# CS 6601 Midterm – Fall 2020

*Please read the following instructions thoroughly.*

Fill out this PDF form and submit it on [Gradescope](). Remember to also submit on Canvas.

You have unlimited resubmissions until the deadline. You can: **(a)** type directly into the form – we highly recommend using Adobe Reader DC (or Master PDF on Linux). Other programs may not save your answers, so **please keep a backup**; or **(b)** print, hand-write & scan. You can combine the methods as well.

**Submit only a single PDF** – no phone pictures, please! (You may use an app like CamScanner or Office Lens if you do not have scanner access.) Do not add pages unless absolutely necessary; if you do, please add them at the end of the exam **only**, and clearly label **both** the extra page and the original question page. Submit **ALL** pages of the exam, not only the completed ones.

**Do not forget to fill the checklist at the end before turning in the exam.** The exam may not be graded if it is left blank.

The exam is open-book, open-note, open video lectures, with no time limit aside from the open period. No internet use is allowed, except for e-text versions of the textbook, this semester's CS6601 course materials, Piazza, and any links provided in the PDF itself. No resources outside this semester's 6601 class should be used. Do not discuss the exam on Piazza, Slack, or any other form of communication. If there is a question for the teaching staff, **please make it private on Piazza and tag it as Midterm Exam with the question number in the subject line** (for example, a question on Search would be "Midterm Exam #2"). Please make **different posts for different questions.**

**Please round all your final answers to 6 decimal places, don't round intermediate results**. You can use `round(your_number, 6)` function in Python for help.
**You will not receive full credit if your answers are not given to the specified precision.**

**Point breakdown** (Each question has sub-parts with varying points):

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Bonus | Total |
|---|---|---|---|---|---|---|---|---|
| **Pts** | 16 | 20 | 15 | 14 | 20 | 15 | 6 | 100 |

# 1. Game Playing

**(16 points)**

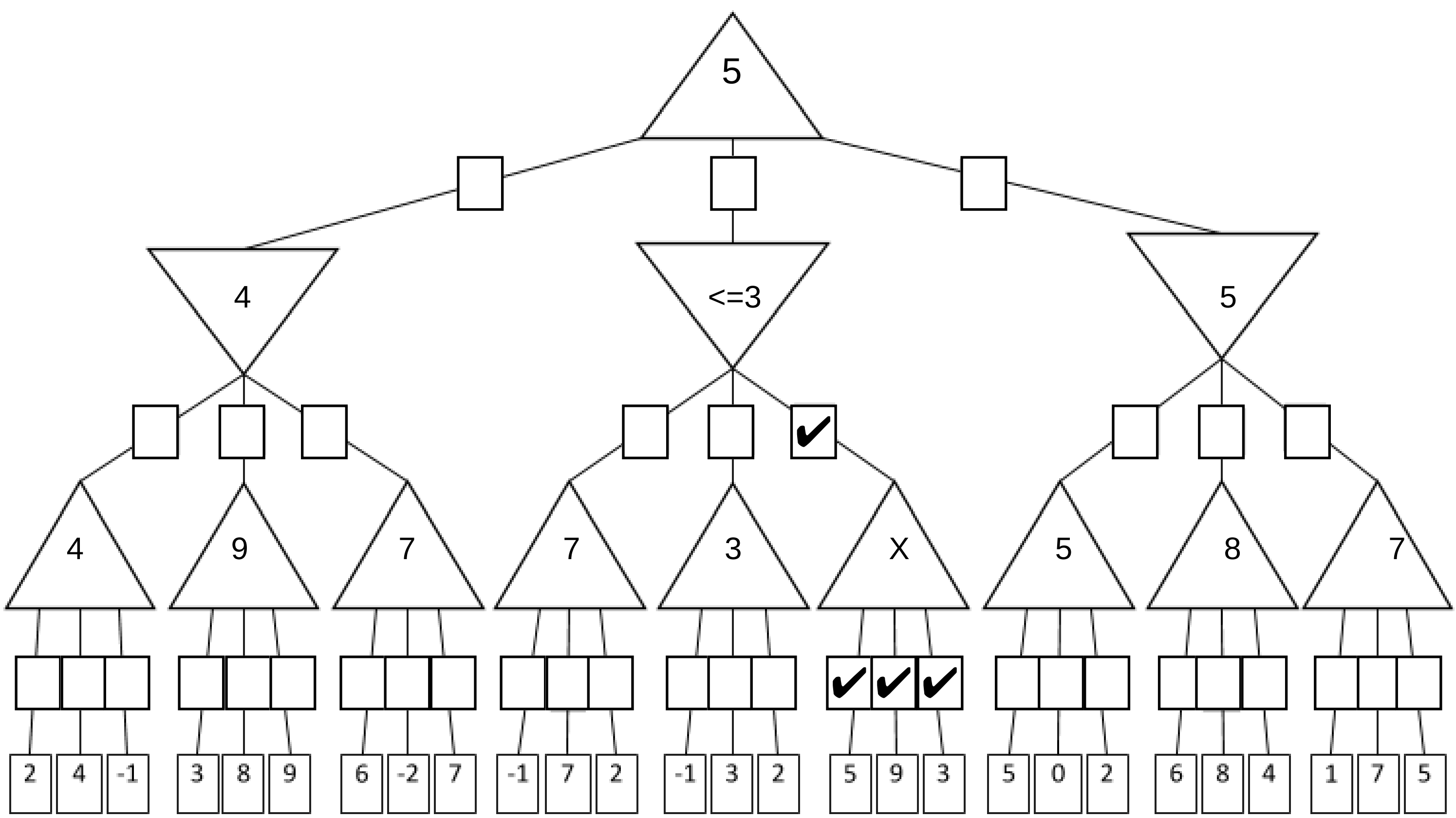
## Question A

**(6 points)**

Solve the game tree using alpha-beta pruning. The algorithm evaluates the branches from left to right. Indicate which branches are pruned by checking the correct box. Show the value of each node. Make sure to use inequality signs if appropriate.

When you prune a branch at a top level, **check all the sub boxes/branches down along that path as well**. If an upper level node in a branch gets pruned and you've selected the checkbox to indicate that it is pruned, you do not need to fill-in values for the "unvisited" underlying nodes, but you should check all the boxes to indicate which branches are pruned.

**1.A**



## Question B (Three Player Game Tree)

**(10 points)**

There are four boxes: A, B, C, and D.

A contains four bills (\$1, \$5, \$10 and \$100);
B contains four bills (\$1, \$20, \$50 and \$100);
C contains four bills (\$5, \$10, \$50 and \$200);
D contains four bills (\$50, \$60, \$80 and \$100);

Suppose three players Amy, Bobby and Chris all know what bills are in which box.

**Step 1.** Amy picks one of the four boxes.
**Step 2.** Bobby picks one bill from the box (that Amy picked) and gives the bill to Chris.
**Step 3.** Of the remaining bills in the box, Chris gives one bill to Amy, one bill to Bobby, then destroys the last bill.

- Chris will always follow one of these two strategies:
  - 1) Give out the smallest remaining bills (first to Bobby and then Amy, in that order), and then destroy the last bill.
  - 2) Give out the largest remaining bills (first to Bobby and then Amy, in that order), and then destroy the last bill.
- If Bobby gave Chris the smallest bill in the box (in Step 2), there is a 90% chance that Chris will distribute the smallest remaining bills (strategy 1), and a 10% chance he will distribute the largest remaining bills (strategy 2).
- If Bobby did not give Chris the smallest bill in the box (in Step 2), there is a 40% chance that Chris will distribute the smallest remaining bills (strategy 1), and a 60% chance he will distribute the largest remaining bills (strategy 2).

Use your knowledge of the expectimax algorithm to answer the questions below with the assumption that Amy and Bobby always play optimally (while Chris always plays one of his two strategies). If Bobby determines that multiple actions are equally optimal for himself, he chooses the one that also maximally benefits Chris. You should build a game tree to solve the following questions.

**1.B.1** How many nodes does the game tree have (including root and leaf nodes)? **(1 point)**

50

**1.B.2** Which box does Amy choose? **(1 points)**

- ○ Box A
- ○ Box B
- ○ Box C
- ⊙ Box D

**1.B.3** What is Amy's expected value? **(2 points)**

$72

---

**1.B.4** What is Bobby's expected value? **(2 points)**

$80

---

**1.B.5** What is Chris's expected value? **(2 points)**

$80

---

Amy has learned it is illegal in their county to destroy bills with a value greater than or equal to $50 and doing so would result in all 3 players paying a $50 fine. Suppose that only Amy knows this, while Bobby and Chris continue to play the game the same way.

