end. Please break any ties by first selecting the node with the letter that comes FIRST in alphabetical order. If relevant, the "end" node should always be selected first to break a tie.

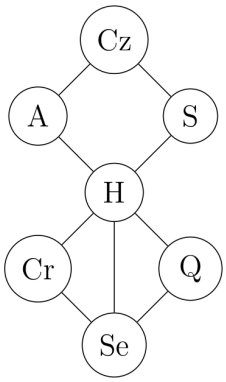
12. (9 points) The following is a constraint graph for a map coloring problem with variable domains {r, g, b}. The values remaining in each domain are given below.

    Given a partial assignment A = g, Cz = b:

    1. Which variable would be selected next using the **maximum degree heuristic**?
    2. What color will this variable be assigned?
    3. What are the remaining values in each variable domain after the assignment and forward checking is applied?

    Note: Please break any ties by first selecting the variable or color with a name that comes FIRST in alphabetical order.

Variable	Cz	S	A	H	Cr	Se	Q
Domain	b	r,g	g	r,b	r,g,b	r,g,b	r,g,b



13. (5 bonus points) Give an example of a utility function for an intelligent agent using a problem setting not given in the class lectures and describe how an agent blindly following that utility function can have unintended consequences. Please fit your answer inside the box.