6.034 Final Examination
16 December 2015

Name: Han Solo **Email:**

Indicate which sections of the final you will be taking. We will grade **only** those sections.

☑ Quiz 1			☑ Quiz 2		☑ Quiz 3		☑ Quiz 4	
Problem 1	Problem 2	Problem 3	Problem 1	Problem 2	Problem 1	Problem 2	Problem 1	Problem 2
Quiz 1 Total			Quiz 2 Total		Quiz 3 Total		Quiz 4 Total	
☑ Bonus SRN			☑ SRN 2		☑ SRN 3		☑ SRN 4	

Survey

We are curious about possible correlations. This will not affect your final grade. Please indicate:

Number of 6.034 lectures that you watched on OCW this semester, if any

Number of labs (out of labs 1-7) that you completed **after** the corresponding quiz

There are 48 pages in this exam. As always, this exam is open book, open notes, open almost everything—including a calculator—but no computers.

Quiz 1, Problem 1: Rule-Based Systems (45 points)

Janis wants to determine whether Cady is a Plastic, starting from the following rules:

P0	`IF '(?a) is a Plastic',`
	`THEN ('(?a) writes in the burn book',`
	`      '(?a) wears pink on Wednesdays')`

P1	`IF OR(AND('(?x) plots against (?y)',`
	`          '(?x) writes in the burn book'),`
	`       '(?x) is popular'),`
	`THEN ('(?x) is a Plastic')`

P2	`IF AND('(?b) is a Plastic',`
	`        '(?b) invites (?c) to lunch'),`
	`THEN ('(?c) plots against (?b)')`

Part A: Backward Chaining (20 points)

Make the following assumptions about backwards chaining:

- The backward chainer tries to find a matching assertion in the list of assertions. If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent. In case no matching consequents are found, the backward chainer concludes that the hypothesis is false.
- The backward chainer never alters the list of assertions, so it can derive the same result multiple times.
- Rules are tried in the order they appear.
- Antecedents are tried in the order they appear.
- Lazy evaluation/short circuiting is in effect

Janis starts with three **assertions for backward chaining:**

```
A0: 'Regina invites Cady to lunch'
A1: 'Regina is a Plastic'
A2: 'Cady writes in the burn book'
```

A1 (17 points) Using these assertions, perform backward chaining starting from the hypothesis:

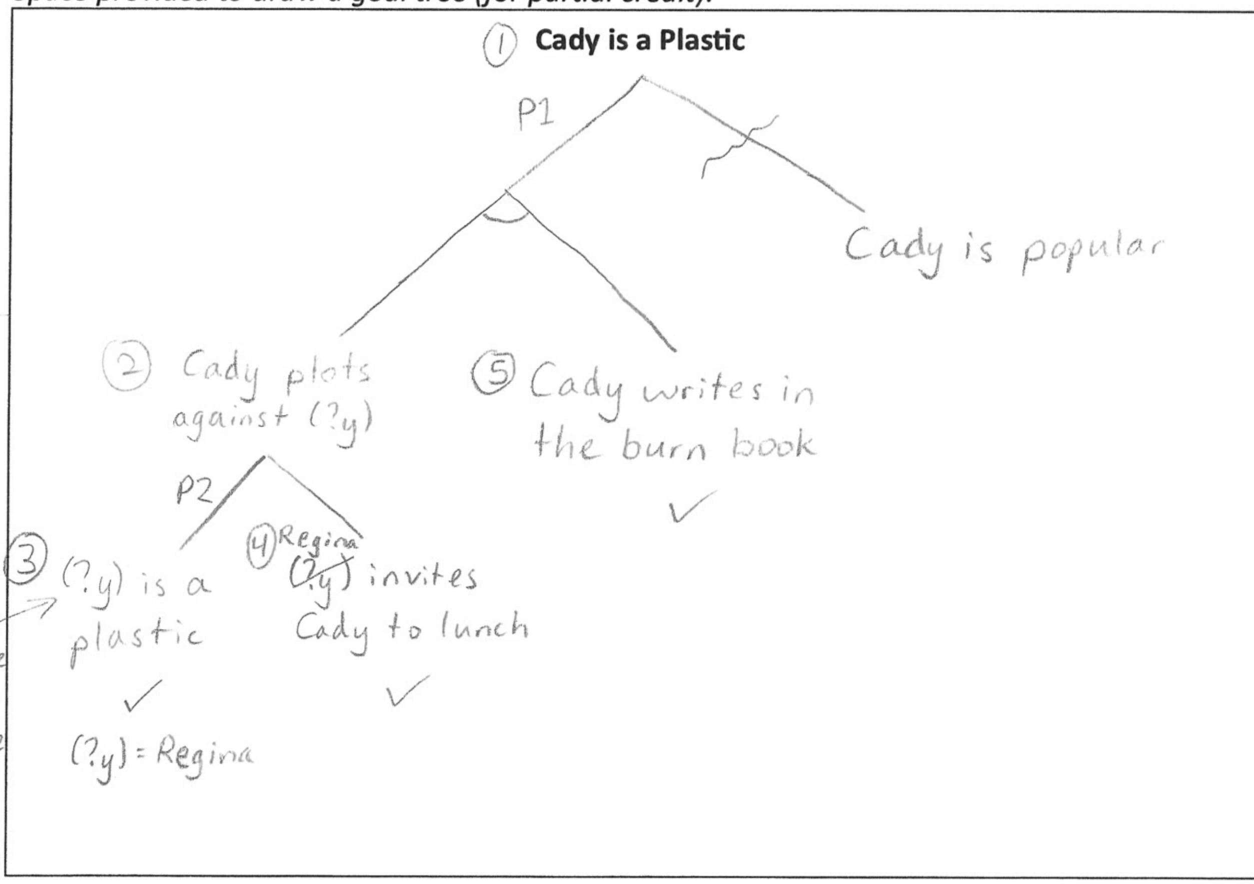
`'Cady is a Plastic'`

- In the table on the next page, write all the hypotheses that the backward chainer checks, in the order they are checked. (The first line has been filled in for you, and the table may have more lines than you need.)
- You can show your work for partial credit: Use the space provided to draw the goal tree that would be created by backward chaining from this hypothesis.

Hypotheses looked for in backward chaining

1. `Cady is a Plastic`
2. Cady plots against (?y)
3. (?y) is a plastic
4. Regina invites Cady to lunch
5. Cady writes in the burn book
6.
7.
8.

Space provided to draw a goal tree (for partial credit).



(?y) because it needs to have the same value as the (?y) above

A2 (3 points) Does backward chaining prove the hypothesis `'Cady is a Plastic'`? (Circle one)

YES NO

Part B: Forward Chaining (10 points)

Damian uses a smaller set of rules and starts with **different assertions**.

Rules:

P0	`IF '(?a) is a Plastic',`
	`THEN ('(?a) writes in the burn book',`
	`'(?a) wears pink on Wednesdays')`

P1	`IF OR(AND('(?x) plots against (?y)',`
	`'(?x) writes in the burn book'),`
	`'(?x) is popular'),`
	`THEN ('(?x) is a Plastic')`

Assertions:

```
A0: 'Cady writes in the burn book'
A1: 'Cady plots against Regina'
```

Using **<u>these new rules and assertions</u>**, fill in the table below for the first three iterations of forward chaining.

- List the rules whose antecedents match the assertions, the rule that fires, and any new assertions that are added.
- If no rules match or fire, or no new assertions are generated, write NONE in the corresponding box.

Make the following assumptions about forward chaining:

- When multiple rules match, rule-ordering determines which rule fires.
- New assertions are added to the bottom of the list of assertions.
- If a particular rule matches in multiple ways, the matches are considered in top-to-bottom order of the matched assertions. (For example, if a particular rule has a match with A1, and another match with A2, the match with A1 is considered first.)

Iter	Rules Matched	Rule Fired	New Assertions Added
1	P1	P1 ((?x) = Cady (?y = Regina)	A2: Cady is a Plastic
2	P0, P1	P0 ((?a) = Cady)	A3: Cady wears pink on Wednesdays
3	P0, P1	NONE	NONE

We don't add "Cady writes in the burn book" because it's already in the list of assertions.

Part C: General Forward Chaining (15 points)

This part contains 5 independent forward-chaining scenarios. In each one:

- Every rule in the system is represented by a column.
- The table shows the first two rounds of forward chaining. Forward chaining may continue for more than two rounds (not shown).
- In each round, each rule can match and/or fire, as indicated in the cells of the row.
- Assume there are no NOT or DELETE statements.

For each question, answer the question "Is this a valid forward-chaining scenario?" by circling YES or NO. If the answer is NO, briefly explain why (in about 10 words or less).

C1

	Rule 1	Rule2	Rule 3
Round 1	Match	Match, Fire	–
Round 2	Match	Match	Match, Fire

YES

NO (explain):

C2

	Rule 1	Rule2	Rule 3
Round 1	Match, Fire	Match, Fire	–
Round 2	Match	Match	Match, Fire

YES

NO (explain):

Only one rule can fire per round

C3

	Rule 1	Rule2
Round 1	Match, Fire	Match
Round 2	–	Match, Fire

YES

NO (explain):

Without NOTs and DELETEs, a rule will never stop matching once it matches.

C4

	Rule 1	Rule2
Round 1	–	Match, Fire
Round 2	–	Match, Fire

YES (Some rules might never match)

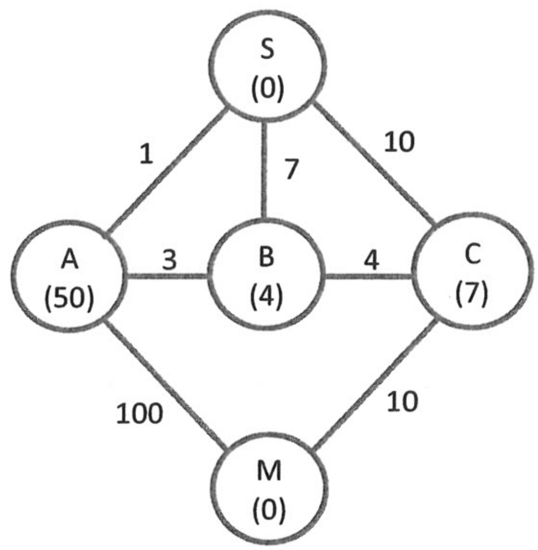
NO (explain):

Quiz 2, Problem 2: Search for the One Ring! (30 points)

Part A: Frolicking Frodo (13 points)

Frodo Baggins, a small Hobbit, has been tasked with destroying the One Ring. In order to do this, he must travel from his home in the Shire all the way to Mordor, across daunting mountains and spooky forests.

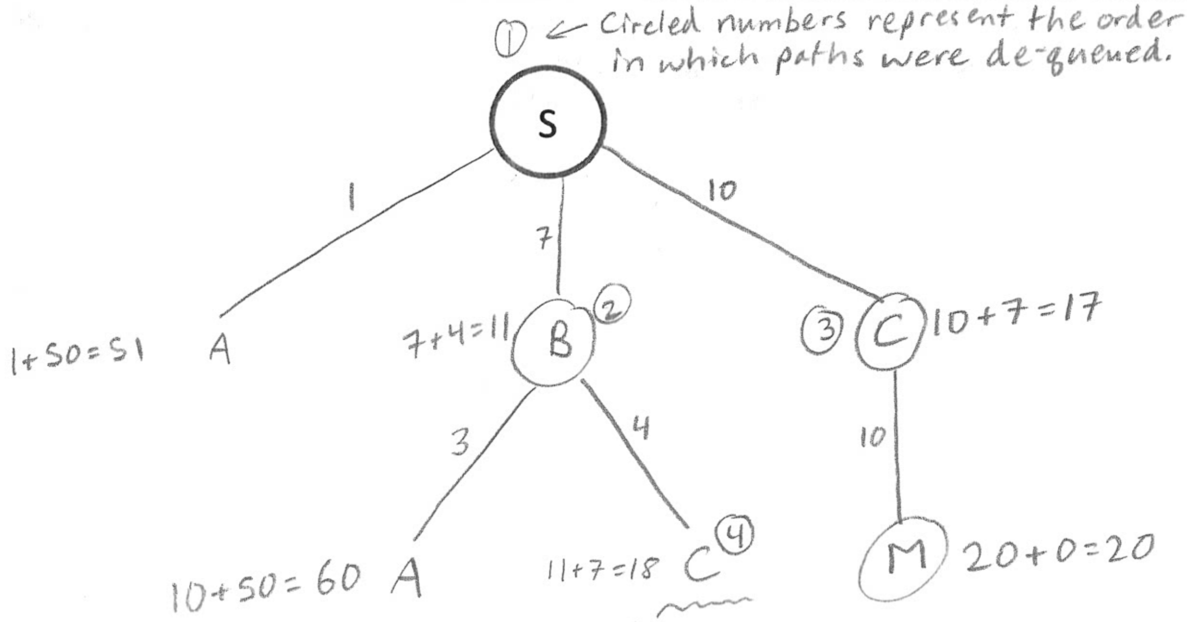
A1 (10 points) Frodo wants to take the shortest path to reach Mordor using a map of Middle Earth, shown below. The map has distances marked, numbers inside each node indicate heuristic estimates of the distance to Mordor. Frodo has not taken 6.034, but he remembers learning A* search from Gandalf, his wizard pal.



On the next page, help Frodo **perform <u>A* search</u>** to find a path from the Shire (**S**) to Mordor (**M**).

- Draw the children of each node in alphabetical order. (A < B < C)
- Break any ties using lexicographic/dictionary order. (S-P-Y < S-Q-X) (If two paths S-P-Y and S-Q-X are tied, extend path S-P-Y before path S-Q-X because 'SPY' would come before 'SQX' in the English dictionary.)
- Use the naive implementation of an extended set: If a path's last node is already in the extended set, don't extend that path.

Draw your search tree for <u>A* search</u> in the space below.



Extended set: S, B, C, M

A2 (2 point) What path does Frodo find using A* search in part A1? (Write the path found, including nodes S and M, or write NONE if he found no path.)