**1.B.6** How many leaf nodes in the complete game tree would lead to all of them paying a fine? **(1 points)**

19

---

**1.B.7** Which box should Amy choose at the beginning in order to get the most money? **(1 points)**

- ○ Box A
- ⊙ Box B
- ○ Box C
- ○ Box D

# 2. Search

**(20 points)**

## Question A

**(5 points)**

Consider a simple two-link robotic arm. It has two joints. Each joint is like a hinge and allows the links to rotate in the plane. The figure below shows the functioning of the joints.

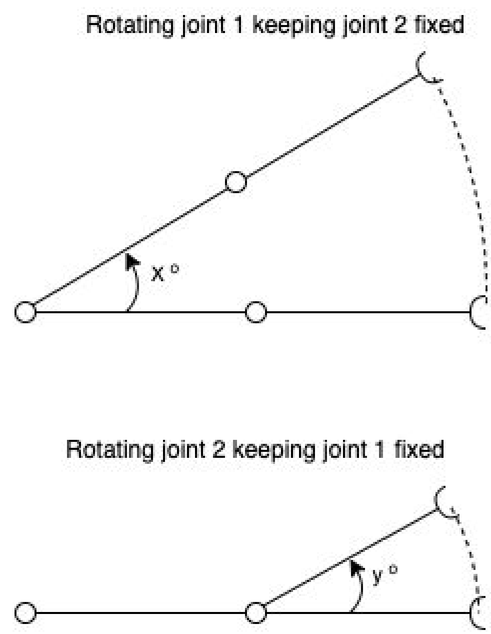



**Figure 2.1:** Functioning of the two joints

Assume that the length of each link is 10 units and let (x, y) denote the configuration of the arm where x is the angle of the first joint with respect to the horizontal, and y is the angle of the second joint with respect to the first link. The diagram below shows this state.

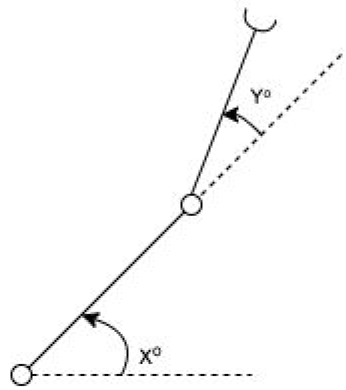



**Figure 2.2:** Rotational angles of the two joints

The first joint can have the configurations {-120, -60, 0, 60, 120, 180} with respect to the horizontal axis and the second joint can have the configurations {-60, 0, 60} with respect to the first link. To move the robot from one configuration to another, we can rotate the joints by $60^o$ in either clockwise or anticlockwise direction. For example, starting from the configuration of (0,0), we can move to the configurations (60, 0), (-60, 0), (0, 60), (0, -60) in one step.

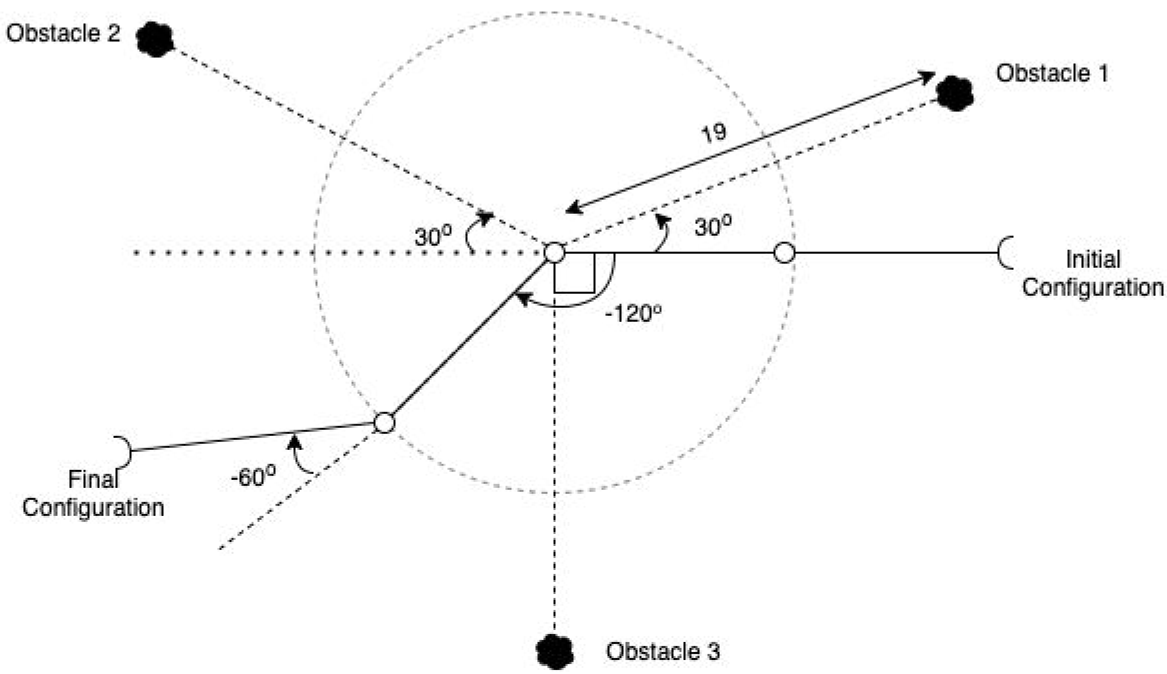



**Figure 2.3:** Challenge plane with obstacles

However, there are some obstacles in the plane which hinder the free movement of the joints. For example, due to obstacle 1, we cannot directly move the arm from the state (0,0) to the state (60,0). One possible way to do that is to rotate the second joint first, then rotate the first joint by $60^o$ : (0,0) → (0, 60) → (60,60) → (60,0). Each obstacle is at a distance of 19 units from the center and their positions are shown in the figure above.

Let the initial configuration of the arm be (0,0). We want to move the arm to the configuration of (-120, 60). Conduct a **breadth-first search** to find out the best path to reach the final configuration. Please follow the following guidelines while adding nodes to the frontier:

1. Add the configuration corresponding to the rotation of the first joint before the second joint.
2. Add the configuration with the clockwise rotation first before the anticlockwise rotation for each joint.
3. If a configuration isn't already explored, you should ensure that only one instance of it should exist in the frontier and it should be the one with the smallest priority.

For example, while exploring the configuration (0,0), we add the configurations (-60,0), (0,-60), (0, 60) to the frontier in that order.

**2.A.1** Which of the following configurations lie on the best path found out by BFS? Mark all that apply. **(2 points)**

- [x] (0, 60)
- [ ] (-60,-60)
- [ ] (-60, 0)
- [x] (-60, 60)

**2.A.2** What is the number of nodes explored in the previous part? A node is explored when it is removed from the frontier. **(2 points)**

8

**2.A.3** What is the length of the best path found out by BFS? **(1 point)**

3

## Question B

**(15 points)**

It is 2025, and Georgia Tech has opened a new campus on Mars. Figure 1 shows a simplified version of the map. The white squares are traversable, and the black squares represent the walls that you cannot go through. When you are at a square, you can only move to one of your 4-connected neighbors (UP, DOWN, LEFT, RIGHT). Travelling from a square to its neighbor takes one unit of time. A student at this campus travels every day from her dorm at S to the Crater Commons building at G to attend her classes. After taking the AI class, she decides to find the path that takes the minimum time to get from S to G.

**Rules for searching:**

- Process the neighbors of a square in the order: RIGHT, DOWN, LEFT and UP. i.e. add the traversable neighboring squares to the frontier in the above order.
- If there is a tie when removing squares from the frontier, follow the FIFO ordering.
- A square is considered **explored** when it is popped from the frontier.

