6.034 Final Examination

16 December 2014

Name:

Email:

<u>Indicate</u> 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 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 want to know if our front-loading policy in 6.034 makes sense. So please indicate:

Number of subjects you are taking with a final, **not** including 6.034

Number of subjects you are taking with a term project or paper that requires substantial effort at or near the end of the semester

There are 38 pages on this exam, not including blank pages and tear-off sheets. 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 (50 points)

Part A: General questions (19 points)

A1 (15 points) For each of the following statements, circle the **single best answer**.

1. **True** / **False**: During backward chaining, at most one rule can match the current hypothesis.

2. **True** / **False**: During forward chaining, at most one rule can match per round.
multiple rules can match, only one can fire

3. **True** / **False**: During forward chaining, a rule may match in multiple rounds, but each rule can fire *at most* once. ↳ a rule can fire multiple times in separate rounds

4. **True** / **False**: In each round of forward chaining, the first rule that matches will fire.
It only fires if it generates a new assertion

5. **True** / **False**: In each round of forward chaining, at most one new assertion can be added to the list of assertions. ↳ (see 2015 Q1 solutions)

A2 (4 points) In general, which of the following conditions can cause backward chaining to short circuit? That is, which of these will cause the evaluation of subtrees to stop? (Circle **ALL** answers that apply)

A) A child of an AND subtree returns true. → still need to verify others are true

B) A child of an AND subtree returns false. → must be false

C) A child of an OR subtree returns true. → must be true

D) A child of an OR subtree returns false → need to check if others are true

E) The backward chainer cannot find a match for the hypothesis in the list of assertions. → need to check rules

F) The backward chainer cannot find a match for the hypothesis in the antecedent of any rule.

↳ this means it also didn't find a match in the assertions (since it checks assertions first)

Part B: Backward chaining (8 points)

Rules:

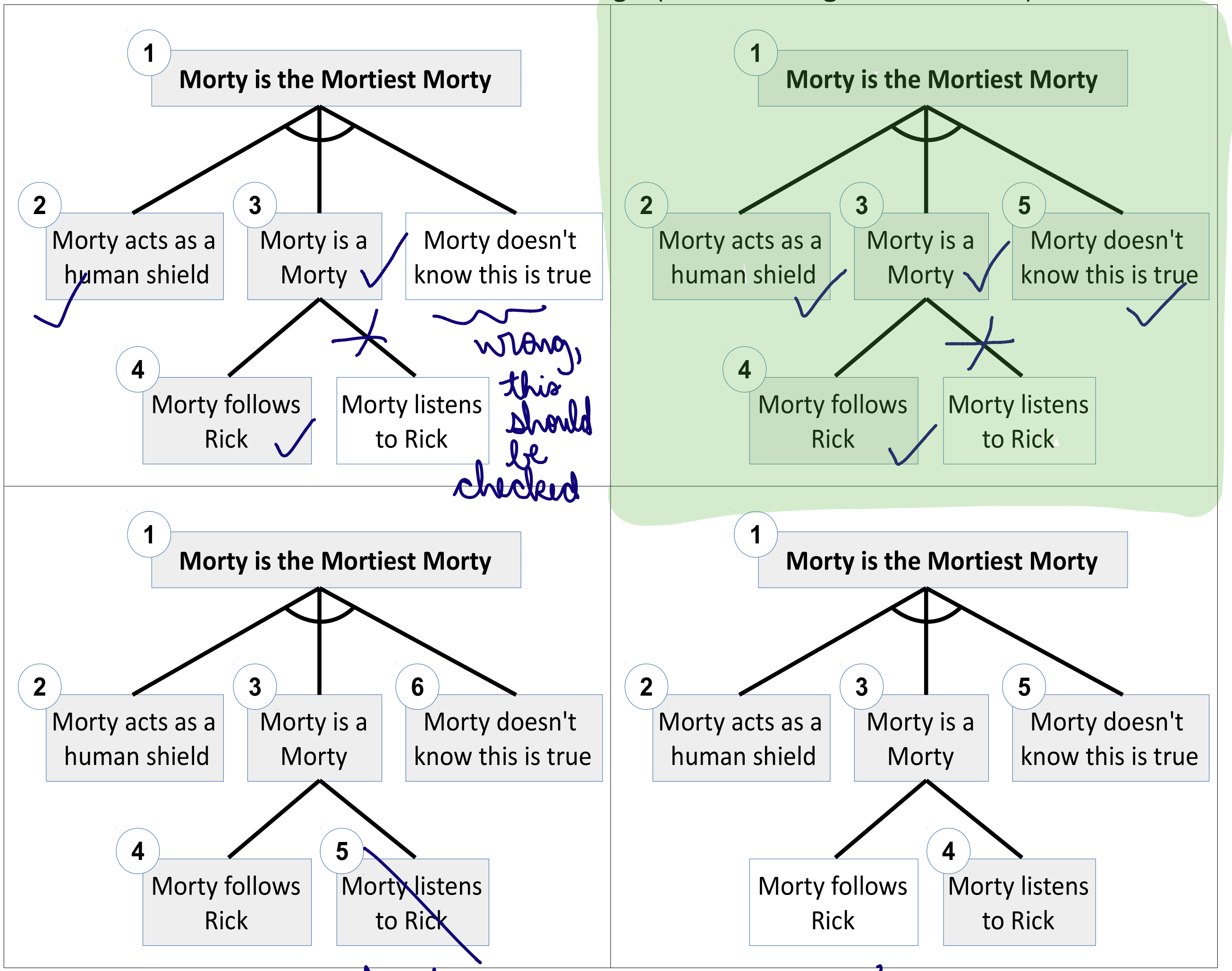
```
R0     IF    (OR '(?x) follows Rick',
                 '(?x) listens to Rick'),
       THEN '(?x) is a Morty'

R1     IF    (AND '(?y) acts as a human shield',
                  '(?y) is a Morty',
                  '(?y) doesn't know this is true'),
       THEN '(?y) is the Mortiest Morty'
```

Assertions:

A0: Morty follows Rick
A1: Morty listens to Rick
A2: Morty acts as a human shield
A3: Morty doesn't know this is true

B1 (7 points) Suppose you are performing backward chaining with this set of rules and assertions, starting from the hypothesis "Morty is the Mortiest Morty". Numbers indicate the order in which hypotheses are checked. Unnumbered nodes have not been explored. Which of the following trees would result at the end of backward chaining? (Circle the **single best** answer.)



Short circuited

wrong, left node is always checked 1st

B2 (1 point) Based on the results of backward chaining, is the hypothesis "Morty is the Mortiest Morty" true? (Circle one)

YES **NO** **CAN'T TELL**

Part C: Forward chaining (12 points)

Consider the following rule-based system:

Rules:

```
R0    IF   (OR '(?x) follows Rick',
               '(?x) listens to Rick'),
      THEN '(?x) is a Morty'

R1    IF   (AND '(?y) acts as a human shield',
                '(?y) is a Morty',
                '(?y) doesn't know this is true'),
      THEN '(?y) is the Mortiest Morty'
```

Assertions:
A0: Morty follows Rick
A1: Morty listens to Rick
A2: Morty acts as a human shield
A3: Morty doesn't know this is true

Suppose you are going to perform three rounds of forward chaining using these rules and assertions. (Hint: It may help to actually perform the forward chaining.)

A0 ~~B~~A1

C1 (3 points) In the first round, which rule(s) will match? (Circle one)

Only R0 **Only R1** **Both R0 and R1** **Neither R0 nor R1**

C2 (9 points) Which rule will fire in each round? Circle the **single best answer** in each case.

adds 'Morty is a Morty'

Round 1	**Only R0**	**Only R1**	**~~Both R0 and R1~~**	**Neither R0 nor R1**
Round 2	**~~Only R0~~**	**Only R1**	**~~Both R0 and R1~~**	**Neither R0 nor R1**
Round 3	**~~Only R0~~**	**~~Only R1~~**	**~~Both R0 and R1~~**	**Neither R0 nor R1**

only fires once (since there's nothing else that could be ?x or ?y)

only one rule fires at a time

Part D: More Forward Chaining (11 points)

These are independent questions about unrelated rule-based systems. Assume there are no DELETE statements in any system.