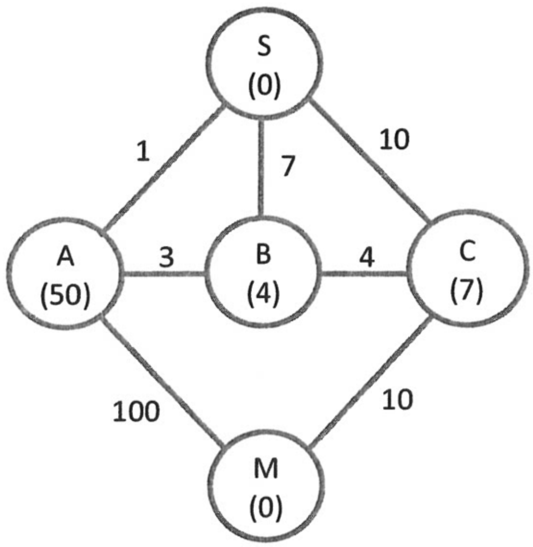
S, C, M

A3 (1 point) Does Frodo find the shortest path to Mordor? **YES** **NO**

Part B: Greedy Gollum (13 points)

B1 (10 points) Frodo embarks on his quest to destroy the One Ring. Little does he know that Gollum, who is obsessed with the Ring, is determined to steal it before Frodo reaches Mordor. In order to do this, Gollum must find a shorter path!

Gollum uses the same map of Middle Earth:

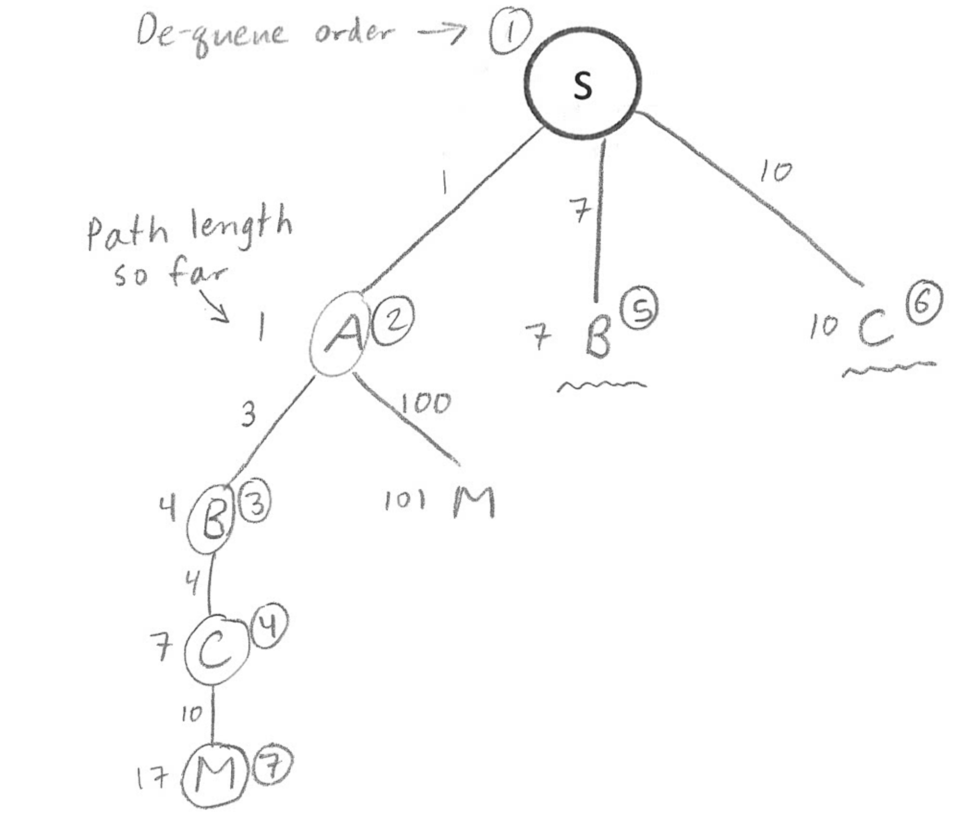


Gollum knows that A* search is not guaranteed to result in the shortest path, so he uses a different search method.

On the next page, help Gollum **perform branch and bound search with an extended set** and no heuristic to find a path from the Shire (**S**) to Mordor (**M**).

- Draw the children of each node in alphabetical order. (A < B < C)
- Break any ties using lexicographic/dictionary order. (S-P-Y < S-Q-X)
- Use the naive implementation of an extended set: If a path's last node is already in the extended set, don't extend that path.

Draw your search tree for **branch and bound with an extended set** in the space below.



Extended set: S, A, B, C, M

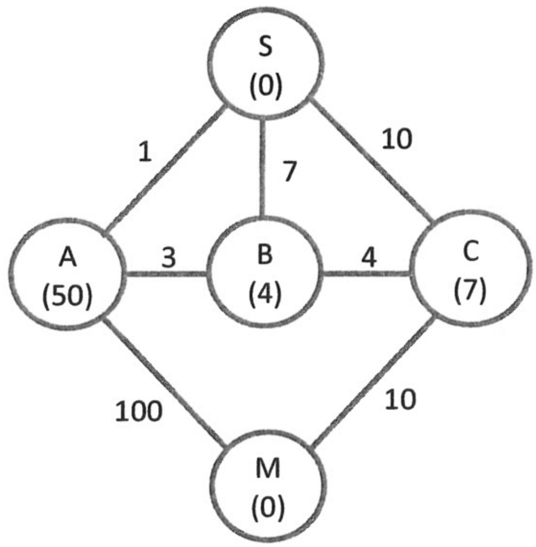
B2 (2 point) What path does Gollum find using branch and bound with an extended set in part B1? (Write the path found, including nodes S and M, or write NONE if he found no path.)

S, A, B, C, M

B3 (1 point) Does Gollum find the shortest path to Mordor? **YES** **NO**

Part C: Frodo's Fallacy (4 points)

Frodo mistakenly believes that A* search always guarantees the shortest path to the goal, but this is only true when the heuristic is consistent. Find one node whose heuristic value you could change to make the graph's heuristic consistent. (If there is more than one that works, pick any one.)



Node to change: A

New heuristic value: 1

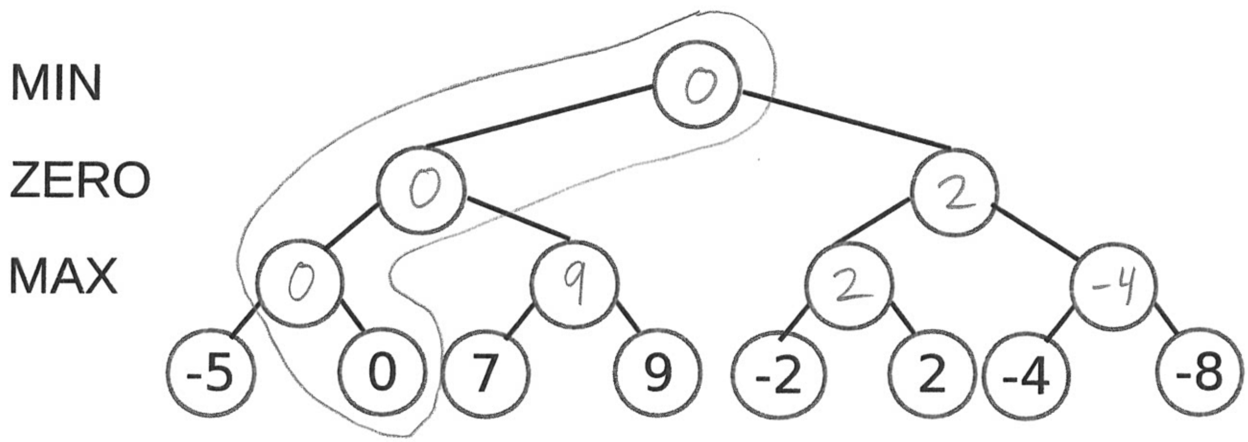
Quiz 1, Problem 3: Games (25 points)

Part A: Mini Midi Maxi (10 points)

The year: 1969; the place: a secret, sub-volcanic lair; our characters: Mini-Me, Dr. Evil, and Fat Bastard. Each character has their own unique strategy to tackling life's problems, even 3-player games.

Mini-Me tries to minimize the overall score and **Fat Bastard tries to maximize** the overall score. Unable to make up his mind, **Dr. Evil tries to keep the score as close to zero as possible**.

The game tree below shows the static values (heuristic scores) three moves ahead. Mini-Me (MIN) plays first, then Dr. Evil (ZERO), then Fat Bastard (MAX).



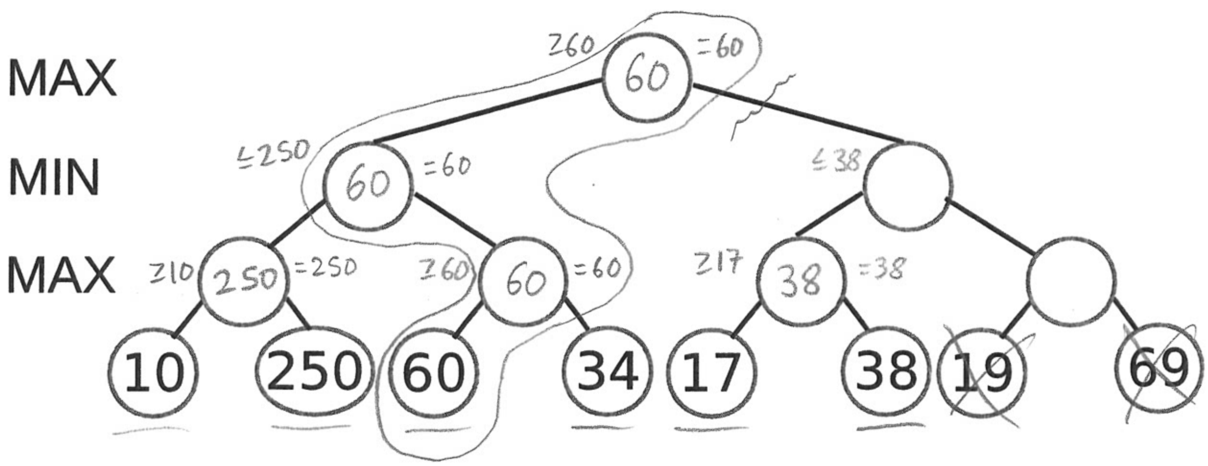
Complete the game tree above using NO pruning:

1. Write the score in each node.
2. <u>Circle</u> the optimal sequence of moves that the three characters will play.

Part B: Funk Alpha-Beta (15 points)

Tired of this rivalry, Dr. Evil escapes, only to be cornered by Austin Powers, who challenges him to a DJ battle! Austin and Dr. Evil will take turns playing songs to change the room's funk score. **Austin wants to maximize** the funk, while **Dr. Evil wants to minimize** it. (There is no zero player this time.) To win his mojo back, Austin will need to get a funk score greater than 50.

The game tree below shows the static values (heuristic scores) three moves ahead. Austin (MAX) plays first, then Dr. Evil (MIN).



B1 (13 points) Complete the game tree above using **minimax with alpha-beta pruning:**

1. Write the score in each node, if known.
2. **Cross out** any leaf nodes that will never get evaluated.
3. **Circle the minimax path** (the optimal sequence of songs that Austin and Dr. Evil will play).

B2 (2 points) Does Austin win his mojo back? In other words, does Austin achieve a funk score greater than 50? (Circle one)

YES **NO**

No quiz material on this page.

Quiz 2, Problem 1: Constraints in Sudoku (50 points)

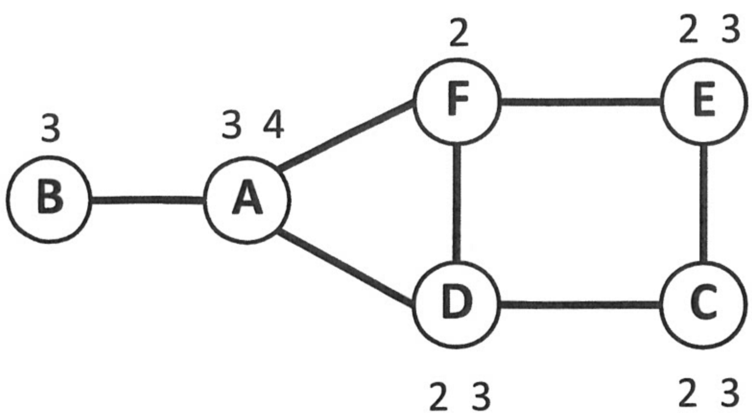
Sarah loves solving Sudoku puzzles, especially the ones in The Tech. She wonders if it's possible to use constraint propagation from 6.034 to solve her puzzles! Before tackling the standard 9x9 Sudoku, Sarah wants to test her idea on a simpler 4x4 puzzle, shown below.

1	F	E	4
4	D	C	1
2	1	4	3
B	A	1	2

The object of Sudoku is to fill each empty box with a 1, 2, 3, or 4, subject to the following rules:

1. Each number can appear only once in each row.
2. Each number can appear only once in each column.
3. Each number can appear only once in each 2x2 quadrant (top left, top right, bottom left, and bottom right).

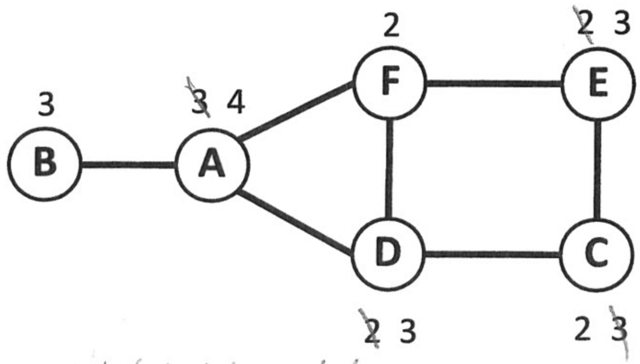
Sarah has labeled the empty boxes **A**, **B**, **C**, **D**, **E**, and **F**, which are the variables for this Sudoku problem. Below, she has drawn the corresponding constraint graph, in which each constraint is "must not be equal."



Part A: Domain Reduction Before Search (22 points)

First, Sarah wants to try reducing domains before search. Additionally, she has a feeling that box **F** cannot have the value **3**. Perform **domain reduction before search***, starting from the initial domains shown in the table below. The constraint graph is duplicated here with initial domains for your convenience.