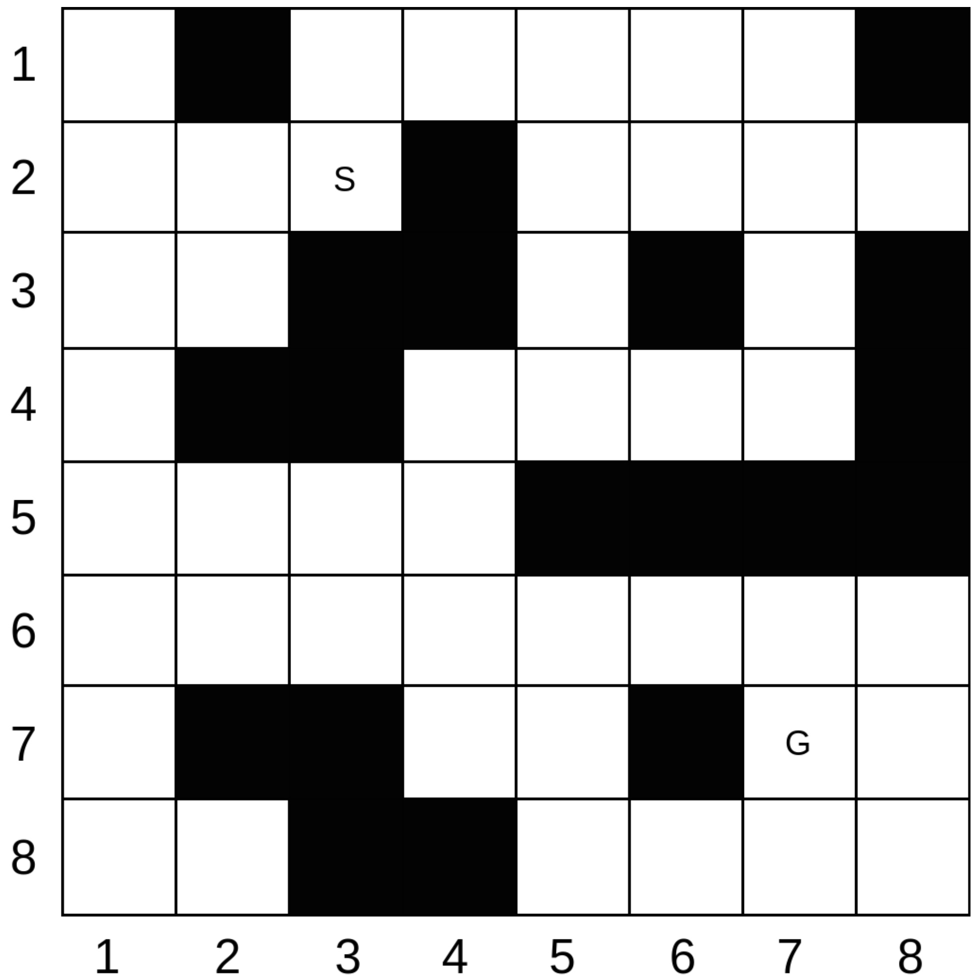



**Figure 2.4:** Georgia Tech campus on Mars

**2.B.1** Help the student find the path which takes the minimum time to travel from S to G by performing Uniform Cost Search. **(4 points)**

a. What is the total time taken along the optimal path?

13

b. How many squares are explored (including S and G)?

42

**2.B.2** Which of the following is true with regards to UCS and BFS for the above question? Mark all that apply. **(2 points)**

- ☐ The path found by UCS and BFS is the same.
- ☑ The path found by UCS and BFS is different.
- ☐ The number of squares explored by UCS and BFS is the same.
- ☑ The number of squares explored by UCS and BFS is different.

**2.B.3** It turns out that although students cannot walk through the black squares, a robot built by a group of AI students can, as it has a laser gun mounted on top of the robot that shoots lasers to blast through the obstacles. The laser gun can only fire vertically or horizontally on the map, and every time it fires, it costs 3 units of energy per black square it turns into a walkable square. Depending on the amount of energy the robot consumes, it can clear any number of black squares on a vertical or horizontal line. However, by decree of the Martian government, there is a limit on the number of energy cells that a robot can carry at any time, so that it can fire at most twice per trip. The robot itself consumes 1 unit of energy per square it moves across.
Help the student to find the path which takes the minimum energy to travel from S to G by performing A* search. You should use the Manhattan distance heuristic. **(4 points)**

a. What is the total energy consumption along the path found by the A* search?

14

b. What are the coordinates(x,y) of the first black square that is blasted in the optimal path?

(2,4)

**2.B.4** Let the only two Manhattan walks from a node x to the goal node G be $P_1$ and $P_2$. Let the number of white and black squares on $P_1$ and $P_2$ be $w_1$, $w_2$, $b_1$, $b_2$ respectively. Which of the following heuristics are admissible? Which ones are consistent? Please list them (use "(a)","(b)","(c)" ordered alphabetically) in the blank spaces below or "None" if none of them are admissible/consistent. **(2 points)**

(a) $h1(x) = \min(4\, b_1 + w_1, 4\, b_2 + w_2)$
(b) $h2(x) = \min(4\, b_1 + 4\, w_1, 4\, b_2 + 4\, w_2)$
(c) $h3(x) = \min(b_1 + w_1, b_2 + w_2)$

Admissible heuristics:

(c)

---

Consistent heuristics:

(c)

---

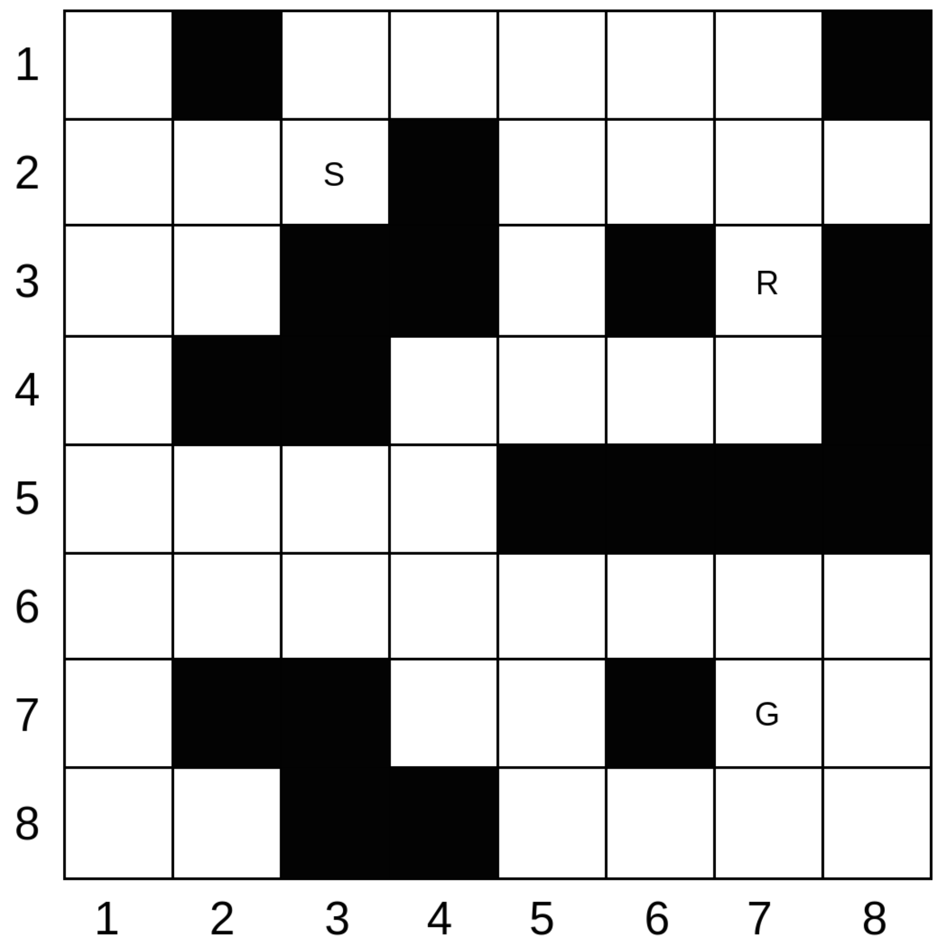



**Figure 2.5**: Campus map with charging station at R

**2.B.5** The university has just installed a robot charging station on campus at R shown in the map above, and the robot gets back 6 units of energy per recharge, thereby reducing the total energy consumption. For this question, consider a modified UCS algorithm that can visit a square that has been visited before. Note that the optimal solution is the one having the optimal energy consumption. Which of the following are true? **(3 points)**

- [ ] Modified UCS algorithm returns the optimal solution for the robot to travel from S to G.
- [x] Modified UCS algorithm returns the optimal solution if the robot uses the charging station at most twice.
- [ ] Depending on the location of the charging station, the modified UCS algorithm could return either the optimal or suboptimal solution, assuming also that the robot uses the charging station at most twice.
- [ ] Modified UCS algorithm returns the optimal solution if the robot gets back only 1 unit of energy per recharge instead of 6.
- [ ] Modified UCS algorithm returns the optimal solution if the robot gets back only 3 units of energy per recharge instead of 6.