D1 (3 points) Suppose a rule matches in the first round of forward chaining and **does not contain any "NOT"s in its antecedent**. Is this rule guaranteed to match in every subsequent round? (Circle one)

YES **NO**

D2 (3 points) Suppose a rule matches in the first round of forward chaining and **contains a "NOT" in its antecedent**. Is this rule guaranteed to match in every subsequent round? (Circle one)

YES **NO**

It is possible that a new assertion is added to make the NOT false

D3 (5 points) Consider the following rule and possible sets of assertions. If you were to perform forward chaining, which of the sets would cause the statement `'Rick says "This is Rickdiculous!"'` to be **added** to the list of assertions? Circle **all** that apply.

```
R0    IF   AND('(?x) is lost',
                NOT '(?x) is going to find Morty')
      THEN '(?x) says "This is Rickdiculous!"'
```

Set 1: A0: Rick is lost

Set 2: A0: Rick is lost
A1: Rick is ***not*** going to find Morty

Set 3: A0: Rick is going to find Morty
A1: Rick is lost

Set 4: A0: Rick is lost
A1: Rick says "This is Rickdiculous!"

Set 5: A0: Rick says "This is Rickdiculous!"

Quiz 1, Problem 2: Search and Games (50 points)

Part A: Search Questions (27 points)

Here's a list of search algorithms and questions. Each question is worth **3 points**. For each question, list **ALL** of the search algorithms (A,B,C,D,E,F,G,H) that apply. If none of the search algorithms apply, write **NONE** instead. Some algorithms may be used more than once, and some may not be used at all.

A	B	C
A* search	Branch and bound (no heuristic, no extended set)	Depth-first search (with backtracking)
D	**E**	**F**
Branch and bound **with a heuristic** (no extended set)	Branch and bound **with an extended set** (no heuristic)	*Breadth*-first search
G	**H**	
Best-first search	Hill-climbing (with backtracking)	

① Which of these algorithms add children to the front of the agenda like a stack (without sorting the agenda)?

C, H, sorts children, not the agenda

② Which of these algorithms add children to the back of the agenda like a queue (without sorting the agenda)?

F

③ Which of these algorithms will need to know the weights of the edges in the graph?

A, B, D, E

④ Which of these algorithms will need a heuristic estimate of remaining distance to the goal?

A, D, G, H

⑤ Which of these algorithms will always find the shortest path to the goal? (If an algorithm uses a heuristic, do not assume that the heuristic is admissible or consistent.)

BE

⑥ Which of these algorithms can use a consistent heuristic to always find the shortest path to the goal (if one exists)? Think carefully.

AD (consistency implies admissibility)

⑦ Which of these algorithms have a built-in limit on the size of the agenda?

None (Beam Search does, but it is not listed)

⑧ Which of these algorithms use some kind of cost function to determine which path to extend next?

heuristic, path length, or heuristic + path length

ABDEGH

⑨ ~~Which of these algorithms may continue searching even after finding a path to the goal?~~

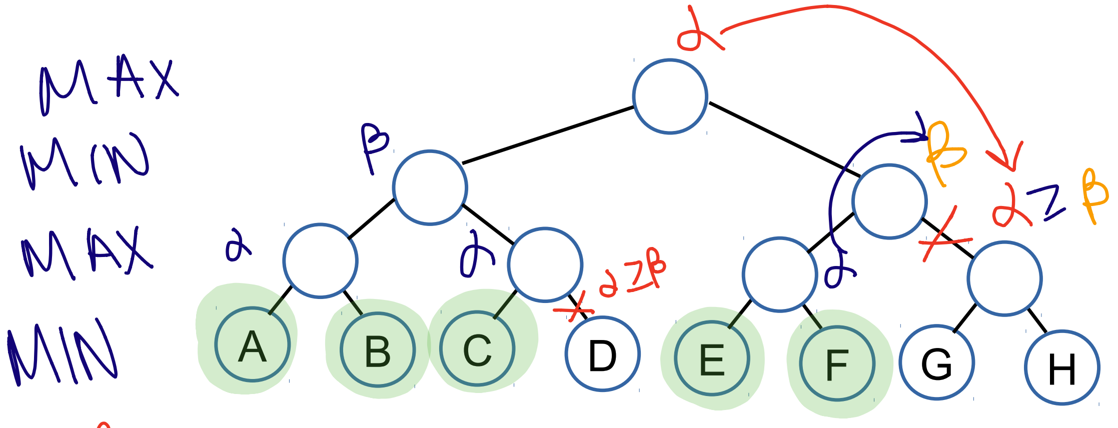
ambiguous question

Search

- Non-optimal
  - Blind
    - DFS
    - BFS
  - Informed
    - Hill Climbing
    - Beam
    - Best First
- Optimal = Finds shortest path
  - Branch & Bound
    - B&B + set
    - B&B + heuristic → must be admissible
    - A* → must be consistent
  - British Museum

Part B (14 points)

In this problem, you will evaluate how alpha-beta pruning performs in the best case and in the worst case.



(still applies if MIN is 1st)

B1 (7 points) Suppose you perform alpha-beta pruning on the above game tree. List ***all*** of the leaf nodes you would have to **statically evaluate** in the **best** case, i.e. if the static values in the tree cause alpha-beta to prune the greatest possible number of nodes.

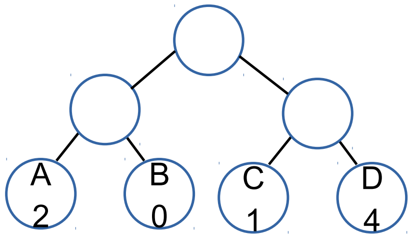
A BCEF

B2 (7 points) Suppose you perform alpha-beta pruning on the above game tree. List ***all*** of the leaf nodes you would have to **statically evaluate** in the **worst** case, i.e. if the static values in the tree cause alpha-beta to prune the least possible number of nodes.

ABCDEFGH

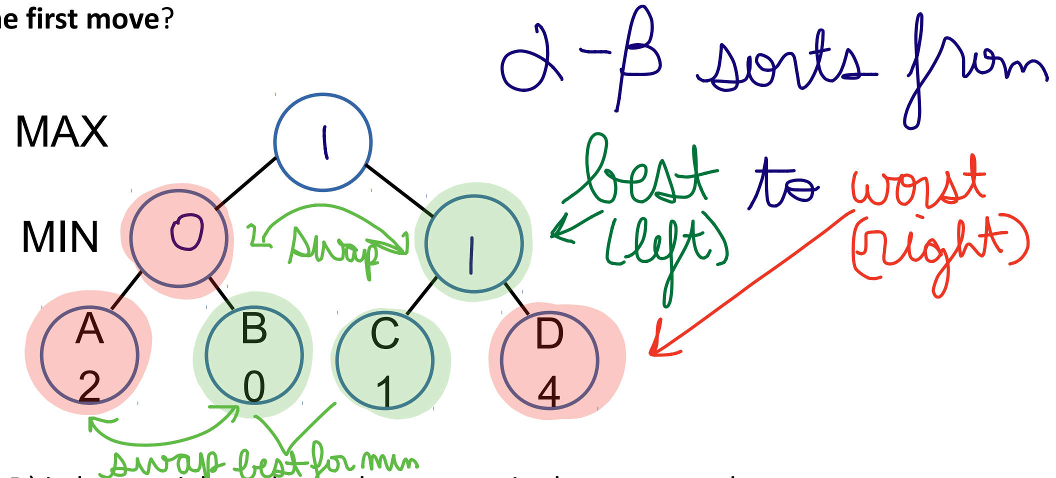
Part C (9 points)

While performing alpha-beta pruning with progressive deepening, you generate this game tree that looks <u>two</u> moves ahead.