*That is, add all the variables the propagation queue, then propagate through their constraints.



A	3 4
B	3
C	2 3
D	2 3
E	2 3
F	2

Queue: ~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ ~~F~~ ~~A~~ ~~D~~ ~~E~~ ~~C~~

★ For credit, show your work below by filling out the domain worksheet.
Add variables to the queue in ALPHABETICAL ORDER.

Fill out this worksheet as you reduce domains. There may be more rows than you need.

1. Every time you **remove a variable from the propagation queue**, fill out a new row in the table. (The same variable might appear in more than one row.)
2. In that row, write the name of the **variable you de-queued.** In the second column, list the **values that were just eliminated from neighboring variables** as a result. If no values were just eliminated, write **NONE** instead.
3. If you add several variables to your propagation queue at once, break ties by adding variables to your propagation queue **in alphabetical order**. Only add variables to your propagation queue **if they are not currently in the queue**.

	Var assigned or de-queued	List all values just eliminated from neighboring variables		Var assigned or de-queued	List all values just eliminated from neighboring variables
1	A	NONE	7	A	NONE
2	B	A≠3	8	D	C≠3
3	C	NONE	9	E	NONE
4	D	NONE	10	C	NONE
5	E	NONE	11		
6	F	D≠2, E≠2	12		

Example row showing a de-queued (propagated) variable

ex	X	W ≠ 1, 4 (example)

Part B: Singleton Sudoku (28 points)

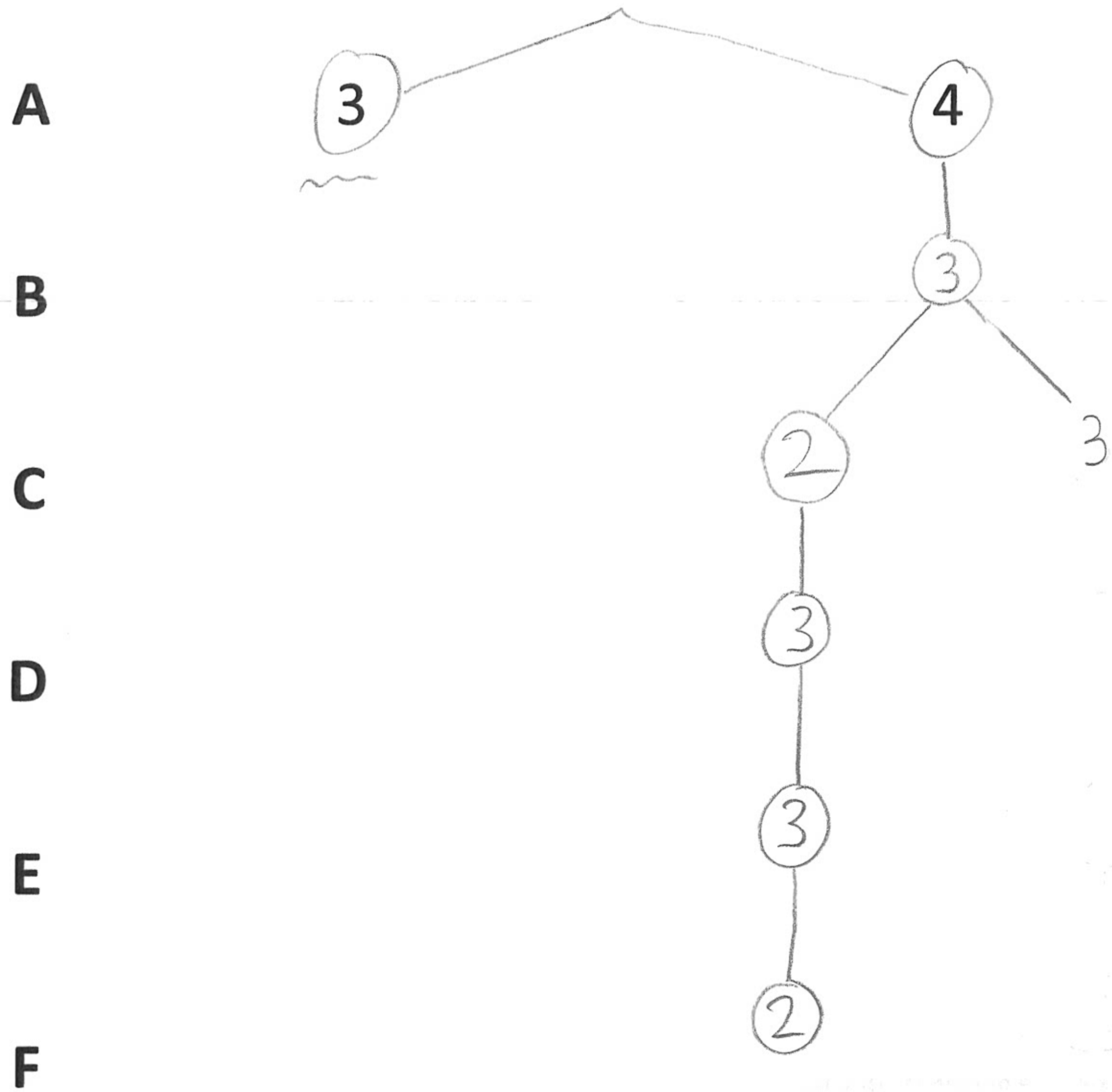
Neil, another Sudoku fanatic, notices Sarah using domain reduction to solve her Sudoku puzzle. He suggests, "Instead of guessing the value for box **F**, you could use use a constraint satisfaction method."

B1 (25 points) Use **depth-first search with forward checking and propagation through singleton domains** to solve the Sudoku puzzle, assigning variables in ALPHABETICAL ORDER (**A, B, C, D, E, F**). Start with the initial domains shown on the next page (note that **F**'s domain now includes **3**), and **DO NOT** perform any additional domain reduction before search.

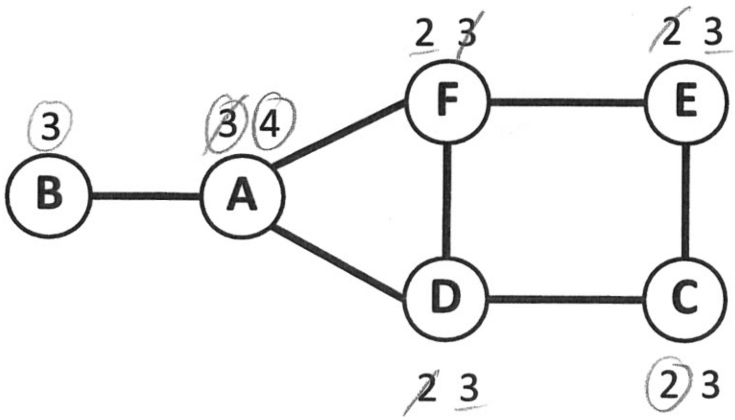
★ Show your work by simultaneously

(1) filling out the domain worksheet on the next page (or the duplicate copy on page 18), and

(2) drawing the search tree below (optional, for partial credit).



Constraint graph for this problem



Domains for this problem

(Do not reduce these domains before starting search.)

A	3 4
B	3
C	2 3
D	2 3
E	2 3
F	2 3

For credit, fill out the worksheet below. There may be more rows than you need.

1. Every time you **assign a variable** or **remove a variable from the propagation queue**, fill out a new row in the table. (The same variable might appear in more than one row, especially if you have to backtrack.)
2. In that row, indicate **which variable you assigned or de-queued**; write its **assigned** value if it has one (e.g. X=x), otherwise just write its **name** (X). In the second column, list the **values that were just eliminated from neighboring variables** as a result. If no values were just eliminated, write **NONE** instead.
3. If your search has to backtrack after assigning or de-queuing a variable: first, **finish listing** all values eliminated from neighboring variables in the current row. Next, check the "backtrack" box in that row. Then, continue with the next assignment in the following row as usual.
4. If you add several variables to your propagation queue at once, break ties by adding variables to your propagation queue **in alphabetical order**. Only add variables to your propagation queue **if they are not currently in the queue**.

	Var assigned or de-queued	List all values just eliminated from neighboring variables	Back track
1	A=3	B≠3, D≠3, F≠3	☑
2	A=4	NONE	☐
3	B=3	NONE	☐
4	C=2	D≠2, E≠2	☐
5	D	F≠3	☐
6	E	NONE	☐
7	F	NONE	☐
8	D=3	NONE	☐
9	E=3	NONE	☐
10	F=2	NONE	☐
11			☐
12			☐
13			☐
14			☐

Example row showing an assigned variable

ex	X = 3	Y ≠ 3, 4 Z ≠ 3 (example)	☐

Example row showing a de-queued (propagated) variable

ex	X	W ≠ 1, 4 (example)	☐

B2 (3 points) Based on the results of your search in Part B1, fill out the Sudoku puzzle below.

1	F 2	E 3	**4**
4	D 3	C 2	**1**
2	**1**	**4**	**3**
B 3	A 4	**1**	**2**

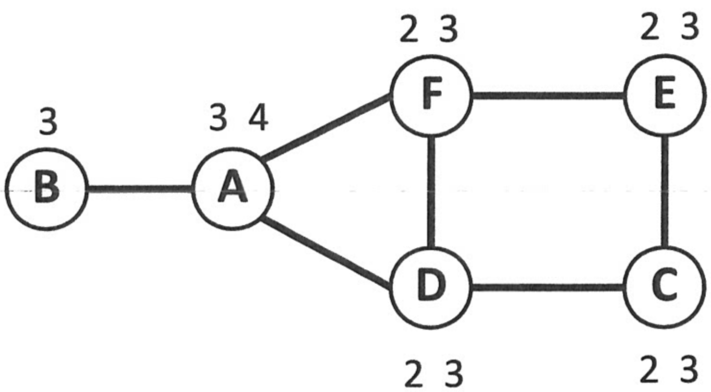
(If no solution exists, circle NO SOLUTION instead.)

NO SOLUTION

This is a duplicate copy of Part B1. If you want to rewrite your domain reduction worksheet here and have this copy graded, check the box:

☐ **I want to start over; grade this copy**

Constraint graph for this problem



Domains for this problem

(*Do <u>not</u> reduce these domains before starting search.*)

A	3 4
B	3
C	2 3
D	2 3
E	2 3
F	2 3

	Var assigned or de-queued	**List all values just eliminated from neighboring variables**	**Back track**		**Var assigned or de-queued**	**List all values just eliminated from neighboring variables**	**Back track**
1			☐	8			☐
2			☐	9			☐
3			☐	10			☐
4			☐	11			☐
5			☐	12			☐
6			☐	13			☐
7			☐	14			☐

Quiz 2, Problem 2: ID Trees & kNN (50 points)

Part A: Identification Trees (28 points)

Hydra spies want to determine whether various heroes are members of the Justice League. They have information about 7 known heroes, shown in the table below. They ask you to use the information to create an identification tree.

Hero	Justice League	Has Powers	Is From Earth	Lives in Gotham	Lives in Central
1 Bruce Wayne	Yes	No	Yes	Yes	No
2 Clark Kent	Yes	Yes	No	No	No
3 Oliver Queen	Yes	No	Yes	No	No
4 Barry Allen	Yes	Yes	Yes	No	Yes
5 Joe West	No	No	Yes	No	Yes
6 Jim Gordon	No	No	Yes	Yes	No
7 Felicity Smoak	No	No	Yes	No	No

A1 (4 points) Which of the following tests (if any) separate out a homogeneous branch? Circle all that apply, or circle NONE.

Has Powers — "Yes" branch is homogeneous

Is From Earth — "No" branch is homogeneous

Lives in Gotham

Lives in Central

NONE

A2 (5 points) Which two tests have the same test disorder? There are exactly two: Circle them.

Has Powers
Yes: YY
No: YY NNN

Is From Earth
Yes: YYY NNN
No: Y

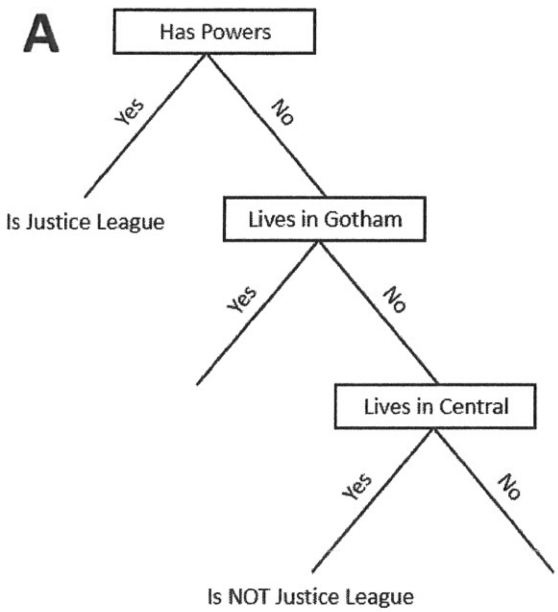
Lives in Gotham
Yes: Y N
No: YYY NN

Lives in Central
Yes: Y N
No: YYY NN

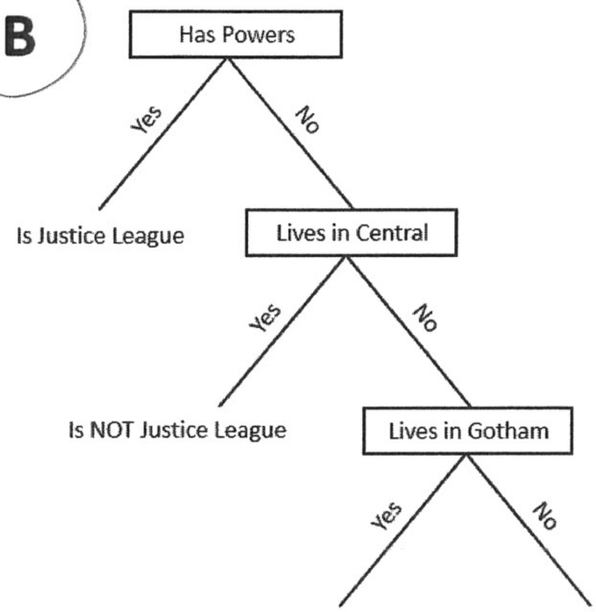
A3 (12 points) Grant Ward notices that the current tests cannot perfectly classify all 7 known heroes, so he sends the spies to get more information. In the meantime, Grant will construct a greedy, disorder-minimizing identification tree to partially classify the training data. He will break ties according to the order in the table: *Has Powers* before *Is From Earth*, etc. **Which of the following trees is the correct greedy, disorder-minimizing ID tree?** (Circle one)