# 3. Genetic Algorithms

**(16 points)**

**Note: No partial credit will be allowed for this question. Please note that a partial solution has been provided to you in Q4, such that it can be used to verify your computations.**

Genetic algorithms are a unique class of optimization processes popularized by John Holland, a prominent computer scientist who modeled optimization algorithms using the naturally occurring processes of natural selection as a guide. As the name suggests, genetic algorithms utilize the mechanisms found in genetic evolution and apply them to various problem spaces to 'evolve' solutions into optimal solutions.

When conducting search, genetic algorithms process a population of members (potential solutions), which represents the search space solution, using three operations — selection, crossover, and mutation. The population of individuals goes through a sequence of unary (mutation) and higher-order (crossover) transformations. These individuals strive for survival; a selection scheme, biased toward fitter individuals thus selecting the surviving generations. After some number of iterations, the program converges and the best individual hopefully represents an optimal solution.

Genetic algorithms have had significant success in many modern domains such as encryption, power flow optimization, music composition, design of Machine Learning models, and many more. In this section, you'll apply genetic algorithms to the problem of automatically designing Machine Learning models, more formally known as Neural Architecture Search.

## Neural Architecture Search

To have a thorough understanding of this application of Genetic Algorithms, it would help to have a cursory understanding of this domain. Machine Learning is a subfield of Artificial Intelligence wherein we study and design algorithms that improve automatically through experience. ML experts model these algorithms to learn complex functions for solving real-world applications. However, with the rapid growth of machine learning applications there is a demand for off-the-shelf ML models that can be used easily and without expert knowledge. Therefore AutoML emerged as a new field of research to make Machine Learning available for non-Machine Learning experts. Neural Architecture Search (NAS) is a popular technique of AutoML that is used to automate the design of Machine Learning models. Several search strategies such as random search, Bayesian optimization, evolutionary methods, reinforcement learning, and gradient-based methods are utilized to carry out NAS. However, in this section, our focus will primarily be on understanding how genetic algorithms are applied to NAS.

## Problem Formulation

In Machine Learning, classification is a type of problem wherein data points are assigned to predefined classes / categories by the learnt model. Your task is to find the best performing Machine Learning model that can classify images of ten different dog breeds from the ImageWoof dataset. A model is defined by a tuple of four attributes, each of which is assigned a value from the attribute's domain:

| **Attributes** | **Domain** |
|---|---|
| Depth | {2,3,4} |
| Width | {3,4,6} |
| Kernel Size | {3,5,7} |
| Image Resolution | {160, 196, 224} |

A Machine Learning model 'M' of the population is encoded by the 4 characteristics list above in this order: **{depth, width, kernel size, image resolution}**. For instance a model with depth=4, width=3, kernel size=5 and image resolution=224 would be represented as {4, 3, 5, 224}.

## Fitness Function

In the genetic algorithms population, there are some individuals who are 'fitter' than others. To quantify this fitness characteristic, genetic algorithms use what is called a **fitness function** to compare the members of the population. Moreover, machine learning models are commonly quantized by a performance metric known as accuracy where higher accuracy implies better performance. Therefore, for this particular problem we will estimate the accuracy of a model (M) with the function given here:

$$Fitness(M) = depth^{2.71} + width^{1.8} + \left(\frac{image\ resolution}{kernel\ size}\right)^{0.75}$$

**3.1** To generate new ML model architectures, we need an initial population of models to transform and mutate. Given below is a sub-sample of this population. Your first task is to calculate the accuracy of these models based on the fitness function provided. Report your **final answer** according to the rounding rules of the exam. **(2 points)**

| Initial Population | | |
| --- | --- | --- |
| **Model** | **Encoding** | **Fitness Score** |
| **Parent #1** | {2, 4, 7, 160} | |
| **Parent #2** | {4, 6, 5, 224} | |
| **Parent #3** | {3, 3, 3, 196} | |
| **Parent #4** | {2, 6, 5, 224} | |

## Crossover

In order for a genetic algorithm to generate offspring from the existing population, a procedure must be defined to 'mate' members together to form new ones. We use an operation called crossover and define it between two models as follows:

1. Separate each model encoding into halves. These encodings have an even number of attributes, so the "first half" will contain the depth and width attributes and the "second half" will contain the kernel size and image resolution attributes.
2. Overlay the first half of one encoding over the second half of the other to create the first child, and the same for the remaining halves to create the second child.

An example of such a crossover is shown below:

```
   Parent 1              Parent 2
{3,4,5,160}     X     {4,3,7,224}
```

1. Separate the halves of Parent 1:

```
    {3,4}                {5,160}
```

2. Separate the halves of Parent 2:

```
    {4,3}                {7,224}
```

3. Overlay the corresponding halves of Parent 2 to create two children:

```
   Child 1               Child 2
{3,4,7,224}           {4,3,5,160}
```

**3.2** Using the initial population shown in Question 1, perform the crossover procedure defined above on the pair of models that have the best fitness (highest and second-highest) and on the other pair of models (lowest and second-lowest). Fill in the below table in **decreasing order of the fitness scores**.. If the fitness scores of any of the offspring are the same, then order them in ascending order by sorting their encodings from left to right. For example, if Child 1 and Child 2 have ties with encodings {3,4,7,22} and {4,3,5,160} then Child 1 is followed by Child 2 in the table below. Report your **final answer** according to the rounding rules of the exam. **(4 points)**

| **Crossover Children** | | |
|---|---|---|
| **Model** | **Encoding** | **Fitness Score** |
| **Child #1** | | |
| **Child #2** | | |
| **Child #3** | | |
| **Child #4** | | |

## Mutation

Genetic algorithms rely on mutation procedures to introduce some diversity in the population and to continually produce different solutions. In order to introduce new characteristics into a population, the crossover operation is often not enough. Thus, we need to 'mutate' children in a small but potentially impactful manner to traverse our search space efficiently. This method allows children to have characteristics that their parents do not have, and in this way the population as a whole avoids homogeneity.

The mutation process for this problem is defined in the next few paragraphs. We mutate one attribute per child on any iteration of the procedure.

We first identify the attribute to mutate. To do so, index each attribute with an integer, like so: {1:'depth', 2:'width, 3:'kernel size', 4:'image resolution'}. Let the integer denoting the indexed attribute be 'd' and call it an **identified** attribute.

Next, in order to replace this value of the attribute with the new 'mutated' value, we must identify a domain value to replace our old value with. To do so, index the domain values (list in the first table) in the same way we indexed the attributes and randomly sample an integer corresponding to the new domain value. Let 's' denote the indexed domain value and call it an **identified** domain. An **identified** domain will always correspond to an **identified** attribute.

We resample the attributes and corresponding domains to mutate for each child. We provide an example of this procedure below for Child 1.

First obtain the attribute to mutate:
*'d' = randInt(1,4) for Child 1 → generates 2*
*Hence 'd' =2*

To identify the value to replace the attribute with:
*'s' = randInt(1,3) for Child 1 → generates 2*
*Hence 's' =2*

Since d = 2 was returned, the second attribute i.e. width is mutated. Since s = 2, it is replaced with the second value from the corresponding domain of width, which has the values of {3,4,6} associated with it. Hence, if `Child 1 was {4, 3, 5, 224}` initially, its encoding will be changed to `{4,4,5,224}`

**3.3** Mutate each of the above children obtained from crossover and fill in the resultant models below. Use the identified attributes d = {2,3,1,4} with corresponding identified domains s = {1,2,3,2} to mutate each of the four children from Question 2, where the first value of d and s corresponds to Child 1, the second value to Child 2 and so on.
After mutation, **order the models again by the criteria described in Question 3.2**. Report your **final answer** according to the rounding rules of the exam. **(4 points)**

| **Mutation Children** | | |
|---|---|---|
| **Model** | **Encoding** | **Fitness Score** |
| **Child #5** | | |
| **Child #6** | | |
| **Child #7** | | |
| **Child #8** | | |

## Competition

Congratulations! You have now completed the first generation of NAS. In order to carry forth the next generation, a genetic algorithm reduces the population size back to its initial numbers (N). A common strategy that is employed in this case is global competition amongst all members of the population whereby we keep the top N models and discard the remaining members.