In the next round of progressive deepening, you will look <u>three</u> moves ahead.

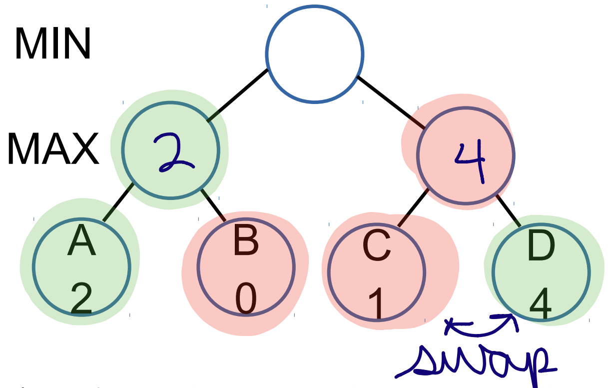
C1 (5 points) How should you reorder this tree to maximize the possibility of pruning in the next round **if MAX will make the first move**?



List the leaf nodes (A, B, C, D) in left-to-right order as they appear in the rearranged tree.

CDBA

C2 (4 points) How should you reorder this tree to maximize the possibility of pruning in the next round **if MIN will make the first move**?



List the leaf nodes (A, B, C, D) in left-to-right order as they appear in the rearranged tree.

ABDC

Quiz 2, Problem 1: Constraint satisfaction problems (50 points)

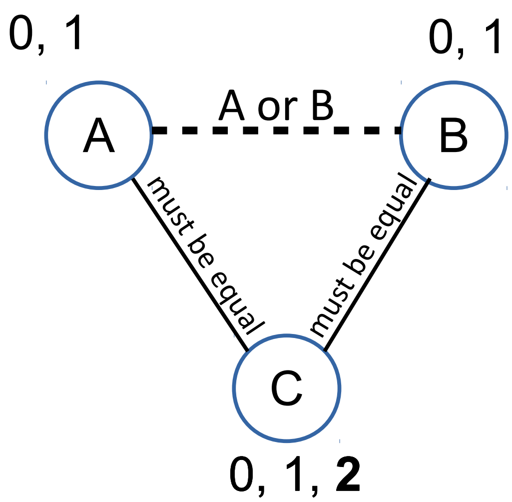
Here is a constraint satisfaction problem with three variables **A**, **B**, **C**. Their initial domains are shown here:

Variable	Domain		
A	**0**	**1**	
B	**0**	**1**	
C	**0**	**1**	**2**

The variables have the following constraints:

1. A and C must be equal.
2. B and C must be equal.
3. Either A must be 1, or B must be 1, or both. They can't both be zero.

To help you visualize these constraints, we have depicted them in the constraint graph here.



Note: For your convenience, a copy of this constraint graph is provided on a tear-off sheet after the last page of the final.

Part A: Value Assignments (5 points)

product of domain sizes

$2 \cdot 2 \cdot 3$

Part A1. How many possible assignments of values to variables are there?

(ignoring constraints)

YOUR ANSWER: 12

Part A2. How many of those possible assignments satisfy all the constraints – that is, how many different solutions does this problem have? (You may use your intuition, or figure the answer out using the later parts of the problem.)

YOUR ANSWER: 1

A=1
B=1
C=1

Part B: First attempt—depth-first search + forward checking (20 points)

IMPORTANT: Don't reduce domains in advance.

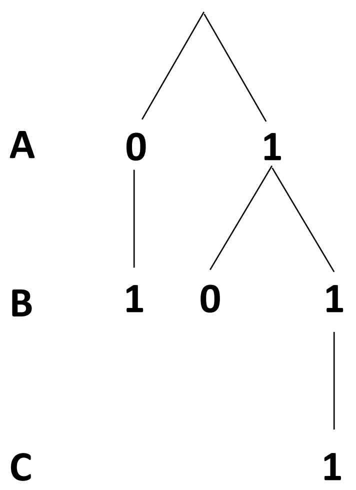
Use **depth-first search with forward checking** (no propagation) to find a consistent assignment of values to variables. Assign variables in the order A, B, C. Break ties by assigning **lower values first**. The search tree has been drawn for you, so all you need to do is fill out the worksheet below.

Fill out this worksheet. 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.

If you add several variables to your propagation queue at once, break ties by adding variables to your propagation queue **in alphabetical order**.

Search Tree



	Var assigned or de-queued	List all values just ELIMINATED from neighboring variables	Back track
1	A=0	B≠0, C≠1,2	☐
2	B=1	C≠0	☑
3	A=1	C≠0,2	☐
4	B=0	C≠1	☑
5	B=1	None	☐
6	C=1	None	☐
7			☐
8			☐

Example row showing an assigned variable

ex	X = 3	Y ≠ 3, 4 Z ≠ 3	☐

Example row showing a de-queued (propagated) variable

ex	X	W ≠ 1, 4	☐

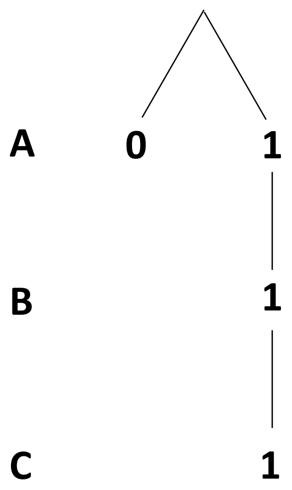
Part C: Second attempt—propagation through singletons (20 points)

IMPORTANT: Don't reduce domains in advance.

Now repeat the process, except this time use **depth first search with forward checking and propagation through domains reduced to size 1** to find a consistent assignment of values to variables. Assign variables in the order A, B, C. Break ties by assigning **lower values first**. The search tree has been drawn for you, so all you need to do is fill out the worksheet below. (There may be more rows than you need.)

★★★ **Note:** If you would add more than one variable to the propagation queue at the same time, add them to the queue in alphabetical order. ★★★

Search Tree



	Var assigned or de-queued	List all values just ELIMINATED from neighboring variables	Back track
1	A=0	B≠0, C≠1,2	☐
2	B	C≠0	☐ ✓
3	A=1	C≠0,2	☐
4	C	B≠0	☐
5	B	None	☐
6	B=1	None	☐
7	C=1	None	☐
8			☐

Example row showing an assigned variable

ex	X = 3	Y ≠ 3, 4 Z ≠ 3	☐

Example row showing a de-queued (propagated) variable

ex	X	W ≠ 1, 4	☐

Part D: Domain Reduction (5 points)

Part D1. Suppose you reduce the domains of the variables before starting search. What would the resulting domains of the variables be? (For each variable, **circle all values** that would still be **in the variable's domain**. If the domain would be empty after domain reduction, circle **EMPTY** instead.)

Propogation Queue

~~A~~
~~B~~
~~C~~

C must equal A

A's domain:	0	1		**EMPTY**
B's domain:	0	1		**EMPTY**
C's domain:	0	1	2	**EMPTY**

A	$C \neq 2$
B	None
C	None

Part D2. Is the following statement true or false?

> Domain reduction guarantees that you won't have to do any backtracking during search, at the cost of performing more computation before search.