This question can be answered without numerical calculation. For your convenience, ***the next page includes a copy of the training data,*** *a table of logarithms, and space to show work for partial credit.*

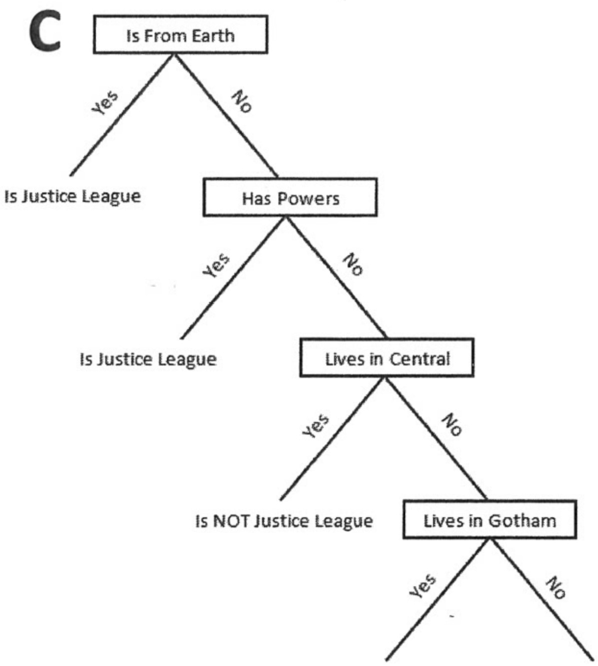
A



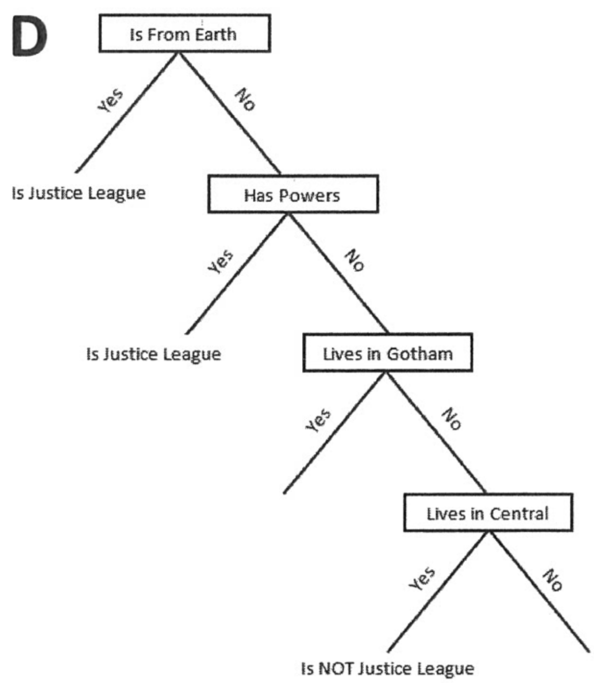
B



C



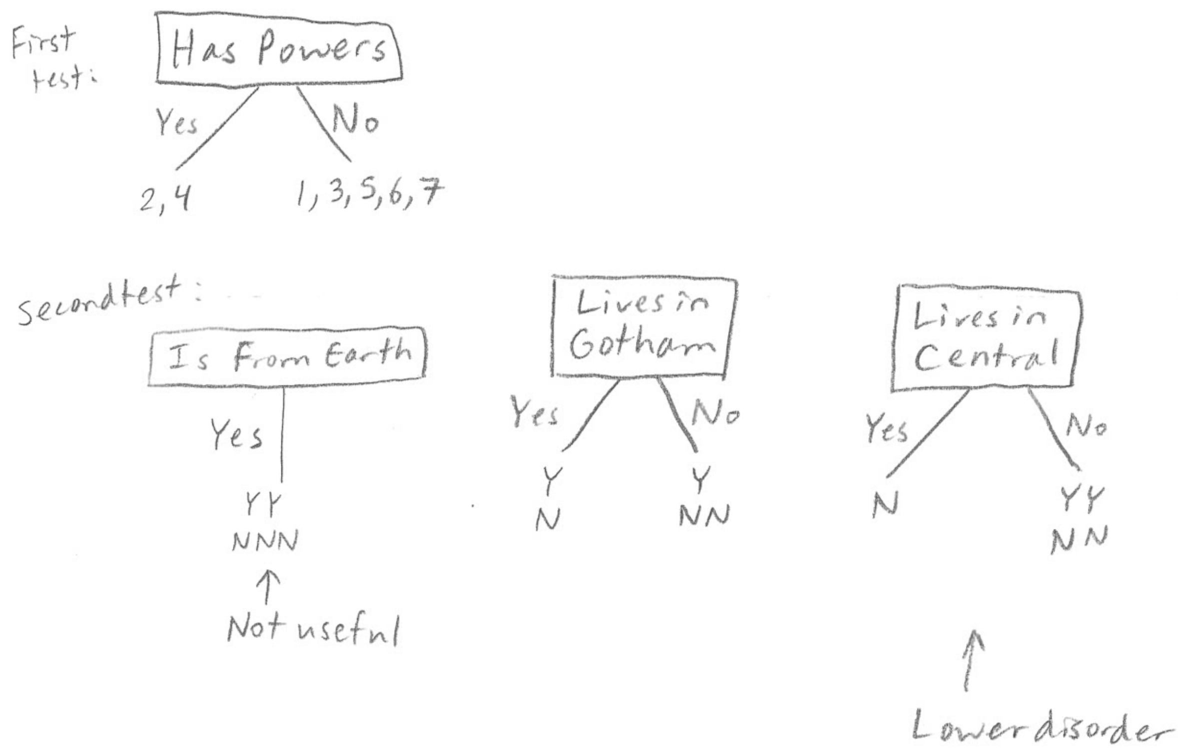
D



(This is a duplicate copy of the data.)

Hero	Justice League	Has Powers	Is From Earth	Lives in Gotham	Lives in Central
1 Bruce Wayne	Yes	No	Yes	Yes	No
~~**2** Clark Kent~~	~~Yes~~	~~Yes~~	~~No~~	~~No~~	~~No~~
3 Oliver Queen	Yes	No	Yes	No	No
~~**4** Barry Allen~~	~~Yes~~	~~Yes~~	~~Yes~~	~~No~~	~~Yes~~
5 Joe West	No	No	Yes	No	Yes
6 Jim Gordon	No	No	Yes	Yes	No
7 Felicity Smoak	No	No	Yes	No	No

Space provided to show your work.



$-\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1$ $\quad -\left[\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right] \approx 0.97$ $\quad -\left[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right] \approx 0.9$

$-\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right] \approx 0.8$ $\quad -\left[\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right] \approx 0.72$ $\quad -\left[\frac{1}{6}\log_2\frac{1}{6} + \frac{5}{6}\log_2\frac{5}{6}\right] \approx 0.65$

A4 (7 points) The Hydra spies have returned with four additional tests, shown below.

Hero	Justice League	Good with Technology	Lives in Star City	Attended MIT	Was a CEO
1 Bruce Wayne	Yes	Yes	~~No~~	~~No~~	Yes
2 Clark Kent	Yes	No	No	No	No
3 Oliver Queen	Yes	No	~~Yes~~	No	~~Yes~~
4 Barry Allen	Yes	Yes	No	No	No
5 Joe West	No	No	No	No	No
6 Jim Gordon	No	No	~~No~~	~~No~~	No
7 Felicity Smoak	No	Yes	~~Yes~~	Yes	~~Yes~~

1,6✓ 3,7✓ ~~1,6~~ ~~3,7~~ ~~1,6~~ 3,7✓ 1,6✓ ~~3,7~~

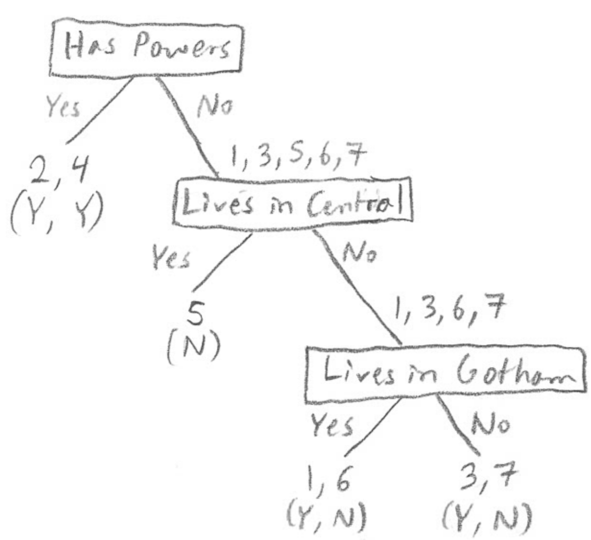
Determine whether you can use **exactly one** of the four new tests to extend your existing tree such that it will correctly classify all the training data points. Note that you may only extend your tree, which means you cannot replace or edit previous tests. However, you may use the new test in multiple branches of your tree.

Circle the one test that you can use to complete your ID Tree from the previous part. If none of the new tests allow you to do this, instead circle CAN'T BE DONE.

Good with Technology **Lives in Star City** **Attended MIT** **Was a CEO**

CAN'T BE DONE

For partial credit, show your work or ***briefly*** *explain your answer:*

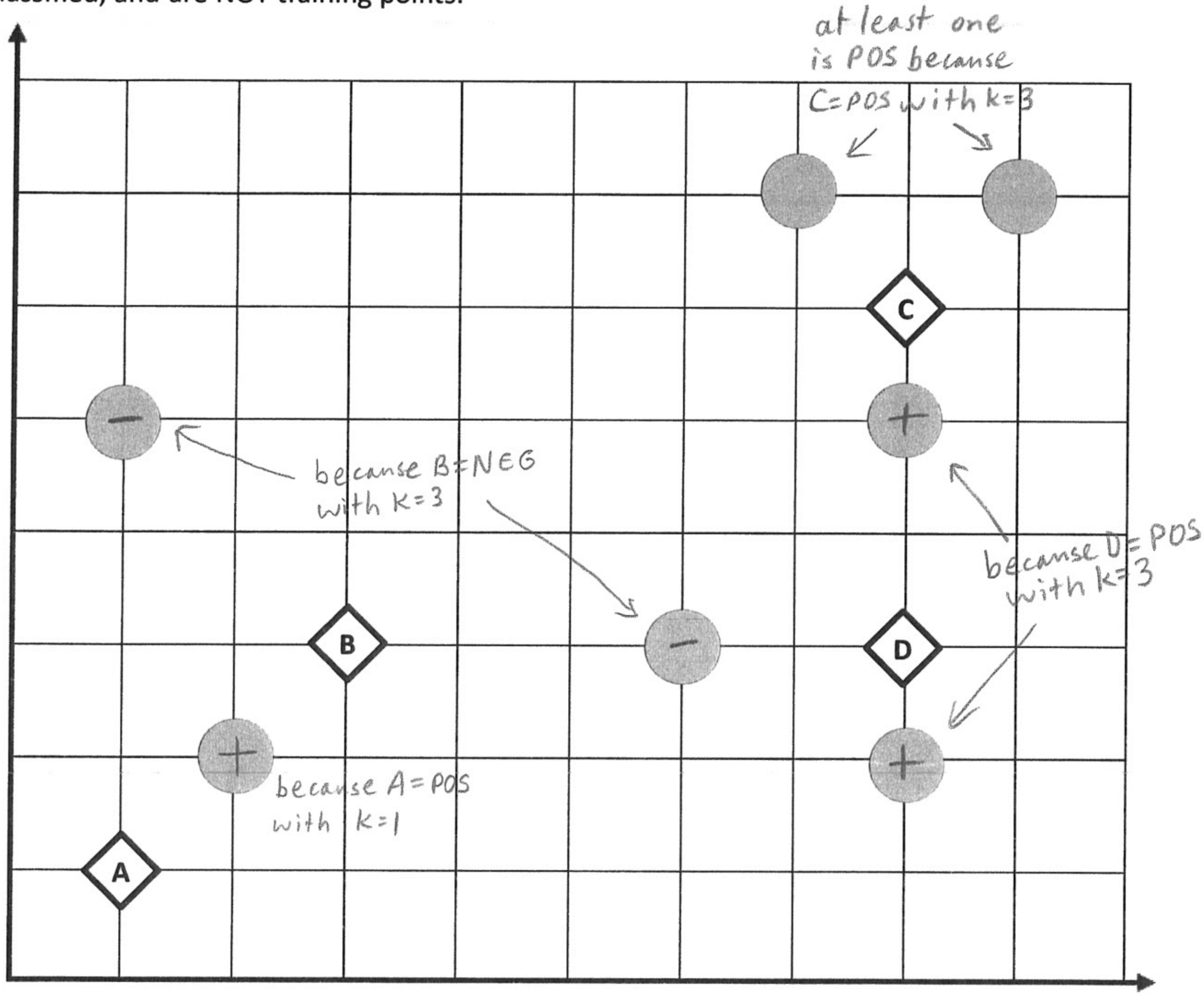


We're looking for a test that separates 1 from 6, and separates 3 from 7.

Only "Good with Technology" separates both pairs.

Part B: k-Nearest Neighbors (22 points)

In the graph below, each circle is a training points with known classification, which is either positive (POS) or negative (NEG). The lettered points are unknown test points that are being classified, and are NOT training points.