**3.4** At the end of the first generation we have a total of 12 members in the population (4 initial + 4 crossover children + 4 mutation children). Carry out global competition and determine the models that will move on to the second generation of NAS. Fill in the below table in **decreasing order of the fitness function**. As noted at the beginning of the question a partial solution has been provided. **(3 points)**

| **Final Population** | | |
|---|---|---|
| **ML Model** | **Encoding** | **Fitness Score** |
| Model #1 | | |
| Model #2 | {4,6,5,224} | |
| Model #3 | | 73.018413 |
| Model #4 | | |

**3.5** You have now seen how a Genetic Algorithm tries to solve the NAS problem by traversing the search space in order to find good solutions. If we increase the search space and have infinitely more domain values available to the corresponding attributes then which of the following properties make(s) genetic algorithms a good solution? One or more options may be correct. **(2 points)**

- ☐ The search space has a high degree of stochasticity.
- ☐ The task at hand considers each individual in the search space to be derived as a permutation of 2 or more parents.
- ☐ There are many possible optimal solutions for this problem.
- ☐ There is only one global optimum for this search space.
- ☐ There are very few local minima.

**3.6** Which of the following statements related to optimization algorithms is/are true? **(1 point)**

- ☐ A local beam search with k states is equivalent to performing k random restarts in parallel.
- ☐ Genetic algorithms with local beam search will reach an optimal solution
- ☐ Genetic algorithms perform a global search whereas vanilla hill climbing performs a local search.

# 4. Constraint Satisfaction Problems

**(14 points)**

## Question A

**(9 points)**

Georgia Burdell is a second year CS PhD student at Georgia Tech. She just took the qualifier exam and is planning for her course work. There are ten classes she is considering. She needs to take **six courses** in **three semesters** to satisfy the following program requirements:

1. Course A is a prerequisite of course B and course C
2. Course D and E can not be taken in the same semester, the same applies for A and F as well as B and H
3. Only F and G are only offered in semester 2
4. G is only offered semester 2
5. She must take at least one course from course E and F
6. She must take at least one course from course G and H
7. She can take at most one course from course I and J
8. Course B is required by her qualifier committee
9. In semester 1 and 3, she needs to take at least two courses
10. In semester 2, she needs to take at least one course

She can choose whether to take a course and which semester to take it in. Now she decides to use her AI knowledge to make the plan.

**4.A.1** Check all courses that have **unary constraints. (1 point)**

- ☐ Course A
- ☐ Course C
- ☐ Course F
- ☐ Course J

**4.A.2** She decides to set courses as variables and set {0-not take, 1-take in semester 1, 2-take in semester 2, 3-take in semester 3} as possible domain values. What are the domains for variables B, D, G after applying only the unary constraints? Please report the domain in increasing order of domain values i.e. {0,2,3} instead of {0,3,2}. **(1 point)**

| Variable | Domain |
|---|---|
| B | |
| D | |
| G | |

**4.A.3** What are the domains for variables B, D, G after applying both the **unary and binary constraints**? Please report the domain in increasing order of domain values i.e. {0,2,3} instead of {0,3,2}. **(1 point)**

| Variable | Domain |
|---|---|
| B | |
| D | |
| G | |

**4.A.4** When constructing the constraint graph for this problem, show which variables are connected to each other by checking all the pairs of courses that have **binary constraints**. **(1 point)**

- ☐ A and B
- ☐ A and C
- ☐ D and F
- ☐ F and G
- ☐ B and H
- ☐ C and H
- ☐ E and F
- ☐ I and J
- ☐ B and J

She has named these courses based on how much she wants to take them (A being the most desired course, followed by B and as so on until J, which is the course she prefers the least). Also, she would like to take the course she wants the most as early as possible. She tries to use backtracking to solve the problem.

**4.A.5** Which constraint will not cause her to backtrack? She doesn't apply forwarding checking for domain consistency and arc consistency. Mark all that apply. **(1 point)**

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 5

**4.A.6** To save time in searching for the results, she applies **domain consistency** and **arc consistency** at one step when she is trying to assign the next value. **(2 points)**
Semester 1: A, D
Semester 2:
Semester 3: B, C

**a.** What are the domains for E, F, G? Please report the domain in increasing order of domain values i.e. {0,2,3} instead of {0,3,2}.

| Variable | Domain |
|---|---|
| E | |
| F | |
| G | |

**b.** Which course and which value will she assign next? Please answer this question with the format of Course = Semester, i.e. A = 1.

______________________________________________

**4.A.7** What is the solution you get? Please list courses in each semester in alphabetical order i.e. {A, B, C} instead of {B, A, C}. **(2 points)**

| Semester | Courses |
|---|---|
| Semester 1 | |
| Semester 2 | |
| Semester 3 | |

## Question B

**(5 points)**

She recently developed research interest in Area I, so Course I becomes the top priority. This is her current planning of the course work:
Semester 1: I, A, D, E
Semester 2:
Semester 3: B, C

**4.B.1** How many constraints does this plan violate? **(1 point)**

- ○ 1
- ○ 2
- ○ 3
- ○ 4

During her study of AI, she learned an advanced version of backtracking, called backjumping. In backjumping, instead of going back to the variable assigned one step ago, the algorithm may jump to a variable assigned several steps ago.
Consider the scenario when $x_1$, $x_2$, …, $x_k$ are assigned values and none of the values in $x_{k+1}$'s domain are feasible. If you find out when the current values of $x_1$, $x_2$, …, $x_j$ ($j < k$) cannot be extended to form a solution with whatever value for $x_{k+1}$, you can directly consider a different value for $x_j$ in which j can be any value from 0 to k instead of reconsidering $x_k$ as backtracking would normally do.

**4.B.2** If applying **backtracking**, which variable should she reassign value to? **(1 point)**

________________________________________

**4.B.3** If applying **backjumping**, which variable should she backjump to? **(1 point)**

________________________________________

**4.B.4** What is the new solution? Please list courses in each semester in alphabetical order i.e. {A, B, C} instead of {B, A, C}. **(2 points)**

| Semester | Courses |
|---|---|
| Semester 1 | |
| Semester 2 | |
| Semester 3 | |

# 5. Probability

**(20 points)**

## Question A

**(12 points)**

**5.A.1** Which of the following expressions are equal to 1? Mark all that apply. **(2 points)**

- ☐ $\sum_b P(A|B=b)$
- ☐ $\sum_a P(A=a|B)$
- ☐ $\sum_a \sum_b P(A=a, B=b)$
- ☐ $\sum_a \sum_b P(A=a|B=b)$
- ☐ $\sum_a \sum_b P(A=a)P(B=b)$
- ☐ $\sum_a P(A=a)P(B=b)$

**5.A.2** The probability $P(A|B,C)$ is equal to which of the following expressions? Mark all that apply. **(2 points)**

- ☐ $\frac{P(B|A,C)P(A|C)}{P(B,C)}$
- ☐ $\frac{P(B|A,C)P(A|C)}{P(B|C)}$
- ☐ $\frac{P(B|A,C)P(C|A,B)}{P(B,C)}$
- ☐ $\frac{P(A,B|C)}{P(B|A,C)}$
- ☐ $\frac{P(B,C|A)P(A)}{P(B,C)}$
- ☐ $\frac{P(A,B,C)}{\sum_a P(A=a,B,C)}$

**5.A.3** Which of the following expressions are equal to $P(A,B,C)$ given that $A \perp B|C$ and $A \perp C$? Mark all that apply. **(3 points)**

- ☐ $P(A)\,P(B,C)$
- ☐ $P(A)P(B)P(C)$
- ☐ $P(A)P(B|C)P(C)$
- ☐ $P(A \mid B,\ C)\,P(B \mid A,\ C)\,P(C \mid A,\ B)$
- ☐ $P(A \mid C)\,P(B \mid C)$
- ☐ $P(A \mid B)\,P(B \mid C)P(C)$
- ☐ $P(A \mid C)\,P(B \mid C)P(C)$

**5.A.4** Given below is the joint distribution of $P(A,B,C)$.

| $A$ | $B$ | $C$ | $P(A,B,C)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0.01 |
| 0 | 0 | 1 | 0.02 |
| 0 | 1 | 0 | 0.03 |
| 0 | 1 | 1 | x |
| 1 | 0 | 0 | 0.01 |
| 1 | 0 | 1 | 0.1 |
| 1 | 1 | 0 | y |
| 1 | 1 | 1 | z |