Circle your answer:

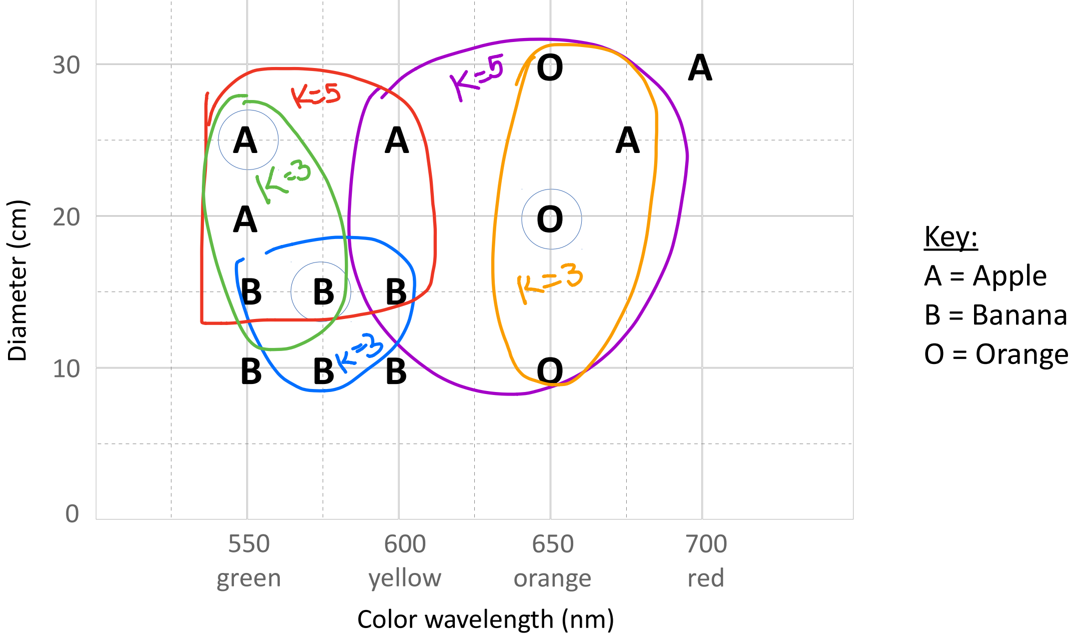
True **False**

(**Hint:** You may consider this problem, for example.)

Quiz 2, Problem 2: Nearest neighbors & Identification trees (50 points)

Part A: Comparing Apples to Oranges (16 points)

The 6.034 TAs have discovered a mysterious yellowish fruit on the table outside Prof. Winston's office. Before offering the fruit to Prof. Winston as a gift, Elisa points out that they should identify it using k-nearest neighbors. Dylan first makes a graph of known fruits, using each fruit's color and diameter. The resulting training data is as follows:



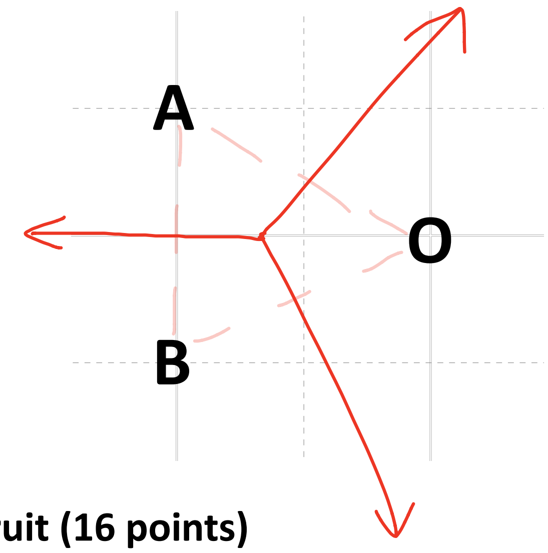
A1 (8 points) Malcom wants to use cross-validation to find the best value of k for k-nearest neighbors classification. For each circled point, classify that point as if it were not part of the training data. In the table below, **fill in the classification** (**A**, **B**, **O**, or **CAN'T TELL**) of each circled training point for k = 1, 3, and 5. The first row has been filled in for you.

	A (550nm, 25cm)	B (575nm, 15cm)	O (650nm, 20cm)
k=1	A	B	A
k=3	A	B	O
k=5	B	B	Can't Tell

A2 (2 points) Based on your cross-validation in part A1, which is the best value of k?

k = 1 **k = 3** **k = 5**

A3 (6 points) Regardless of the results of Malcom's cross-validation, Jessica decides to use k=1 so she can draw decision boundaries. On the zoomed-in section of the graph below, **draw the decision boundaries found by 1-nearest neighbors**. Consider only the three training points pictured. The exact location of each point is where gridlines intersect.



always draw the lines to ∞
draw to end of graph
and/or add arrowheads

Part B: No More Fruit (16 points)

The other TAs are tired of classifying fruits, so they've moved on to classifying other things. This part consists of a series of independent problems.

B1 (2 points) Pedro has some training data, which he classifies using both k-nearest neighbors and identification trees. He then realizes that some of the features don't contain useful information. Which performed better in terms of ignoring the useless tests? (Circle one)

Identification Trees   **k-Nearest Neighbors**

B2 (4 points) Siyao is gathering data about ice cream. Calculate the disorder of the feature test "Flavor" in the data below. Your answer may contain logarithms. Show your work for partial credit.

Classification	Flavor
Good	Chocolate
Good	Chocolate
Good	Vanilla
Good	Strawberry
Bad	Strawberry

homogenous groups are always disorder 0

from inspection the disorder of this branch is 1

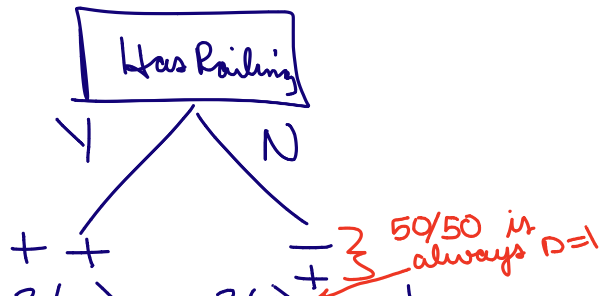
$$= \frac{2}{5}\left(\cancel{-\frac{2}{2}\lg\frac{2}{2}}\right) + \frac{1}{5}\left(\cancel{-\frac{1}{1}\lg\frac{1}{1}}\right) + \frac{2}{5}\left(-\frac{1}{2}\lg\frac{1}{2} - \frac{1}{2}\lg\frac{1}{2}\right) = \frac{2}{5}$$

Disorder (Flavor) = $\boxed{\frac{2}{5}}$

B3 (4 points) Duncan is classifying locations as Safe or Dangerous. One feature he considers is "Has Railing." Duncan has forgotten the classification of one of his training points, Location #3. Given that **"Has Railing" has a disorder of 0.5**, what must have been the classification of Location #3? (Circle one)

Hint 1: It may be helpful to draw the decision stump for the "Has Railing" feature.
Hint 2: Or it may be helpful to calculate what the disorder of the feature test "Has Railing" would be for each classification (Safe or Dangerous) of Location #3.

Location #	Classification	Has Railing
1	Safe	Yes
2	Safe	Yes
3	??	No
4	Dangerous	No



$\frac{2}{4}(\emptyset) + \frac{2}{4}(1) = \frac{1}{2}$

SAFE **DANGEROUS** **CAN'T TELL**

(you can do this without math) → this would give ∅ disorder (both branches w/ homogenous sets)

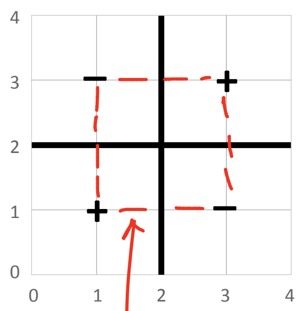
B4 (6 points) Josh is investigating some claims about identification trees and k-nearest neighbors. Circle the one best answer (True or False) for each claim.