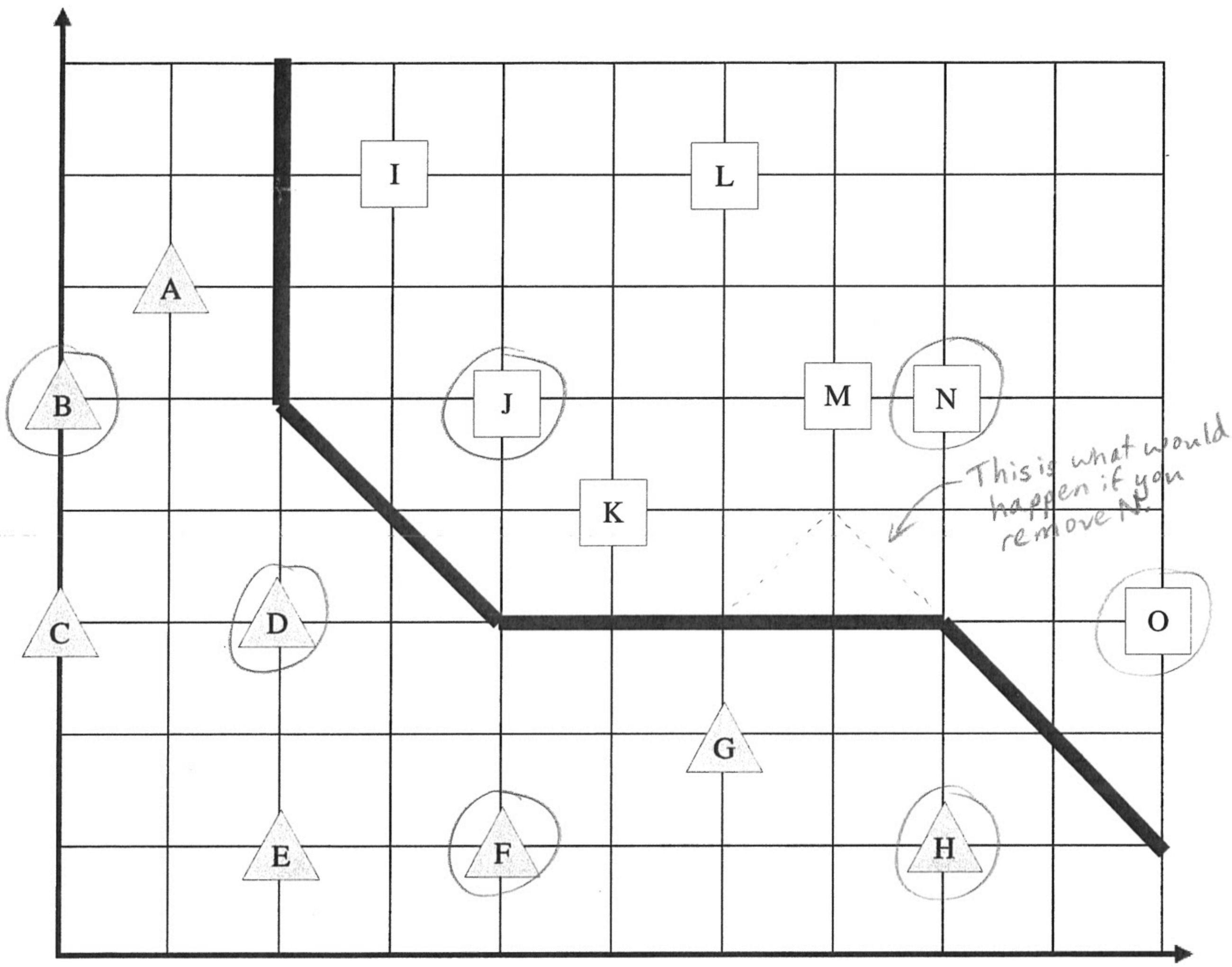


B1 (12 points) Spies started using the training data to classify the test points (A, B, C, D) using k-nearest neighbors, but they went off on a mission, leaving you with incomplete information. Using the graph above and the provided classifications in the table below, fill in the remaining table entries. If a classification cannot be determined, write UNKNOWN. For partial credit, indicate the classifications of the training points on the graph above.

	Number of nearest neighbors to consider		
Test point	**k=1**	**k=3**	**k=7**
A	POS	NEG	POS
B	POS	NEG	POS
C	POS	POS	POS
D	POS	POS	POS

B2 (10 points) The graph below shows a 1-nearest neighbor decision boundary corresponding to a set of exactly 7 training points. Out of the 15 potential training points marked on the graph, which set of 7 would result in this boundary? Write the names of the 7 training points in the box below.

B, D, F, H, J, N, O



Quiz 3, Problem 1: Networks of Neurons (50 points)

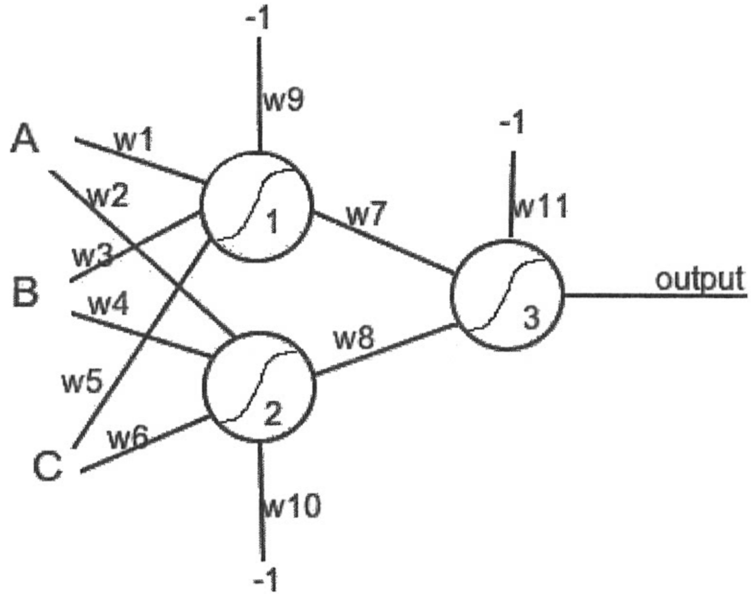
Part A: Forward Propagation (22 points)

The Olympic Games Committee has asked you to help them predict the outcomes of track races. Two runners will be considered for each race, and you'll have access to three inputs:

- A, the distance traveled thus far by Runner A,
- B, the distance traveled thus far by Runner B, and
- C, the total distance of the race.

Your job is to return a number from 0 to 1, representing the probability that the Runner A will win the race.

You decide to use the neural network below with a sigmoid function. The neurons are labeled 1, 2, 3 for reference.



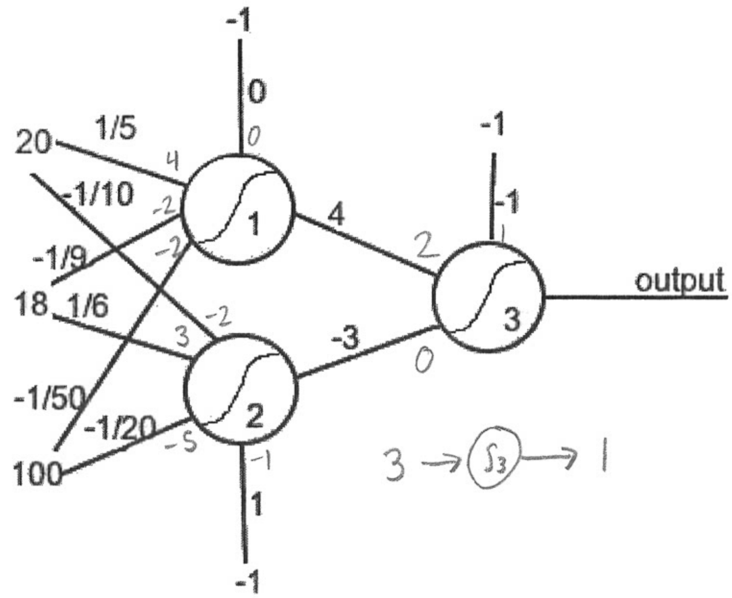
To get a feeling for the behavior of the net, you decide to run forward propagation with a few different configurations of the weights. For the following problems, you are to use a sigmoid function σ(x) with the following approximate values:

if $x \geq 3$, $\sigma(x) \approx 1$
if $x = 0$, $\sigma(x) = 1/2$
if $x \leq -3$, $\sigma(x) \approx 0$

For the purpose of forward propagation calculation, you may assume that the sigmoid function is linear from x=-3 to x=3.

A1 (12 points) Using the inputs and weights show below, find the approximate output of each neuron. The inputs A, B, C have values 20, 18, and 100, and the weights are indicated on their corresponding wires. Show your work for partial credit.

$0 \to S_1 \to 1/2$



$-6 \to S_2 \to 0$

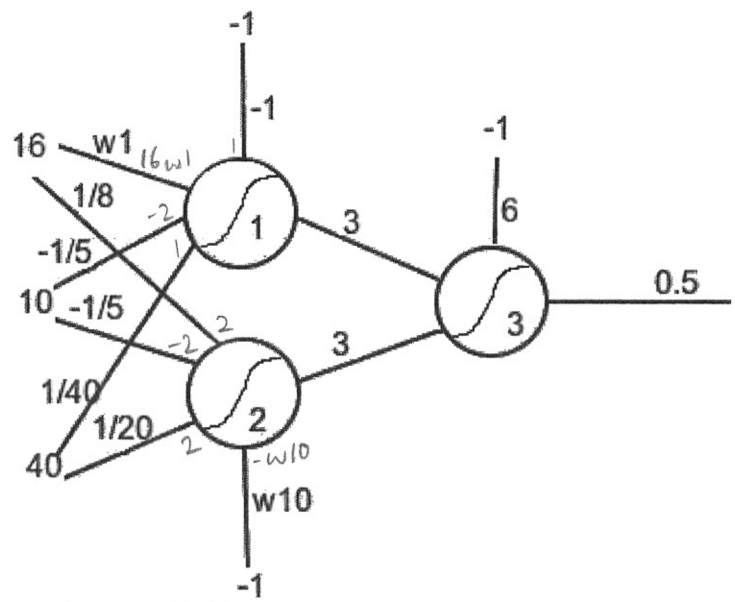
Approximate output of neuron 1: 1/2

Approximate output of neuron 2: 0

Approximate overall output: 1

A2 (10 points) This time, some of the weights are missing! Find a possible value for each missing weight (w1 and w10). The inputs A, B, C are 16, 10, and 40, and the weights are indicated on their corresponding wires. Show your work for partial credit. (Hint: This question does not require significant calculation.)

$16(w1) \rightarrow S_1 \rightarrow out_1$



$2 - w10 \rightarrow S_2 \rightarrow out_2$

At neuron 3, we need $\sigma(3*out_1 + 3*out_2 - 6) = 1/2$,

which means $3*out_1 + 3*out_2 - 6 = 0$, so $out_1 + out_2 = 2$.

Each output can be at most 1, so we need $out_1 \approx out_2 \approx 1$.

That means the input to neurons 1 and 2 must be ≥ 3, which we can achieve by making w1 very large (positive) and w10 very small (negative): for example, $w1 = 100$ and $w10 = -100$.

w1 = $\boxed{100}$

Mathematically, we need:

$16(w1) \geq 3$ and $2 - w10 \geq 3$

so: $w1 \geq \frac{3}{16}$ and $w10 \leq -1$

w10 = $\boxed{-100}$

(many possible answers)

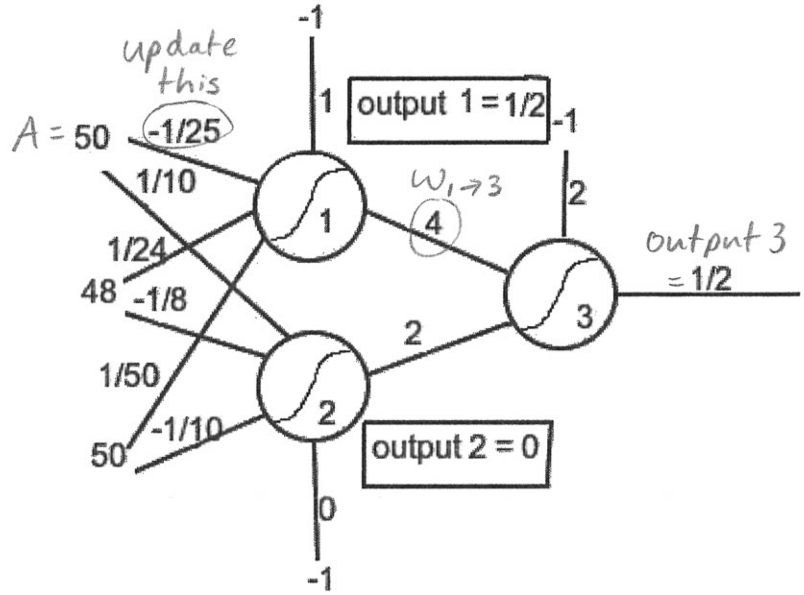
Part B: Back Propagation (13 points)

out* = 1

You perform forward propagation with the weights shown below and notice that the output is not what you want it to be: For a 50m race, if Runner A has traveled 50m and Runner B has only traveled 48, then **you want the neural net's output to be 1** (because the probability of Runner A winning the race when they have just won should be 1).

$r = 1/2$

You begin training the net using back propagation with **a training rate of 1/2**. As shown below, the inputs A, B, C are 50, 48, and 50, and weights are as indicated. The forward-propagation output of each neuron is also provided. What will be the change in w1 (that is, $w1_{new} - w1_{old}$) after a single step of back propagation? w1 is the weight going from input A (50) to neuron 1, with an initial value of -1/25.