Suppose $B \perp C|A$. What are the missing entries (x, y, z)? **(2 points)**

- ○ 0.06, 0.07, 0.7
- ○ 0.7, 0.07, 0.06
- ○ 0.07, 0.06, 0.7
- ○ 0.07, 0.07, 0.06

**5.A.5** You take a test to see if you have contracted a rare disease. The test is 99% accurate (in either case when you have the disease or not) and the disease occurs in 0.1% of the population. You test positive, what's the probability you have contracted the disease? Report your **answer** according to the rounding rules of the exam. **(1.5 points)**

______________________________________________

**5.A.6** After having tested positive once, you take the test again for the second time to improve your certainty. Surprisingly and unfortunately, you test positive again. What is the probability now that you have contracted the disease? **Consider the second test to be independent of the first test.** Report your **answer** according to the rounding rules of the exam. **(1.5 points)**

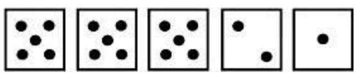
______________________________________________

## Question B (Yahtzee)

**(8 points)**

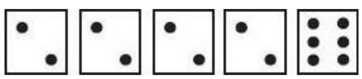
Yahtzee is a dice-based game similar to how Poker is played. The objective of the game is to roll certain combinations of numbers with five dice. At each turn, you roll the dice and attempt to get one of the combinations that will give you a higher score. Each turn in the game consists of at most 3 rounds of rolling the dice. The first round involves rolling all the five dice. Thereafter in the second and third rounds, you can hold out any number of the dice (including none or all of them) and roll the remaining dice. Once a dice is held out you cannot roll it in any of the subsequent rounds in your current turn. Your final combination is what you have at the end of your turn. An important note is that you can use a combination only once per game, so you have to be careful when you add it to your score.

- **3 of a kind**: You roll dice that show at least 3 of the same number. For example:

  **The score is the total of the dice. (In this example 18)**

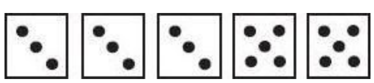
- **4 of a kind:** You roll dice that show at least 4 of the same number. For example:

  **The score is the total of the dice. (In this example 14)**

- **Full House**

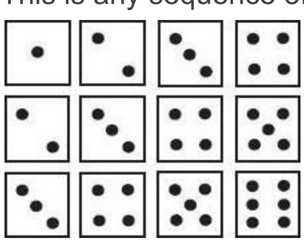
  You need to roll both a "3 of a kind" and a pair.

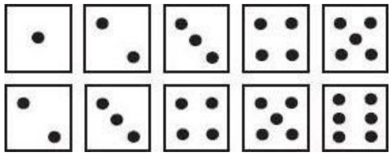
  **A "Full House" scores 25 points.**

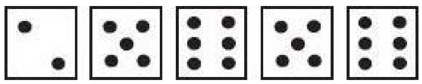
- **Small Straight:**

  This is any sequence of four numbers, such as:

  The 5th dice can be any number. **Any "Small Straight" is valued at 30 points.**

- **Large Straight**
  This is a sequence of five numbers, either:
  
  **A "Large Straight" is valued at 40 points.**

- **Chance**: The player rolls any combination of dice and is just taking the total sum of the dice.
  
  **This combination scores 24 points.**

- **Yahtzee:** This is any "five of a kind". **Yahtzee scores 50 points.**

**Example Turn:**

- Round 1: You roll (1, 4, 1, 5,  6). Currently, you have no high scoring combinations other than just taking the total of all the dice. You are aiming for a 4 of a kind so you set aside 1's and re-roll the dice (4, 5, 6)
- Round 2: Your roll with the 3 dice gives you (1, 2, 5) and your current complete set of dice is (1, 1, 1, 2, 5). You have a 3 of a kind but you will try once more for the four of a kind. You set aside the first 3 dice and reroll the remaining.
- Round 3: Your last roll gives you (6, 6) and your current complete set of dice is (1, 1, 1, 6, 6). While your final combination did not give a three of a kind, you did get a full house. You can choose whether to use the full house or a 3 of a kind. In this case, the 3 of a kind would only give you 15 points while the full house will give you 25 points. So you would ideally pick the full house for this turn.

**Expected Score Calculation:** Probability x Score

**Combination formula:** $C_r^{\,n} = \frac{n!}{(n-r)!r!}$

**5.B.1** If you roll a single dice 3 times, what is the probability of getting at least two 5s? **(1 point)**

- ○ 0.062500
- ○ 0.096291
- ○ 0.074074
- ○ 0.585585

**5.B.2** You have rolled a 3, 4, 5, 2, 3. You pick one of the 3s and roll it at most twice more. What is the probability of getting a “large straight”? Report your **answer** according to the rounding rules of the exam. **(1.5 points)**

______________________________________________________________

**5.B.3** Suppose you are unable to get any good combinations and are just aiming for the chance score, which is the total of all your dice. Let’s say your strategy is to keep all dice that are greater than 3 i.e. 4, 5, 6 and re-roll all the other dice. What is your expected score after two **rounds**? Report your **answer** according to the rounding rules of the exam. **(2 points)**

______________________________________________________________

**5.B.4** You have rolled a 6, 4, 2, 1, 3. You want to maximize your expected score by aiming for 4 of a kind or a Yahtzee. You re-roll all the dice that are not 6 for at most two more rounds. Answer the below questions to find out which combination gives you the highest expected score. Note that a Yahtzee combination does not count as four of a kind. For example, (6, 6, 6, 6, 6) is a Yahtzee and does not count for 4 of a kind. **(3.5 points)**

a. Which of the following combinations will give you the highest expected score?

- ○ 4 of a kind with 6’s
- ○ Yahtzee with 6’s

b. What is the expected score corresponding to the combination selected for the above part?

______________________________________________________________

# 6. Bayes Nets

**(15 points)**

## Question A (D-Separation)

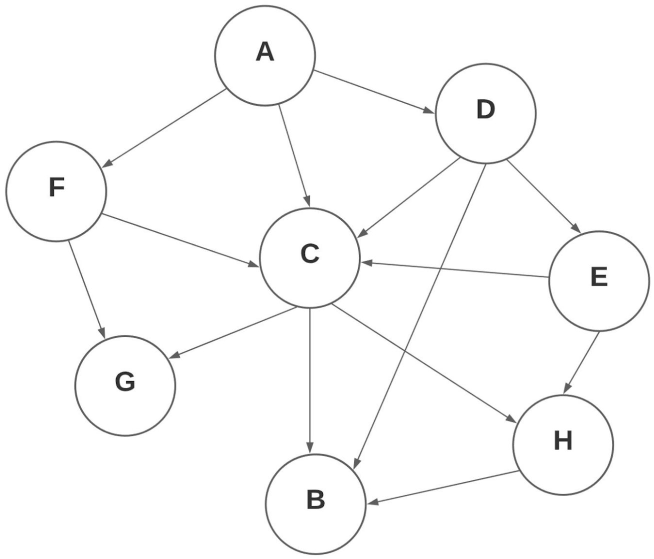
**(3.5 points)**

A Bayesian network is constructed over eight variables. However, a naughty child came and added some extra connections between the nodes on the graph, and now the graph you see below no longer represents the original Bayesian Network.

Luckily, we remember some relevant properties of the original, unaltered network. Please help us to determine the original Bayesian network by eliminating the unnecessary or incorrect dependencies. Use the properties provided below to determine which connections still apply to the original network.

Note: The original network is a single connected network, i.e., the nodes do not form multiple disjoint networks.