1. After a feature test is used in an identification tree, it will always have a disorder of 0. **TRUE** **FALSE**

2. When it is impossible to perfectly classify the training data using identification trees, you can asymptotically improve the error rate by repeatedly testing whether a point has a particular feature. **TRUE** **FALSE**

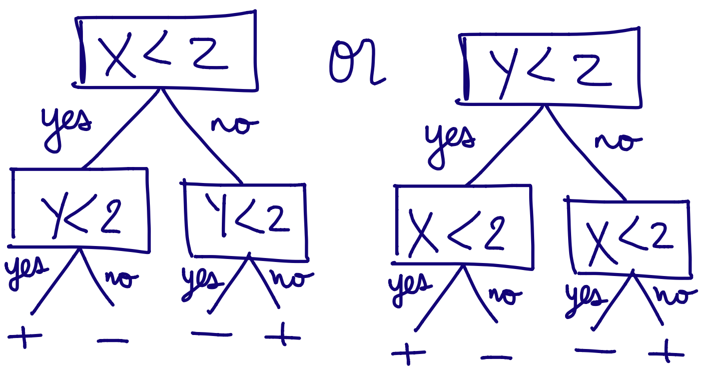
Testing again will yield the same result, so it won't improve the results

3. Given the following four training points in two classifications (+ and -), identification trees and 1-nearest neighbors would both draw these boundaries: **TRUE** **FALSE**



nearest neighbors boundaries are all perpendicular bisectors ✓

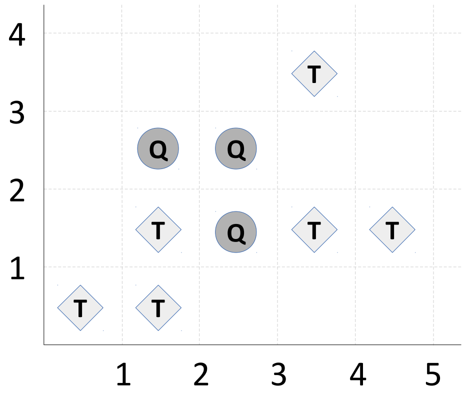
ID trees:



Part C: Unshredding (18 points)

Your friend Jack, whose thesis project involves reconstructing shredded documents, has the found the pieces of what appears to be a 6.034 quiz solution! However, there are multiple ways that the pieces fit together, and Jack doesn't know which possible solutions are correct.

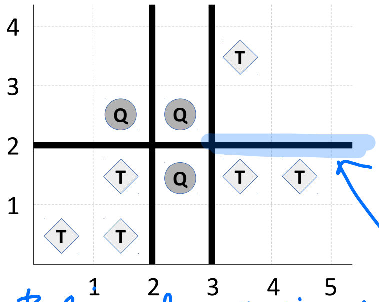
The original problem was to draw decision boundaries on the graph at right that perfectly classify the training data into two classifications (**Q** and **T**) using a greedy disorder-minimizing identification tree.



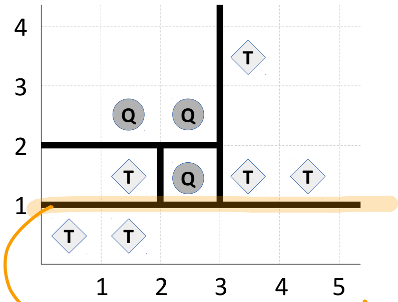
Jack has pieced together the following possible solutions. For each one, circle YES or NO to answer the question "**Does this graph depict disorder-minimizing identification tree boundaries?**"

YES

NO



inefficient & obviously not disorder minimizing i.e this boundary



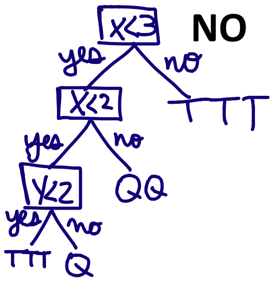
YES

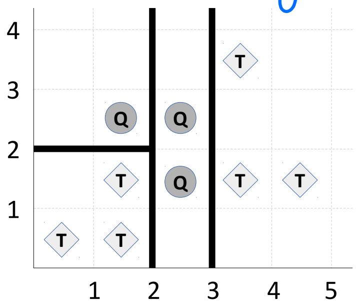
NO

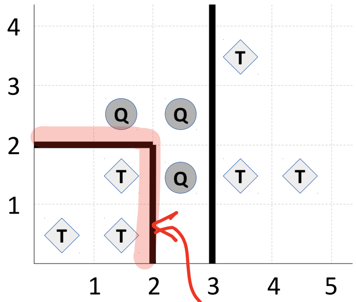
Y<1 is the first test, but it is not the test w/ the least disorder

YES

NO







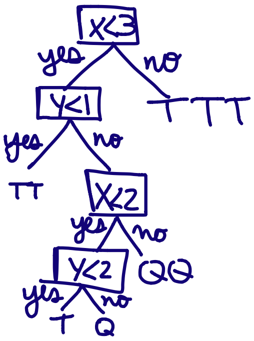
YES

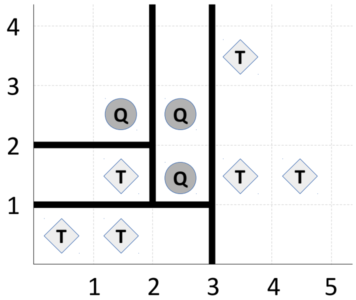
NO

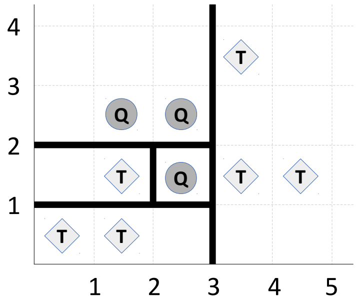
You cannot generate this with an ID tree that tests only 1 feature at a time

YES

NO

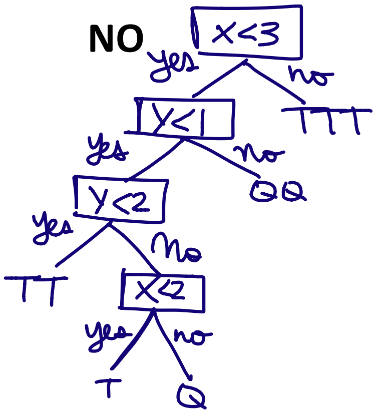




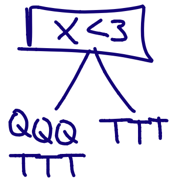


YES

NO



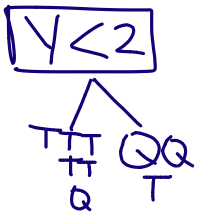
Test 1



$D = \frac{6}{9}(1) + \frac{3}{9}(0) = \frac{2}{3}$

$D = \frac{2}{9}(0) - \frac{7}{9}\left(\frac{3}{7}\lg\frac{3}{7} + \frac{4}{7}\lg\frac{4}{7}\right)$

$= .766$



$D = -\frac{6}{9}\left(\frac{1}{6}\lg\frac{1}{6} + \frac{5}{6}\lg\frac{5}{6}\right) - \frac{3}{9}\left(\frac{2}{3}\lg\frac{2}{3} + \frac{1}{3}\lg\frac{1}{3}\right)$

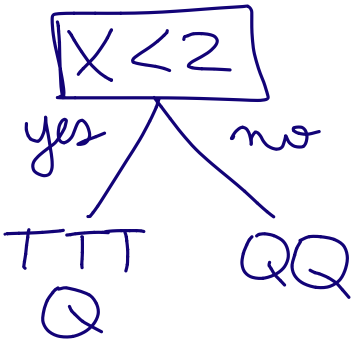
$= .739$

Disorder minimizing

↳ it is possible to qualitatively determine that $\boxed{X<3}$ is disorder minimizing

---

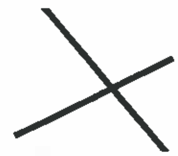
Test 2 (after $X<3$)



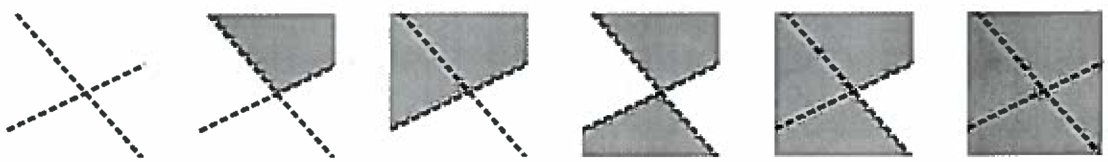
all 3 have the same disorder, pick any for test

Quiz 3, Problem 1: Neural networks (40 points)

Part A: Cutting the Plane (20 points)

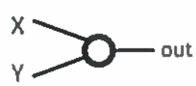


Neural nets with different structures are capable of recognizing different output patterns. Consider the different ways of cutting the plane into four sections using two intersecting diagonal lines. A shaded section indicates that all the points in that area are classified as 1, while an unshaded section indicates that all the points in that area are classified as 0.
Ignoring rotations, there are six shading possibilities, as shown here.