$\Delta w1 =$ $w1_{new} - w1_{old}$ = $\boxed{25/8}$

Space provided to show work for partial credit

$$\Delta w1 = r \cdot out_A \cdot \delta_1 = \tfrac{1}{2} \cdot 50 \cdot (\tfrac{1}{2})^3 = \tfrac{25}{8}$$

$$\delta_1 = out_1(1 - out_1)(w_{1\to3}\,\delta_3)$$
$$= 1/2\ (1 - 1/2)\ (4 \cdot \delta_3) = \tfrac{1}{2}(\tfrac{1}{2})(4 \cdot (\tfrac{1}{2})^3) = (\tfrac{1}{2})^3$$

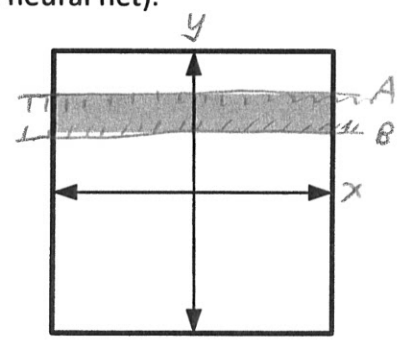
$$\delta_3 = out_3(1 - out_3)(out^* - out) \quad \text{same as } out_3$$
$$= 1/2\ (1 - 1/2)(1 - 1/2) = (\tfrac{1}{2})^3$$

Part C: Pictures (15 points)

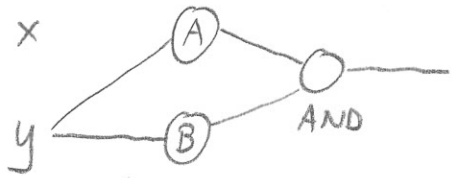
Each of the following pictures can be produced by a neural net with two inputs and one output. For each one, determine the **minimum** number of neurons necessary to produce the picture. **Express your answer as a list indicating the number of neurons in each layer** (as in Lab 6).

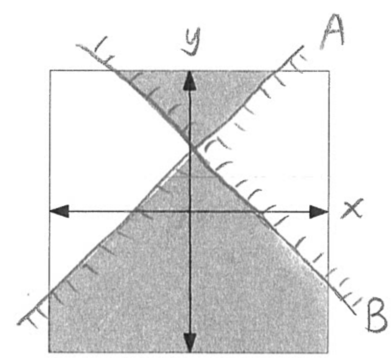
For example, the neural net from Parts A and B would be represented by the list [2, 1].

Show your work for partial credit (for example, sketch your minimal neural net).

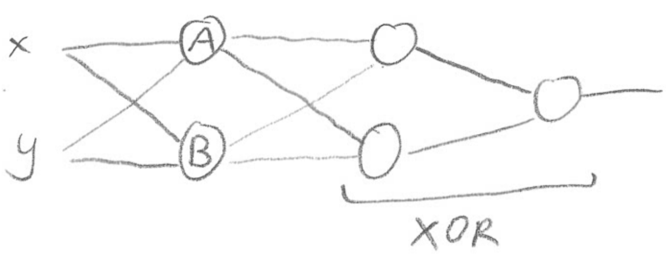


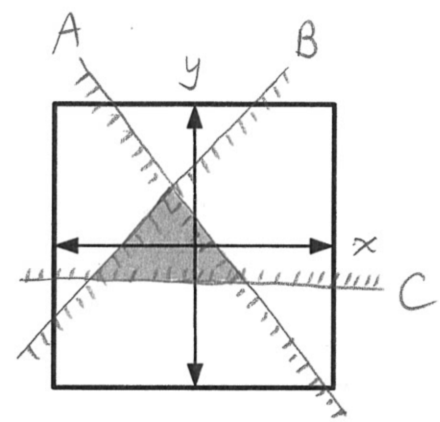
Network Layout: [2, 1]



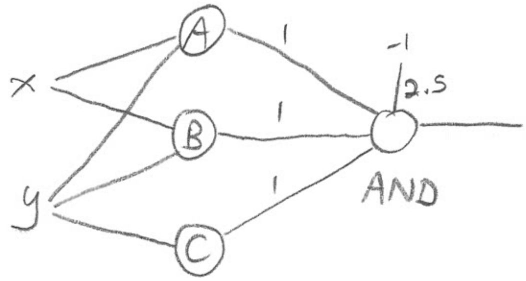


Network Layout: [2, 2, 1]





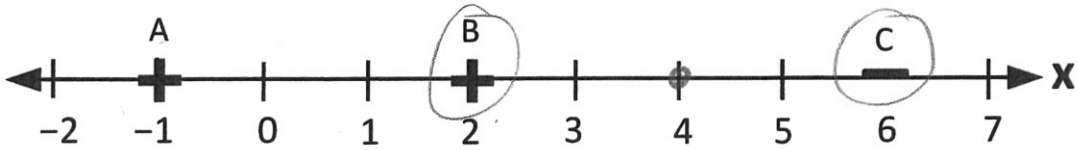
Network Layout: [3, 1]



Quiz 3, Problem 2: Support Vector Machines (50 points)

Part A: 1-D SVM (23 points)

Here is a 1-dimensional set of training data with one numeric feature (**x**), two positive training points (**A**, **B**) and one negative training point (**C**):



A1 (2 points) In general, what shape is the decision boundary for a 1-dimensional linear SVM? (Write one word.)

point

A2 (3 points) On the diagram above:

1. **Draw the decision boundary.** (Use a linear kernel.)
2. **Circle the support vectors.**

A3 (3 points) Write an equation representing...

...the decision boundary: $x = 4$

...the positive (+) gutter: $x = 2$

...the negative (−) gutter: $x = 6$

A4 (15 points) Based on the decision boundary and gutters you found above, determine the values of $\vec{w}$, b, and α_i for the boundary equation $\vec{w}\cdot\vec{x}+b\geq 0$.

$\vec{w}=$ $[-1/2]$	$b=$ 2

$\alpha_A=$ 0	$\alpha_B=$ 1/8	$\alpha_C=$ 1/8

Show your work on the next page for partial credit.

Space provided to show work

Boundary: $x = 4 \rightarrow x - 4 = 0$

$\vec{w} \cdot \vec{x} + b = 0$

$[1][x] + (-4) = 0$

Scale with gutter constraint:

Point B:

$c([1][2] + (-4)) = +1$

$-2c = 1$

$c = -1/2$

$w = -1/2 \cdot [1] = [-1/2]$

$b = -1/2\,(-4) = 2$

Eq. 4: $\alpha_B = \alpha_C$

Eq. 5: $\alpha_B[2] - \alpha_C[6] = [-1/2]$

$\rightarrow 2\alpha_B - 6\alpha_C = -1/2$

Solve: because $\alpha_B = \alpha_C$

$2\alpha_B - 6\alpha_B = -1/2$

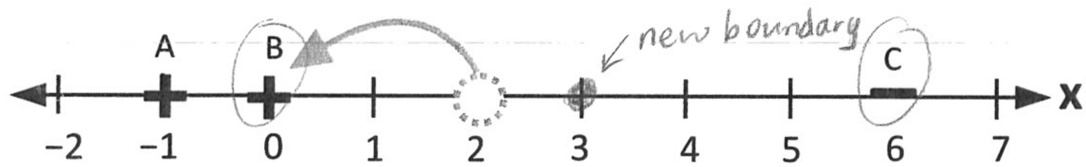
$-4\alpha_B = -1/2$

$\alpha_B = 1/8$

$\alpha_C = \alpha_B = 1/8$

Part B: Modifications (12 points)

Jessica modifies the original data in Part A by moving point **B** to **x**=0, as shown below.



How does moving point **B** affect the supportiveness values (α_i) relative to data in Part A? For each, indicate whether the value has increased, decreased, or stayed the same (no change). (Circle one for each)

margin width ↑
$\|\vec{w}\|$ ↓
α ↓

α_A:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**	$\alpha = 0$
α_B:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**	
α_C:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**	

What happens to the magnitude and direction of $\vec{w}$?

magnitude of $\vec{w}$:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**
direction of $\vec{w}$:	**OPPOSITE DIRECTION**		**NO CHANGE (SAME DIRECTION)**

Part C: Multiple Choice (15 points)

The following questions are about SVMs in general.

C1 (3 points) Neil trains an SVM using a linear kernel, resulting in a margin of width x. Then, he modifies the training data. When Neil retrains the SVM, he finds that the magnitude of $\vec{w}$ has **tripled**. What is the new margin width? (Circle one)

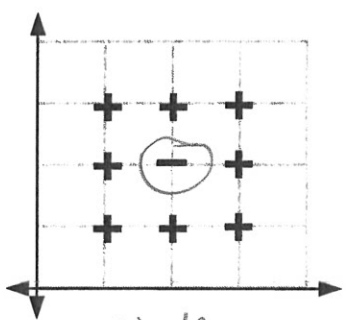
$\frac{1}{6}x$ $\frac{1}{3}x$ $\frac{2}{3}x$ x $\frac{3}{2}x$ $3x$ $6x$

old margin width $= x = \frac{2}{\|\vec{w}\|} \rightarrow$ old $\|\vec{w}\| = \frac{2}{x} \rightarrow$ new $\|\vec{w}\| = 3\left(\frac{2}{x}\right) = \frac{6}{x} \rightarrow$ new margin width $= \frac{2}{6/x} = \frac{2}{6}x = \frac{1}{3}x$

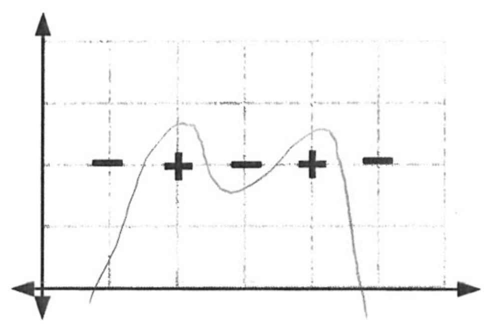
C2 (12 points) Professor Winston is looking for an example dataset to use in a new lecture on kernels. He tells you, "I need a **dataset that can be classified by an SVM with a quadratic kernel, but *not* with a linear kernel**." Which of the following datasets satisfy that property? For each, circle YES if it satisfies the property, otherwise NO. (Recall that a quadratic kernel can draw a conic section.)

YES

NO



circle or ellipse



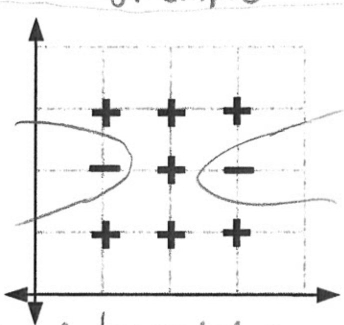
YES

NO

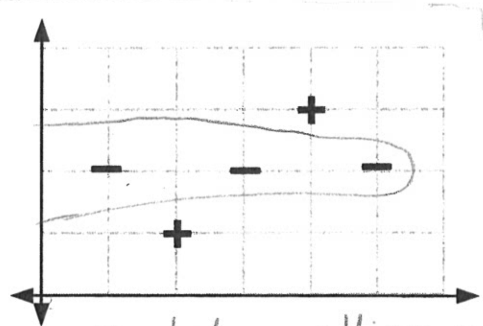
need polynomial of degree 4 or higher

YES

NO



pair of hyperbolas



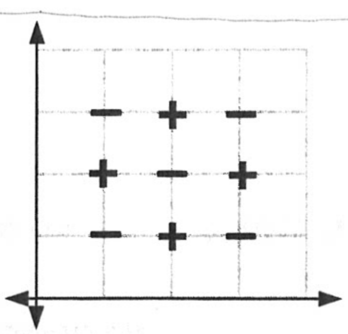
YES

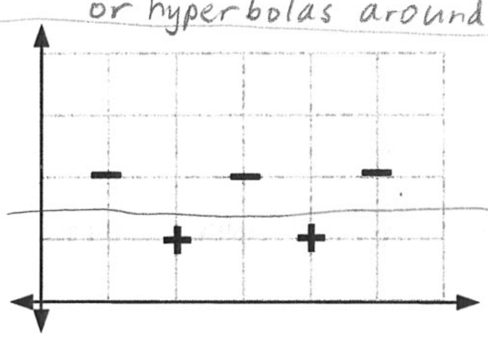
NO

parabola or ellipse around ⊖s or hyperbolas around ⊕s

YES

NO





YES

NO

linearly separable

No quiz material on this page.

Quiz 4, Problem 1: Adaboost (50 points)

Part A: Boosting in Wonderland (28 points)

After much adventuring, Alice once again encounters a fork in the path. She has 4 maps, none of which are entirely correct. At 5 previously encountered forks (A, B, C, D, E), Alice kept track of which maps were incorrect at each fork (or, equivalently, which forks were "misclassified" by each map). Unfortunately, Alice is writing on magical paper that causes some of her table entries to be written in invisible ink.

Here is Alice's (incomplete) data:

Map	Misclassified forks (A, B, C, D, E)
Map 1	B, C
Map 2	B, D, E
Map 3	A, D
Map 4	B, C, D, E

Map 2:
- misclassifies 3 of {A, B, C, D, E} in Round 1
- misclassifies 3 of {B, C, D, E} in Round 2
- does not misclassify C

Map 3:
- misclassifies 2 of {A, B, C, D, E} in Round 1
- in Round 2, misclassifies A with weight 1/2 and one of {B, C, D, E
- misclassifies D

A1 (12 points) Alice wants to know which forks (A, B, C, D, E) each map misclassified. She remembers the following:

- Map 1 misclassified forks B and C only.
- Map 2 did NOT misclassify fork C. (That is, Map 2 correctly classified fork C.)
- Map 3 misclassified at least fork D, and possibly other forks.

Before the ink became invisible, Alice performed Adaboost to construct a classifier to help her decide which path to take. Her Adaboost table, with missing entries, is shown on the next page. In each round, Alice picked the classifier with the **error rate furthest from 1/2**, and she broke ties by picking the map that comes first numerically: Map 1, Map 2, etc.

Using this information, and the incomplete Adaboost table on the next page, **determine which fork(s) each map misclassifies. Fill in the table above.**

Map 4:
- misclassifies 4 of {A, B, C, D, E} in Round 1
- misclassifies either A only or all of {B, C, D, E} in Round 2

Alice's Adaboost table:

	Round 1	Round 2	Round 3
weight A	1/5 ×	1/2 = 4/8	4/12
weight B	1/5	1/8 ×	1/4
weight C	1/5	1/8 ×	1/4
weight D	1/5	1/8	1/12
weight E	1/5	1/8	1/12
Error rate of Map 1	2/5	2/8 = 1/4	
Error rate of Map 2	3/5	3/8	
Error rate of Map 3	2/5	5/8	
Error rate of Map 4	4/5 →	4/8 = 1/2	
weak classifier chosen (h)	Map 4	Map 1	
weak classifier error (ε)	4/5	1/4	
voting power (α)	$-\frac{1}{2}\ln 4$	$\frac{1}{2}\ln 3$	

A2 (16 points) Fill in the missing entries in Alice's Adaboost table, above.