Relevant properties about the original network:
1) Each node has at least one parent or child node.
2) D is conditionally independent of B given C.
3) The Markov blanket of E is D and H.
4) The Markov blanket of F is G and C.
5) Node C has two parents

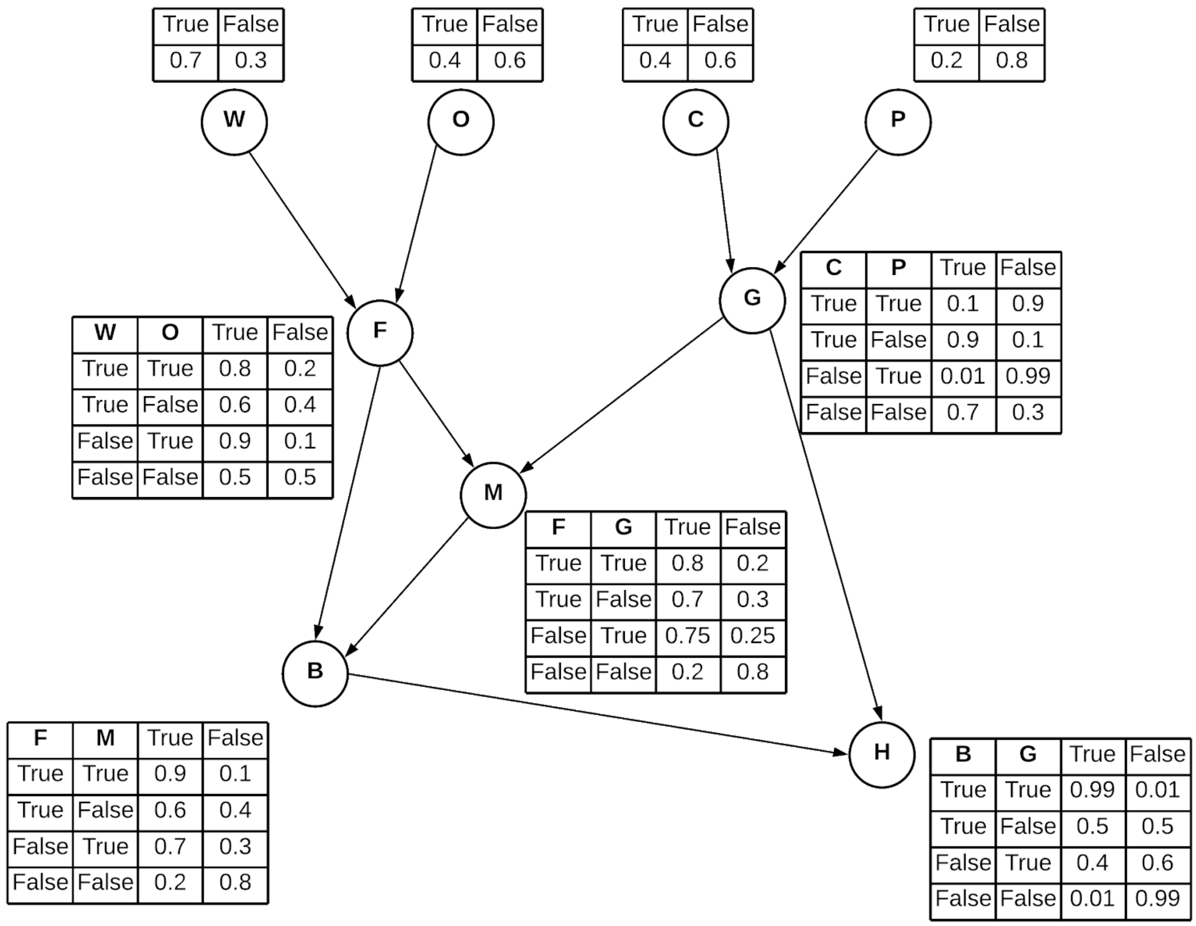



**6.A** Check only the connections that **are necessary** in the original Bayes network for all of the above properties to be true. **(3.5 points)**

- ☐ A → D
- ☐ A → F
- ☐ A → C
- ☐ F → C
- ☐ F → G
- ☐ D → C
- ☐ D → B
- ☐ D → E
- ☐ C → G
- ☐ C → B
- ☐ C → H
- ☐ E → C
- ☐ E → H
- ☐ H → B

## Question B (Inference)

**(11.5 points)**




P(W):

| True | False |
|---|---|
| 0.7 | 0.3 |

P(O):

| True | False |
|---|---|
| 0.4 | 0.6 |

P(C):

| True | False |
|---|---|
| 0.4 | 0.6 |

P(P):

| True | False |
|---|---|
| 0.2 | 0.8 |

| **W** | **O** | True | False |
|---|---|---|---|
| True | True | 0.8 | 0.2 |
| True | False | 0.6 | 0.4 |
| False | True | 0.9 | 0.1 |
| False | False | 0.5 | 0.5 |

| **C** | **P** | True | False |
|---|---|---|---|
| True | True | 0.1 | 0.9 |
| True | False | 0.9 | 0.1 |
| False | True | 0.01 | 0.99 |
| False | False | 0.7 | 0.3 |

| **F** | **G** | True | False |
|---|---|---|---|
| True | True | 0.8 | 0.2 |
| True | False | 0.7 | 0.3 |
| False | True | 0.75 | 0.25 |
| False | False | 0.2 | 0.8 |

| **F** | **M** | True | False |
|---|---|---|---|
| True | True | 0.9 | 0.1 |
| True | False | 0.6 | 0.4 |
| False | True | 0.7 | 0.3 |
| False | False | 0.2 | 0.8 |

| **B** | **G** | True | False |
|---|---|---|---|
| True | True | 0.99 | 0.01 |
| True | False | 0.5 | 0.5 |
| False | True | 0.4 | 0.6 |
| False | False | 0.01 | 0.99 |

After failing to win the Champions League title for the past five seasons, FC Barcelona seems to be in a world of trouble. This season, after restructuring the whole team, Barcelona made it to the Champions League final. Not knowing what else to do, Coach Koeman has enlisted your help in trying to turn things around during this game. In Figure 1, you will find a Bayes net that hopefully contains the key to a successful final. The variables (binary) that might affect the outcome of the game are:

- Weather (W): if the weather is nice and warm
- Opponent (O): if the opponent is one of Barcelona's rivals
- Fans (F): if the fans are fired up
- Messi (M): if Lionel Messi (iconic soccer legend) is effective in the game
- Barcelona Wins (B): if Barcelona wins that game
- Good Health (G): if Messi is in good health
- Cryotherapy (C): if Messi did cryotherapy (ice tub therapy) before the match
- Party (P): if Messi parties the night before the game
- Happy (H): if Messi is happy

Coach Koeman would like to know the answers to the following questions. Please do not round off the *intermediate* probability values. For the following questions, we will require you to ***show your work*** to demonstrate how you computed the various probabilities.

**6.B.1** For the given network, mark the statement(s) as either **True or False**: **(4.5 points)**

a. $P(C, M|H) = P(C|H) \cdot P(M|H)$
   - ○ True
   - ○ False

b. $P(H|G, C) = P(H|G)$
   - ○ True
   - ○ False

c. Node H is in M's Markov Blanket.
   - ○ True
   - ○ False

Provide some ***brief*** reasoning to your answers in 6.B.1.

**6.B.2** How fired up are the fans given that Messi is in good health? **P(F=True|G=True)**
Report your **answer** according to the rounding rules of the exam. **(3.5 points)**

**6.B.3** What is the chance that Messi had partied the night before the game, given he was in good health during the match but not productive in the game and Barcelona won? **P(P=True|G=True, M=False, B=True)**
Report your **answer** according to the rounding rules of the exam. **(3.5 points)**

# 7. Bonus

**(6 points)**

## Question A (Game Playing, Search)

**(2 points)**

**Select T/F for the below statements. (0.2 x10 = 2 points)**

**7.A.1** The game, Tic-tac-toe, is fully-observable and is a zero-sum game. **(0.2 points)**

- True
- False

**7.A.2** In UCS (uniform cost search), for all graphs G(V, E) the path returned remains the same even if a constant positive value of x is added to the cost of each edge E. **(0.2 points)**

- True
- False

**7.A.3** Alpha-beta can be modified to an iterative-deepening search exploration of the game tree and still get the optimal answer. **(0.2 points)**

- True
- False

**7.A.4** In a 3-player game, a variant of minimax called max-N is used which allows for both deep and shallow pruning. **(0.2 points)**

- True
- False

**7.A.5** An admissible heuristic always underestimates the remaining distance (cost) to the goal. **(0.2 points)**

- True
- False

**7.A.6** Pre-Computing landmarks always makes the tri-directional A* search faster, that is it terminates in fewer steps. **(0.2 points)**

- True
- False

**7.A.7** For bi-directional search the stopping condition occurs at the intersection of the explored set of the current search with the explored set of the opposite search. **(0.2 points)**

- True
- False

**7.A.8** In bi-directional A*, both the final path and the intersecting/common node are not affected by the order in which search frontiers are expanded (e.g., 1. frontiers are expanded in alternating fashion 2. one frontier first is expanded completely and then second is expanded etc). **(0.2 points)**

- True
- False

**7.A.9** Pruning is not possible in non-deterministic games. **(0.2 points)**

- True
- False

**7.A.10** Breadth-first search is complete if the state space has infinite depth but finite branching factor. **(0.2 points)**