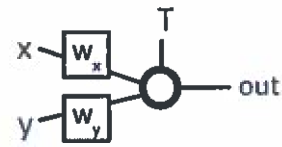
A	B	C	D	E	F
nothing shaded	one shaded region	two shaded neighboring regions	two shaded opposite regions	three shaded regions	everything shaded
0	P and Q	P	P xor Q	P or Q	1

For each of the following neural nets, list all of the above shadings (A, B, C, D, E, F) that the neural net can produce. If a neural net can produce none of these shadings, write NONE instead. Be careful, in particular for answer choice A. Note that we are asking only if the shadings that can be produced, not the lines, because it might be possible to produce the shading but not the lines.

Neural net	List ALL shadings (A, B, C, D, E, F) the neural net can produce; if none, write NONE instead.
≤ 1 line 	A, C, F
≤ 1 horizontal line drawn ≤ 1 vertical line drawn 	A, F
≤ 2 lines drawn combine with "basic" logic 	A, B, C, E, F
≤ 2 lines drawn combine with "more complicated" logic 	A, B, C, D, E, F

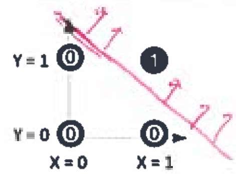
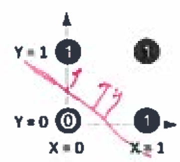
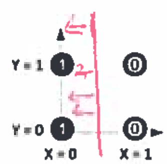
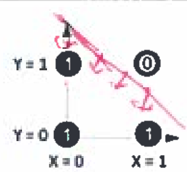
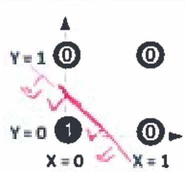
Part B: Logic (20 points)

In this problem, you are trying to build neural nets that can implement certain logic functions. For each logic function, your neural net must be <u>**a single neuron with two inputs**</u>. That is, the output of the net must be:



$$out = \begin{Bmatrix} 1, & if \;\; w_x \cdot x + w_y \cdot y \ge T \\ 0, & otherwise \end{Bmatrix}$$

For each row in the table, **circle values for w_x, w_y, and T** that implement the desired behavior.

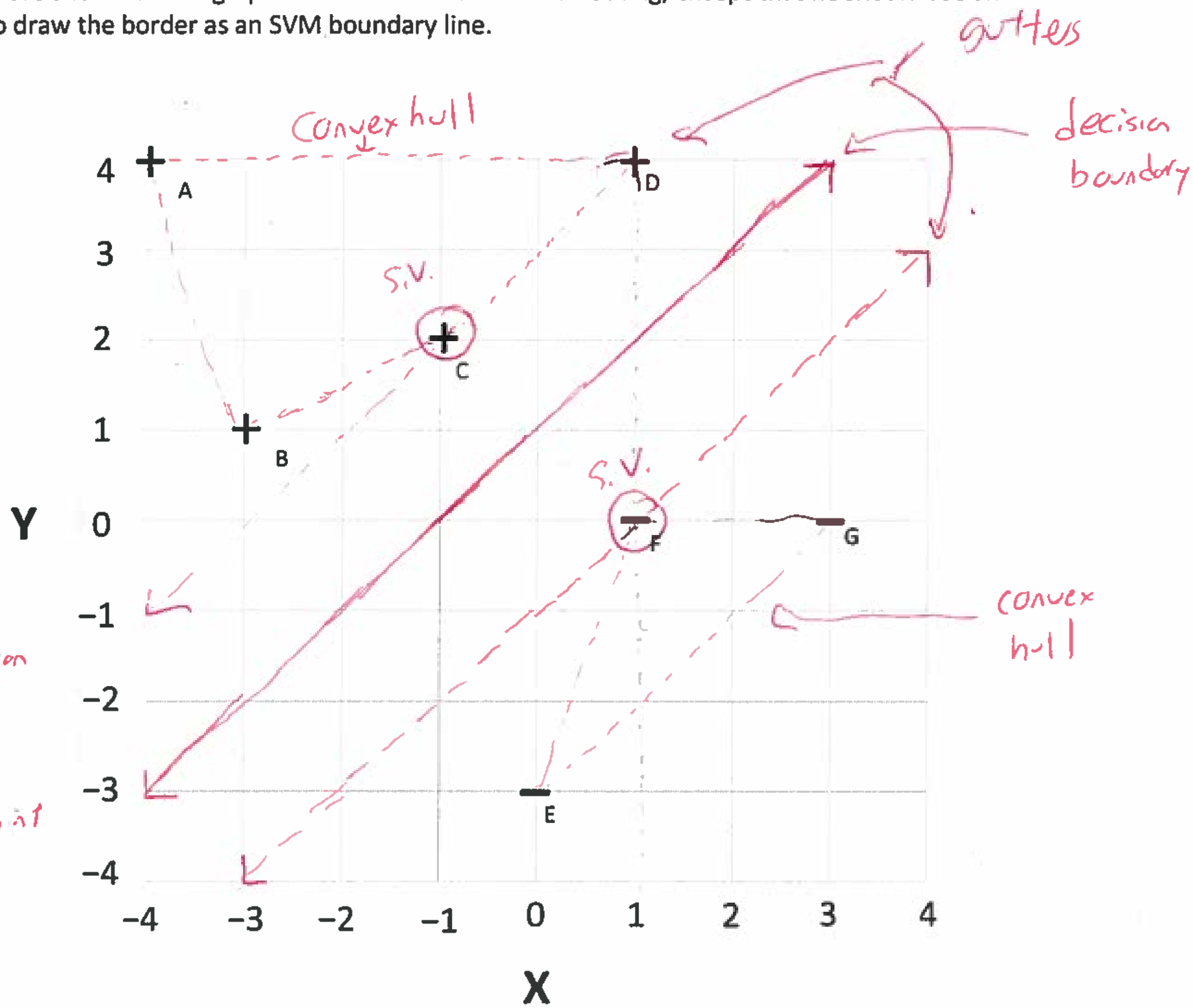
Logic Function	Graph of behavior (shaded = output of 1, unshaded = output of 0)	Weights and threshold
AND(X, Y) $2X+2Y \ge 3$		w_X: -2 0 2 w_Y: -2 0 2 T: -3 -1 1 3
OR(X, Y) $2X+2Y \ge 1$		w_X: -2 0 2 w_Y: -2 0 2 T: -3 -1 1 3
NOT(X) $X \le 0.5$ $-2X \ge -1$		w_X: -2 0 2 w_Y: -2 0 2 T: -3 -1 1 3
NAND(X, Y) $2X+2Y \le 3$ $-2X-2Y \ge -3$		w_X: -2 0 2 w_Y: -2 0 2 T: -3 -1 1 3
NOR(X, Y) $2X+2Y \le 1$ $-2X-2Y \ge -1$		w_X: -2 0 2 w_Y: -2 0 2 T: -3 -1 1 3

Quiz 3, Problem 2: Support Vector Machines (60 points)

Part A: Border beyond the wall (30 points)

The men of the Night's Watch are trying to map the regions beyond the Wall and would like to mark the border between White Walker and Free Folk territory. White Walkers (+) and Free Folk (-) are shown on the graph below. John Snow knows nothing, except that he should use an SVM to draw the border as an SVM boundary line.