Space provided to show your work

$$\varepsilon = \frac{4}{5} \rightarrow \alpha = \frac{1}{2}\ln\frac{1-4/5}{4/5} = \frac{1}{2}\ln\frac{1/5}{4/5} = \frac{1}{2}\ln\frac{1}{4} = -\frac{1}{2}\ln 4$$

$$\varepsilon = \frac{1}{4} \rightarrow \alpha = \frac{1}{2}\ln\frac{1-1/4}{1/4} = \frac{1}{2}\ln\frac{3/4}{1/4} = \frac{1}{2}\ln 3$$

Part B: The Other Definition (10 points)

Alice wonders what would happen if she instead used **lowest error rate**. Perform boosting to fill in the missing entries in the table below. In each round, pick the classifier with the lowest error rate. Break ties by picking the map that comes first numerically.

	Round 1	Round 2
weight A	1/5	1/6
weight B	1/5 x	1/4
weight C	1/5 x	1/4
weight D	1/5	1/6
weight E	1/5	1/6
Error rate of Map 1	2/5	
Error rate of Map 2	3/5	
Error rate of Map 3	2/5	
Error rate of Map 4	4/5	
weak classifier chosen (h)	Map 1	
weak classifier error (ε)	2/5	
voting power (α)	$\frac{1}{2} \ln \frac{3}{2}$	

Space provided to show your work

$$\varepsilon = \frac{2}{5} \rightarrow \alpha = \frac{1}{2} \ln \frac{1 - 2/5}{2/5} = \frac{1}{2} \ln \frac{3/5}{2/5} = \frac{1}{2} \ln \frac{3}{2}$$

Part C: When will it end? (12 points)

This is a boosting question with three training points (A, B, C). There are two weak classifiers, h1 and h2. h1 misclassifies point A only, while h2 misclassifies point B only.

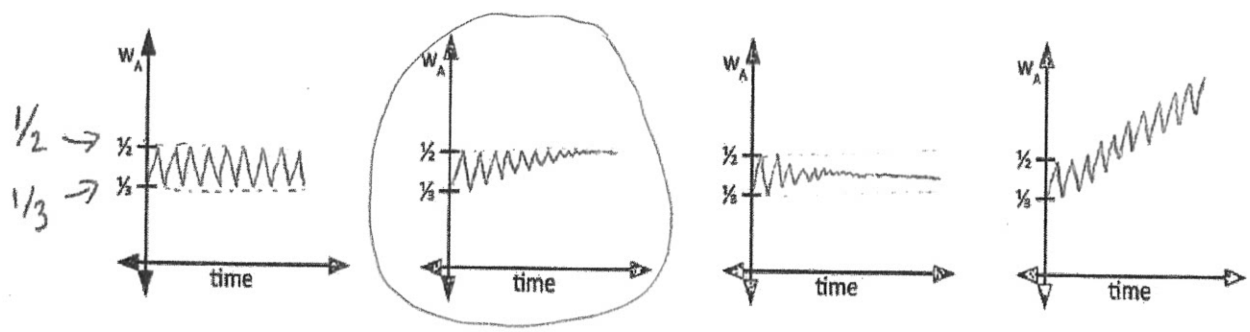
The table below shows the first four rounds of Adaboost, picking the classifier with the <u>**lowest error rate**</u> in each round. In case of a tie, h1 is chosen. Voting power is not shown.

	Round 1	Round 2	Round 3	Round 4
weight A	1/3	1/2	1/3	1/2
weight B	1/3	1/4	1/2	3/8
weight C	1/3	1/4	1/6	1/8
Error rate of h1	1/3	1/2	1/3	1/2
Error rate of h2	1/3	1/4	1/2	3/8
weak classifier chosen (h)	h1	h2	h1	h2
weak classifier error (ε)	1/3	1/4	1/3	3/8

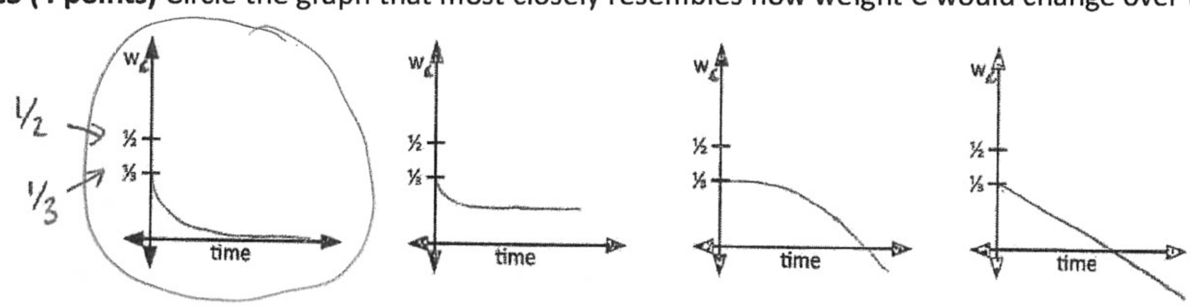
C1 (4 points) With this data, Adaboost can continue to run for many rounds. Which classifier will Adaboost pick in round 2015?

h1 **h2** **CAN'T BE DETERMINED**

C2 (4 points) Suppose you were to graph each weight over many rounds, starting from Round 1. Circle the graph that most closely resembles how weight A would change over time:



C3 (4 points) Circle the graph that most closely resembles how weight C would change over time:



Quiz 4, Problem 2: Bayes Nets (50 points)

Part A: Alice in Wonderland Syndrome (13 points)

Alice in Wonderland Syndrome (AIWS) is a real condition that can cause distorted perception of body size. You've developed a new test, the Rabbit Hole test, to test for AIWS. You conduct a survey on 100 people who are known to either have AIWS, or not have it. Their consolidated test results are shown below. Each number represents the number of people out of 100. Assume that this data is representative of the population.

	Has AIWS (+)	Does not have AIWS (–)	**Total**
Tested Positive (+)	23	17	**40**
Tested Negative (–)	22	38	**60**
Total	**45**	**55**	**100**

A1 (4 points) What is the sensitivity of this test? That is, what fraction of people with AIWS correctly test positive?

P(Tested positive | Has AIWS)

$$= \frac{23}{45}$$

A2 (4 points) What is the specificity of this test? That is, what fraction of people without AIWS correctly test negative?

P(Tested negative | Does not have AIWS)

$$= \frac{38}{55}$$

A3 (5 points) If a randomly selected person tests positive, what is the probability that they actually have AIWS?

P(Has AIWS | Tested positive)

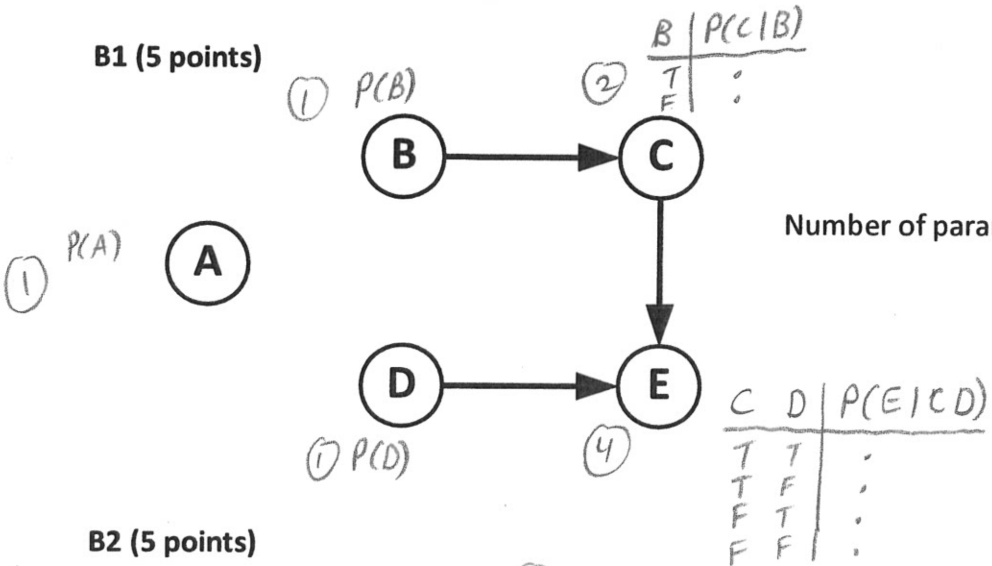
$$= \frac{23}{40}$$

Part B: A Few Bayesian Networks (15 points)

For each Bayes Net described below, answer this question:
Assuming all of the variables are boolean, how many parameters does the Bayes net have?
(The number of parameters is the total number of entries in all probability tables.)

For partial credit: **Next to each variable**, clearly write the number of parameters in that particular variable's probability table.

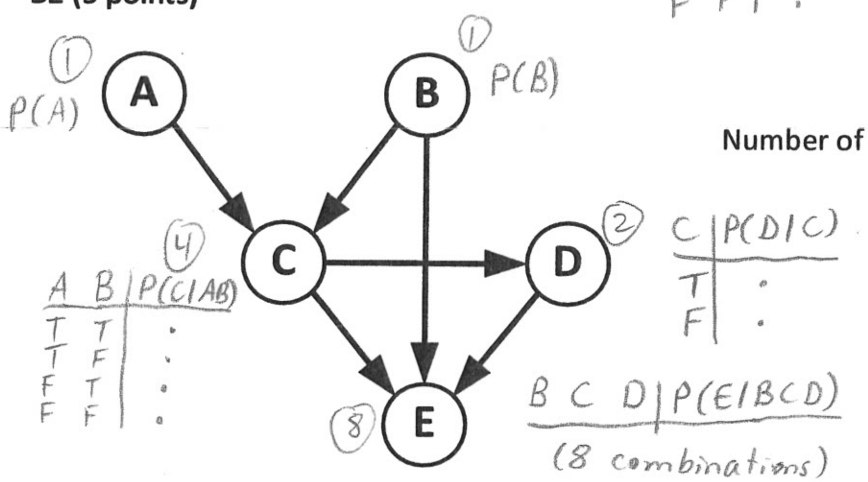
B1 (5 points)



1+1+2+1+4

Number of parameters = 9

B2 (5 points)

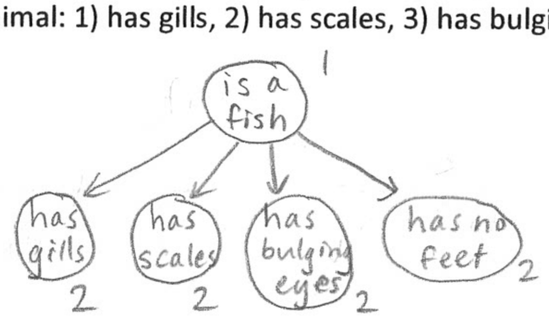


1+1+4+2+8

Number of parameters = 16

B3 (5 points) The Bayes net describing a Naïve Bayes Classification problem with boolean classification and four boolean features. (Recall that in Naïve Bayes Classification, features of a point are assumed to be independent given the point's classification.)

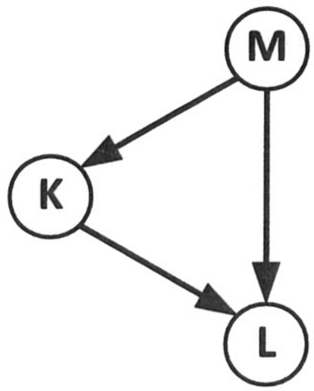
For example: "Use Naïve Bayes Classification decide whether a newly discovered species of animal is a fish (yes or no). The training data uses four boolean features to classify each animal: 1) has gills, 2) has scales, 3) has bulging eyes, 4) has no feet."



Number of parameters = 9

Part C: Looks familiar? (10 points)

Here is a Bayes net with 3 boolean variables and their associated probability tables. Each probability is represented by a lowercase variable.



P(M)
m

M	P(K|M)
T	x
F	y

K	M	P(L|KM)
T	T	a
T	F	b
F	T	c
F	F	d

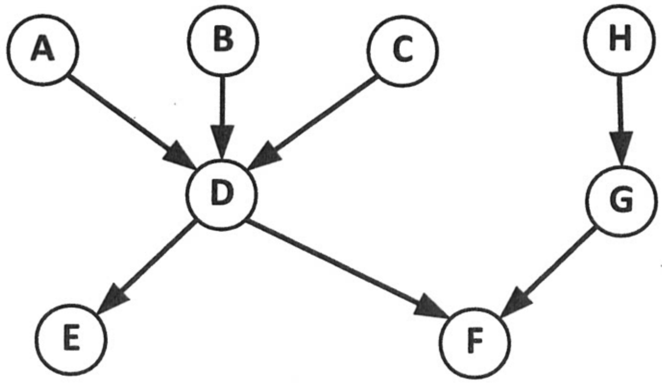
Write an expression for $P(K\,L)$ in terms of the probabilities specified in the Bayes net (m, x, y, a, b, c, d). Draw a box around your final answer. Show your work for partial credit.

$$P(KL) = \sum_M P(KLM)$$
$$= P(KLM) + P(KL\bar{M})$$
$$= P(L|KM)P(K|M)P(M) + P(L|K\bar{M})P(K|\bar{M})P(\bar{M})$$
$$= a \quad x \quad m + \quad b \quad y \quad (1-m)$$

$$\boxed{axm + by(1-m)}$$

Part D: Independence (Again!) (12 points)

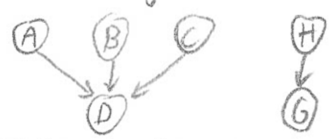
Here is a Bayes net with 8 variables, which are not necessarily boolean. Assume that the only independence statements that are true are the ones enforced by the shape of the network. Show your work for partial credit.