- True
- False

## Question B (Optimization, Constraint Satisfaction Problem)

**(2 points)**

**Hill Climbing with Restarts**

Consider a (10 x 10) grid where each square contains a real number that corresponds to its reward $w$. Assume you will initialize your hill climbing algorithm at a random location within the grid:

$$\{x\ :\ x \sim U(D^2)\} \text{ where } D\ =\ \{i\ :\ i \in N \wedge i\ < 11\}$$

i.e. the starting location is sampled uniformly over the grid. Assume we would like to calculate the average value of $\hat{w}$, the optimal reward after convergence of one run, as the number of hill climbing trials approaches infinity, i.e.,

$$z^*\ = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \hat{w}\ .$$

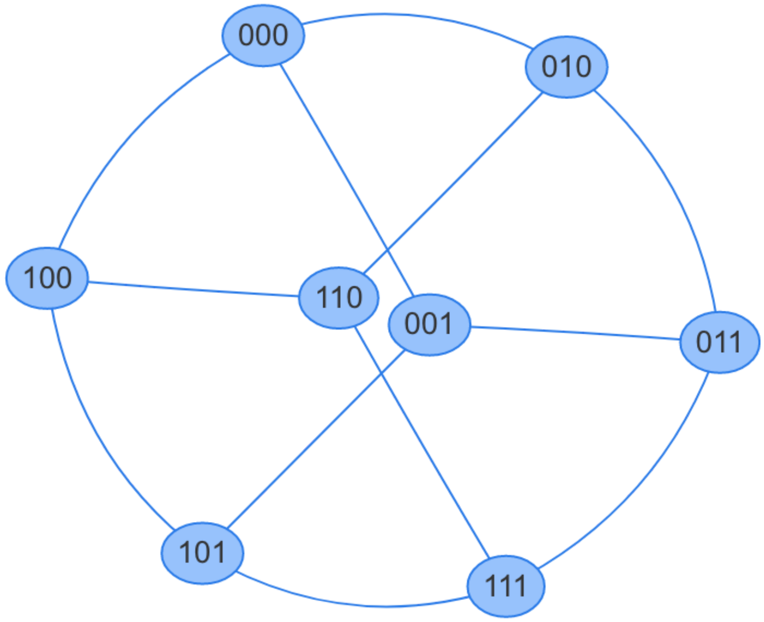
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| -0.8 | -0.1 | 0.5 | 1.1 | 1.8 | 2.4 | 3.0 | 3.6 | 4.3 | 4.9 |
| -1.3 | -0.6 | -0.0 | 0.6 | 1.3 | 1.9 | 2.5 | 3.1 | 3.8 | 4.4 |
| -1.9 | -1.3 | -0.6 | 0.0 | 0.6 | 1.3 | 1.9 | 2.5 | 3.1 | 3.8 |
| -2.4 | -1.8 | -1.1 | -0.5 | 0.1 | 0.8 | 1.4 | 2.0 | 2.6 | 3.3 |
| -2.6 | -1.9 | -1.3 | -0.7 | -0.1 | 0.6 | 1.2 | 1.8 | 2.5 | 3.1 |
| -2.4 | -1.8 | -1.1 | -0.5 | 0.1 | 0.8 | 1.4 | 2.0 | 2.6 | 3.3 |
| -1.9 | -1.3 | -0.6 | 0.0 | 0.6 | 1.3 | 1.9 | 2.5 | 3.1 | 3.8 |
| -1.3 | -0.6 | -0.0 | 0.6 | 1.3 | 1.9 | 2.5 | 3.1 | 3.8 | 4.4 |
| -0.8 | -0.1 | 0.5 | 1.1 | 1.8 | 2.4 | 3.0 | 3.6 | 4.3 | 4.9 |
| -0.6 | 0.1 | 0.7 | 1.3 | 1.9 | 2.6 | 3.2 | 3.8 | 4.5 | 5.1 |

**7.B.1** What is the value of $z^*$? **(1 point)**

- ○ 4.0
- ○ 4.9
- ○ 5.0
- ○ 5.1

**Constraint Satisfaction**

Recall the *chromatic number*: the minimum number of colors to color a graph G.




**7.B.2** Determine the *chromatic number* for the above graph. **(1 point)**

- ○ 2
- ○ 3
- ○ 4
- ○ 5

## Question C (Probability, Bayes Network)
**(2 points)**

300 students were picked at random for surveying the overall quality of two new labs - Schroedinger lab, and Einstein lab. The parameters based on which a lab should be evaluated are known to students. Now based on the experience of the students who took the labs, here are the results of the experience in tabular format:

| Lab | Satisfied | Dissatisfied | Total |
|---|---|---|---|
| Schroedinger | 95 | 50 | 145 |
| Einstein | 115 | 40 | 155 |
| Total | 210 | 90 | 300 |

**7.C.1** One student is randomly picked from these 300 students. Determine the probability that this student was dissatisfied given that the student evaluated the Einstein lab. Report your **answer** according to the rounding rules of the exam. **(0.25 points)**

__________________________________________

**7.C.2** Given that a random student who was dissatisfied by the lab they took, what is the probability that the student took the Schroedinger lab? Report your **answer** according to the rounding rules of the exam. **(0.25 points)**

__________________________________________

**7.C.3** Now, consider a different scenario. Suppose the university believes that students are biased while evaluating labs (i.e rather than evaluating on parameters they were given they have some preconceived biases while evaluating the lab). So the university decides to build a chatbot that can chat with students and figure out the biases and can catch all those students. It is estimated by the university that 50% of the students are biased. The bot claims it can detect 99% of biased students and the probability for a false positive (i.e identifying a non-biased student as biased) is 5%. If the student is marked as biased by the bot, what is the probability that the student is actually not biased? Report your **answer** according to the rounding rules of the exam. **(0.5 points)**

__________________________________________

**Bayes Net:**

Now consider a different scenario, where 3 students A, B, and C have to evaluate 4 different labs L1, L2, L3, and L4 labs. Each lab needs to be evaluated on different parameters denoted by the edges in the graph. Assume that all random variables are boolean, they are ‘True’ or ‘False’. Consider the following Bayesian network for this problem as below:

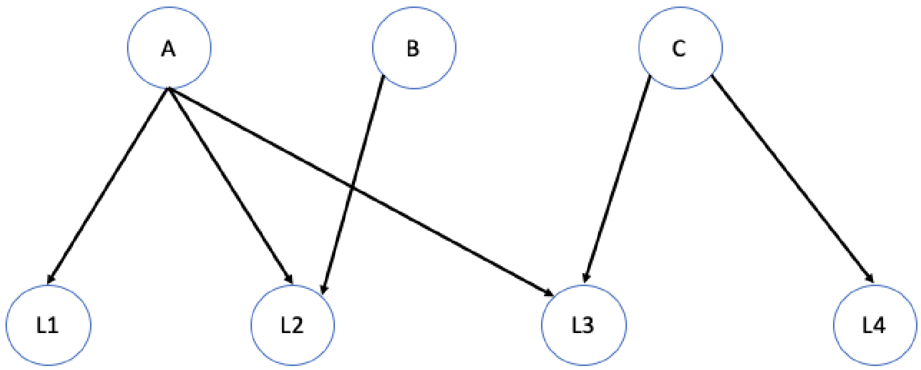



Figure: Note that L1, L2, L3, L4 are labs, and A, B, C are students, and their dependence on the lab is shown by edges between nodes.

**7.C.4** What is the total number of independent parameters that are required to describe the joint probability distribution, which you will get from the product of conditional probabilities? **(1 point)**

______________________________________________________________

## Checklist

And now mark the checklist below making sure you have taken care of each of the points mentioned:

- ☐ I have read the pinned Piazza post with the title 'Midterm Exam Clarifications Thread', and I am familiar with all of the clarifications made by the Teaching staff.
- ☐ All answers with more than 6 digits after the decimal point have been rounded to 6 decimal places.
- ☐ All pages are being uploaded in the correct order that they were presented to me, and none of the pages are missing/removed.
- ☐ Any extra pages **(including blanks)** are only attached at the END of this exam, after page 39 with clear pointers to wherever the actual answer is in the PDF (reference properly).
- ☐ I am submitting only one PDF and nothing else (no docx, doc, etc.).
- ☐ The PDF I am submitting is not blank (unless I want it to be).
- ☐ **I will go over the uploaded pictures on Gradescope and make sure that all the answers are clearly visible. I acknowledge that I am aware that dull / illegible / uneven scans will not be graded.**
- ☐ I have submitted a copy of the PDF to Canvas.