So, we have a point and line. But D is not really a support vector as removing it will not change the decision boundary. So, really we have a point-point situation



Part A1 (10 points): On the diagram above,

- Draw the SVM boundary as a **solid line**
- Draw the gutters as **dotted lines**
- **Circle** the support vectors

Part A2 (8 points)

Find the values of $\vec{w}$ and b that correspond to the SVM you drew in Part A1.

$\vec{w}$ = [-0.5 0.5]

b = -0.5

Show your work for partial credit.

M.W. $= 2\sqrt{2}$, so $\|\vec{w}\| = \frac{1}{\sqrt{2}}$

Decision Bandary $y = x + 1$ or $[1\ -1]\begin{bmatrix} x \\ y \end{bmatrix} + 1$

So, we need to scale such that point C $(-1, 2)$ evaluates to 1, ~~and that~~ which makes $\vec{w}$ points to plus side.

$c\left([1\ -1]\begin{bmatrix} -1 \\ 2 \end{bmatrix} + 1\right) = 1$

or $c(-2) = 1$, so $c = -0.5$.

So, ~~$\vec{W} = -0.5$~~

Decision Boundary is described as follows:

~~$0.5[1\ -1]$~~ $\vec{w} = -0.5[1\ -1]$

$= [-0.5\ 0.5]$

$\vec{b} = -0.5$

Part A3 (6 points)

Circle the classification the SVM would assign to a new point $\langle x, y \rangle = \langle 4, 4 \rangle$.

2 ways:
<4,4> lies below decision boundary, or
$\text{sign}\left([-0.5\ 0.5]\begin{bmatrix}4\\4\end{bmatrix}-0.5\right)$
$= \text{sign}(-0.5)$
$= -$

White Walker (+)   Free Folk (-)   Can't be determined

Circle the classification the SVM would assign to a new point $\langle x, y \rangle$ given: $\vec{w} \cdot \langle x, y \rangle = 1$

So, $\text{sign}(\vec{w} \cdot \vec{x} + b)$
$= \text{sign}(1 - 0.5) =$
$\text{sign}(0.5)$
$= +$

White Walker (+)   Free Folk (-)   Can't be determined

Part A4 (6 points)

For each training point, indicate whether its alpha value is POSITIVE, NEGATIVE, or ZERO. Circle the best answer in each case.

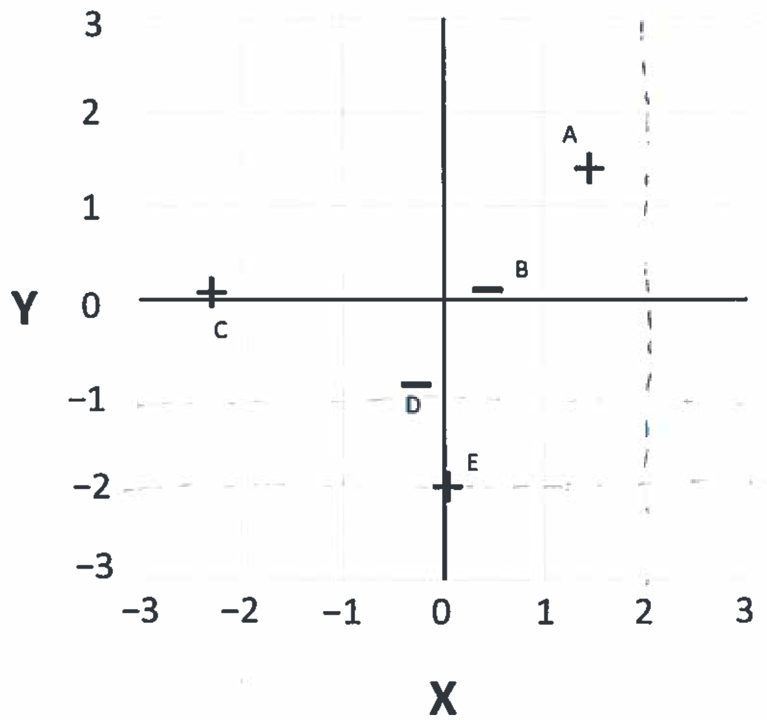
α_A	POSITIVE	NEGATIVE	ZERO
α_B	POSITIVE	NEGATIVE	ZERO
α_C	POSITIVE	NEGATIVE	ZERO
α_D	POSITIVE	NEGATIVE	ZERO
α_E	POSITIVE	NEGATIVE	ZERO
α_F	POSITIVE	NEGATIVE	ZERO
α_G	POSITIVE	NEGATIVE	ZERO

A, B, D, E, G are not support vectors so their α's are 0.

C, F are support vectors, so we have their α's are positive.

Part B: A Lannister always separates their data (30 points)

Tyrion is trying to figure out if the other characters are honest (+) or sneaky (-). He is having trouble because the data he has is not linearly separable. A graph of Tyrion's data is shown below.



Part B1 (5 points)

Consider each of the strategies below. Circle **ALL** of the strategies that will enable Tyrion to perfectly classify this entire training dataset using a support vector machine:

1. Reduce tolerance by minimizing, rather than maximizing, the margin width.
2. Reduce training error by eliminating a redundant feature.
3. Separate the space by introducing a third feature, Z.
4. Separate the space by introducing a nonlinear kernel.
5. Perform cross-validation to identify outliers.

Part B2 (5 points) Instead, Tyrion decides to use kernels and transforms to separate his data. Which kernels can perfectly separate Tyrion's data? (Circle **all** that apply)

Linear   Polynomial (Quadratic)   Radial Basis

Part B3 (10 points) Tyrion considers using the following transform:

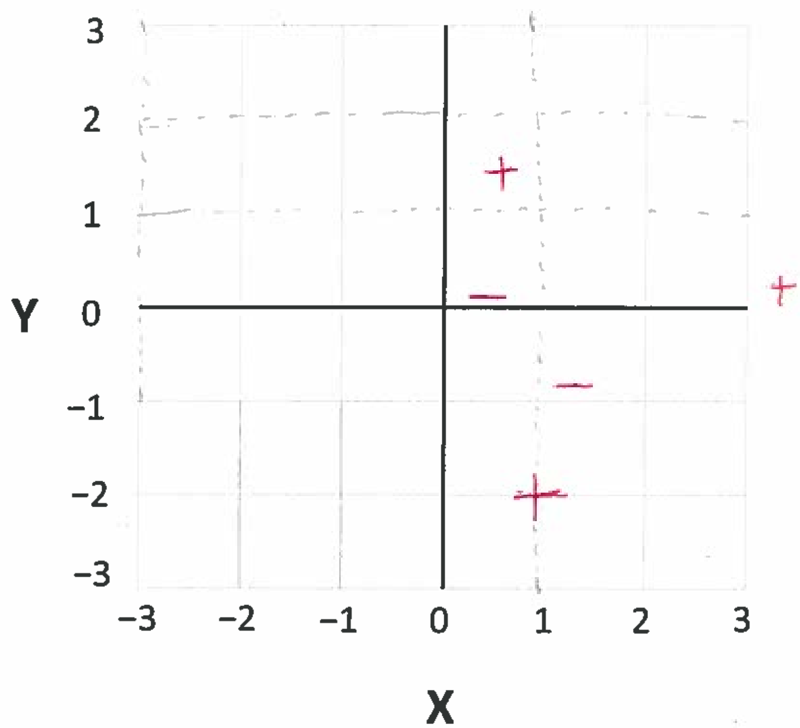
$$\varphi(\langle x, y \rangle) = \langle |x - 1|, y \rangle$$

Will this transform make the data linearly separable? (Circle one)

Yes   No

Note: For your convenience, this blank plot is provided to optionally show your work.