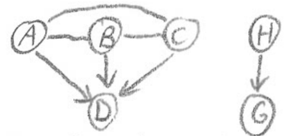
D1 (4 points) Is P(DG) = P(D) P(G)? (Circle one)

YES   NO

Ancestral graph:



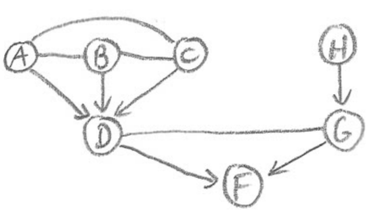
Moralized:



D2 (4 points) Are C and G conditionally independent, given F? (Circle one)

YES   NO

Moralized:

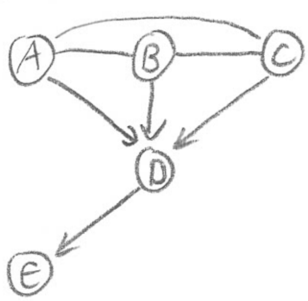


D3 (4 points) Is P(A|BDE) = P(A|BCDE)? (Circle one)

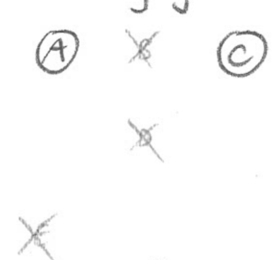
(Are A and C conditionally independent, given B, D, and E?)

YES   NO

Moralized:

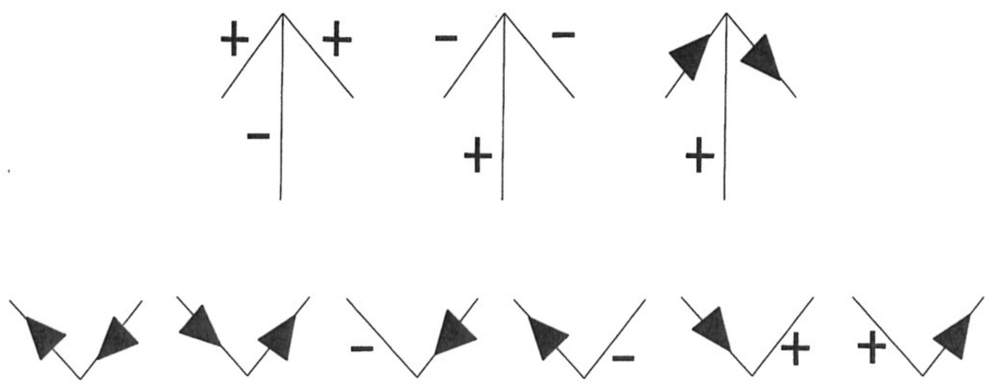


After disorienting and deleting givens:

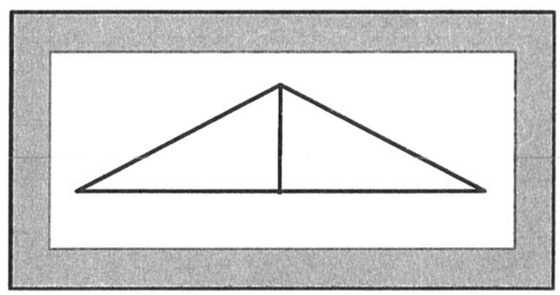


SRN, Quiz 2, Constraint propagation

You have attended Winston's lecture on line-drawing analysis and dutifully recorded the following table of legal junction arrangements for three-faced vertexes:



You see a triangular fragment of an object through a window:



Remember: junctions may be rotated however you like, but not mirrored.

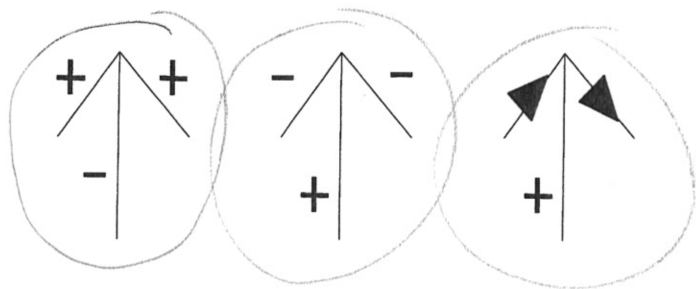
For **T junctions**, the two parts of the top of the T (the two collinear horizontal segments) must have the same label, but there is no constraint on the vertical segment.

Part A

You are to perform **pure constraint propagation** (that is, domain reduction before search). That is, you pile up all possible junction labels on each junction and then eliminate those junction labels that are not compatible with at least one junction label at each neighbor.
After performing pure constraint propagation, answer the following questions.

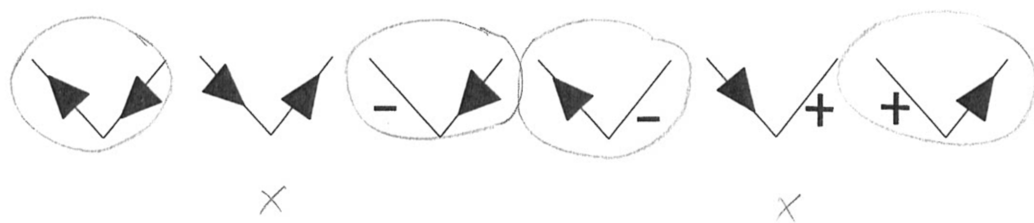
Of the 3 arrow junction labels in the junction library, how many are left at the arrow junction after constraint propagation? Write the number of junction labels and circle the ones that remain:

3



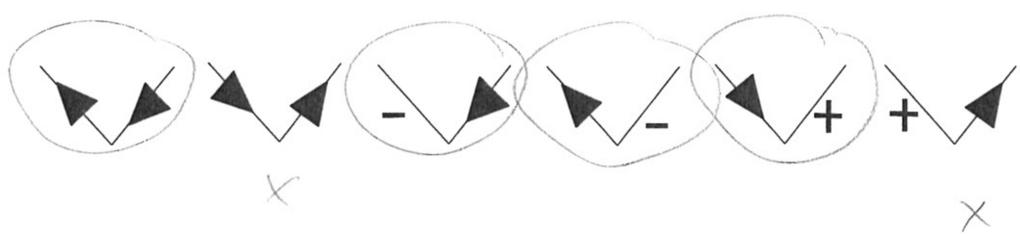
Of the 6 L junction labels in the junction library, how many are left at the left-side L junction after constraint propagation? Write the number of junction labels and circle the ones that remain:

4



Of the 6 L junction labels in the junction library, how many are left at the right-side L junction after constraint propagation? Write the number of junction labels and circle the ones that remain:

4



Part B

After performing pure constraint propagation, you decide to perform a search for consistent ways to label the drawing with just one label on each junction. How many such ways are there?

4

SRN, Quiz 3, Near-miss learning

President Reif is thrilled with the success of the new MicroMasters program and asks Professor Winston to develop a "PicoPhD Credential" in Computer Science.

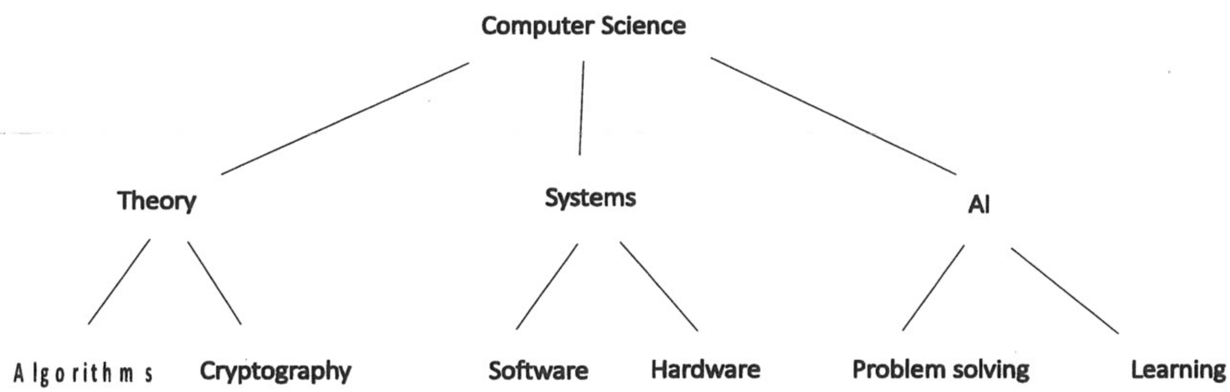
To earn a PicoPhD, each candidate fills out a form that determines if the candidate is qualified to enter a highest-bidder auction. The 100 who offer the most tuition will then talk on the telephone with a faculty member for ten minutes about life experience, and if judged to have contributed to knowledge (definition not yet determined), the candidate receives the credential and can say they have a PicoPhD Credential from MIT, usually shortened to PhD.

Professor Winston looks over a few of the first forms submitted and realizes that a near-miss learning exercise can develop a model of qualified candidates.

The **set** of all possible self-identified personality types is composed of **arrogant**, **confident**, and **timid**.

All candidates are either a Wizard or not (a Wizard is someone who has written a program used by at least one other person). That is, Wizard is a property with a **set** of two possible values, **yes** and **no.**

All candidates claim to be good at some aspect of computer science. Computer Science is divided into parts as follows:



The first six forms come in, with whether qualified or not, this way:

Candidate	Qualified?	Change	Heuristic	Personality	Wizard	Good at
1	Yes	---	---	Arrogant	Yes	Hardware
2	Yes			Arrogant	Yes	Software
3	No			Arrogant	Yes	Problem solving
4	Yes			Confident	Yes	Hardware
5	Yes			Confident	No	Hardware
6	No			Timid	No	Algorithms

Your job is to fill in all the missing information in the following table. Start by using Candidate 1 as a seed model, then train using the rest of the candidates in order.

For each example, fill out the table describing what you learned, such as *Must be Wizard* or *Personality doesn't matter.*

- Also include the heuristic(s) you used; choose from
    - require link (feature must be present)
    - forbid link (feature must not be present)
    - extend set (include more varieties)
    - drop link (include all possible varieties)
    - climb tree (generalize to a parent category)
- **Note:** Some examples may not teach you anything new. In that case, put "nothing" as the rule learned and "none" as the heuristic. Others may teach you more than one rule.

Candidate	Change(s)	Heuristic(s)
1		
2	Systems okay	Climb tree
3	nothing	none
4	Arrogant or confident	Extend set
5	Wizard doesn't matter	Drop link
6	nothing	none

Describe the candidates who qualify:

Arrogant or confident systems people

SRN, Quiz 4

You decide you want to do a UROP in computer vision and approach Winston. He mumbles something about using transition space to recognize actions, such as those that the Mind's Eye research program focused on:

Approach	Carry	Dig	Fall	Give	Hit
Lift	Push	Run	Touch	Arrive	Catch
Drop	Flee	Go	Hold	Move	Put down
Snatch	Turn	Attach	Chase	Enter	Fly
Hand	Kick	Open	Raise	Stop	Walk
Bounce	Close	Exchange	Follow	Haul	Jump
Pass	Receive	Take	Bury	Collide	Exit
Get	Have	Leave	Pick up	Replace	Throw

You decide to focus on the nine highlighted examples. With Winston you work up transition space descriptions of six of the nine on the board and take photographs, but alas, you neglected to label the examples with the corresponding action, and Winston wants a progress report by noon. For each description, select one of the nine highlighted action labels from those in he table above, noting that there are no duplicates. Use the following conventions:

A = Appear **D = Disappear**
$\uparrow$ = Increase **$\downarrow$= Decrease**
Δ = Change **$\neg$ = not**

P is a person; if more than one, an index number appears
O is an object; if more than one, an index number appears
H is the hand of a person; if more than one person, index number links person to hand
G is the ground

	Interval 1	Interval 2	Interval 3
Distance P O	$\downarrow$	D	$\neg$ A
Speed P	$\neg\Delta$	D	$\neg$ A
Speed O	$\neg$ A	$\neg$ A	$\neg$ A

Action: Collide

	Interval 1	Interval 2
Contact H O	$\neg$ A	A

Action: Touch

	Interval 1	Interval 2	Interval 3	Interval 4
Contact H1 O1	¬ D	¬ D	D	¬ A
Contact H1 O2	¬ A	A	¬ D	¬ D
Contact H2 O2	¬ D	¬ D	D	¬ A
Contact H2 O1	¬ A	A	¬ D	¬ D

Action: Exchange

	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5
Contact H O	¬ D	¬ D	¬ D	¬ D	D
Speed O	¬ A	A	¬ D	D	¬ A
Contact G O	¬ A	¬ A	¬ A	A	¬ D

Action: Put down

	Interval 1	Interval 2	Interval 3	Interval 4
Contact H1 O	¬ D	¬ D	D	¬ A
Contact H2 O	¬ A	A	¬ D	¬ D

Action: Give

	Interval 1	Interval 2	Interval 3	Interval 4	Interval 5
Speed O	¬ A	A	↑	D	¬ A
Contact H O	¬ D	D	¬ A	¬ A	¬ A
Contact G O	¬ A	¬ A	¬ A	A	¬ D

Action: Drop

Spiritual and Right-Now bonus points

Circle the one **best** answer for each of the following questions. There is **no penalty for wrong answers**, so it pays to guess in the absence of knowledge.

1. The Chinese Room idea is about:
    1. Why computers can only seem to be intelligent, not really be intelligent.
    2. Nonstop emergency programming with programmers subsisting on take-out Chinese food.
    3. Government agency efforts to use AI technology to defeat state-sponsored cyber crime.
    4. Where Chinese codes were cracked in World War II using Alan Turing's speculations about AI.
    5. The part of the Genesis system that models Chinese culture.

1

2. The Genesis story-understanding system:
    1. Uses big data and machine learning to take the "temperature" of thinking in various countries.
    2. Uses neural nets to acquire common-sense knowledge from Sesame Street videos.
    3. Finds concept patterns in its own story as it solves problems.
    4. Resolves ambiguities using constraint propagation.
    5. Cannot model irrational or delusional thinking.

3

3. According to Winston, self-organizing maps have been used to:
    1. Identify the most popular restaurants in large cities using Google search data.
    2. Build negotiation systems.
    3. Model behavior of rats as they learn mazes.
    4. Increase learning speed in deep neural nets.
    5. Determine communication patterns from corporate email records.

3

4. According to Winston, a newly proposed approach to action recognition involves:
    1. Surrounding actors with multiple cameras.
    2. Equipping actors with touch and pressure sensors.
    3. Use of mime groups to provide exaggerated-movement videos for training.
    4. Mounting cameras on infant foreheads to record what they see as their vision systems mature.
    5. Body alignment followed by empathetic guesses about sensory information.

5