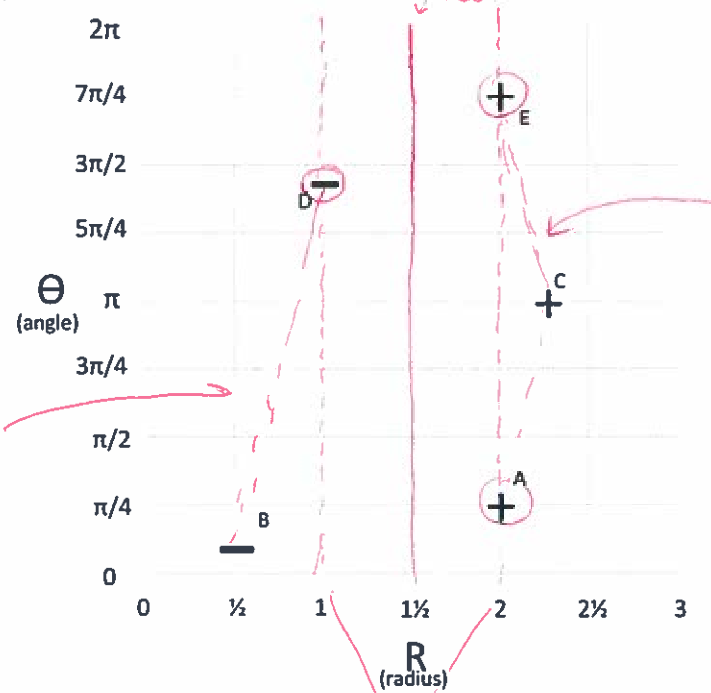
Translated data is now!



Part B4 (10 points)

In the end, Tyrion decides to use a **polar transform**, ($\varphi(\langle x, y\rangle) = \langle r, \theta\rangle$) resulting in the following transformed space:

decision boundary



convex hull

point-line

convex hull

gutters

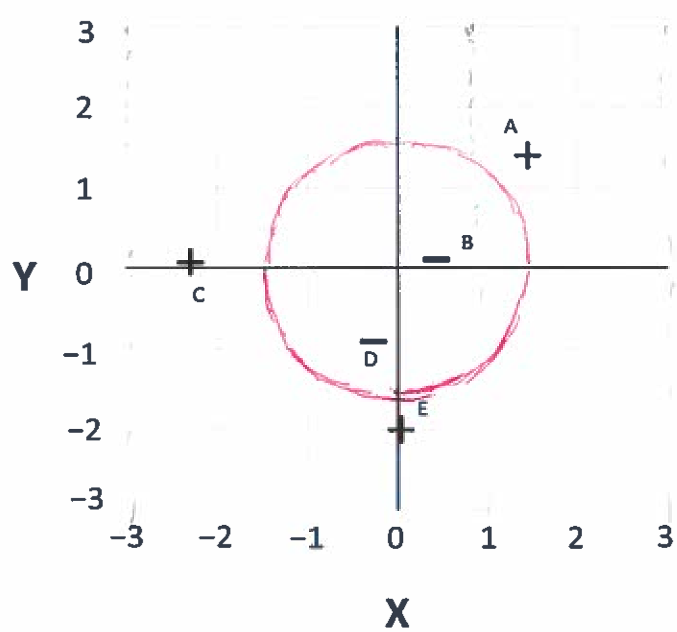
On the diagram **above**, **sketch** the SVM boundary as a **solid line.**

Sketch the boundary as it would appear on the original, untransformed space on the diagram **below**.

R = 1.5

⇒

Circle of radius 2.5 centered at (0,0)



Quiz 4, Problem 1: Adaboost (50 points)

Part A: Venture Capitalism (38 points)

Congratulations—you've just won $10,000 in this month's lottery! You've heard that investing in startups is a good idea but aren't sure how to predict which startups will be successful. You decide to use Adaboost to classify upcoming startups as successful or not, so that you can make lots of money and stay on the cutting edge of new technology.

To begin, you look at six recent startups, noting some of their characteristics and whether they've become successful:

features

Startup ID	Name	Successful	Hires Business Majors	# of Team Members	Reached Kickstarter Goal	Founder from MIT
1	FaceStalk	**Yes**	Yes	5	Yes	No
2	CouchSurfer	**Yes**	No	6	No	Yes
3	Freeloader	**No**	Yes	4	No	Yes
4	MyFace	**No**	No	1	No	No
5	Googoo	**Yes**	No	3	No	Yes
6	NapApp	**No**	Yes	5	No	Yes

A1 (6 points) You've come up with several feature tests to help you predict whether a startup will succeed. For each of the tests in the table below, circle **all** the training points that the test **misclassifies**.

Test (if True, startup is Successful)	**Misclassified Training Points** (Circle all that apply)
Hires Business Majors = Yes	1 2 3 4 5 6
# of Team Members > 3.5	1 2 3 4 5 6
# of Team Members > 4.5	1 2 3 4 5 6

A2 (4 points) You've decided to use a different set of feature tests as weak classifiers to perform Adaboost. The classifiers and the errors they make are listed here.

Test ID	Test (if True, startup is Successful)	Misclassified Training Points
A	Reached Kickstarter Goal = Yes	2 5
B	# of Team Members < 3.5	4 5
C	Founder from MIT = Yes	1 3 6
D	Founder from MIT = No	2 4 5

note that this isn't true, but we are using this for the rest of the test

closest to 0, not furthest from 1/2

If you run Adaboost choosing **the weak classifier with the lowest error rate** in each round and breaking ties randomly, which tests from the table above would you **never** choose? (Circle **all answers** that apply)

A **B** **C** **D**

NONE OF THESE

(space to show work for Adaboost, part A3)

Errors from D: 2, 4, 5 } errors from D are a superset
Errors from A: 2, 5 } of errors from A

Could have looked @ B instead of A and reached the same conclusion

A3 (24 points) Perform two rounds of boosting using only the four weak classifiers (A, B, C, D) from part A2. In each round, *choose the weak classifier with the* ***lowest error rate***. *In case of a* tie, choose the weak classifier that comes first **alphabetically**.

	Round 1	Round 2
$weight_1$	1/6	1/8
$weight_2$	1/6	1/4
$weight_3$	initialze as uniform 1/6	1/8
$weight_4$	1/6	1/8
$weight_5$	1/6	1/4
$weight_6$	1/6	1/8
Error rate of A	$\frac{1}{6}+\frac{1}{6} = \frac{2}{6}$ 2,5	$\frac{1}{4}+\frac{1}{4} = \frac{1}{2}$ ✓ 2,5
Error rate of B	$\frac{1}{6}+\frac{1}{6} = \frac{2}{6}$ 4,5	$\frac{1}{8}+\frac{1}{4} = \frac{3}{8}$ 4,5
Error rate of C	$\frac{1}{6}+\frac{1}{6}+\frac{1}{6} = \frac{3}{6}$ 1,3,6	$\frac{1}{8}+\frac{1}{8}+\frac{1}{8} = \frac{3}{8}$ 1,3,6
Error rate of D	$\frac{1}{6}+\frac{1}{6}+\frac{1}{6} = \frac{3}{6}$ 2,4,5	$\frac{1}{4}+\frac{1}{8}+\frac{1}{4} = \frac{5}{8}$ 2,4,5
weak classifier (h)	A (break AB tie)	B (break BC tie)
weak classifier error rate (ε)	2/6	3/8
voting power (α)	$\frac{1}{2}\ln(2)$	$\frac{1}{2}\ln\left(\frac{5}{3}\right)$

sum to 1/2

sum to 1/2

A4 (4 points) Here are the characteristics of a new startup called Glassr:

Name	Reached Kickstarter Goal	# Team Members	Founder from MIT
Glassr	No	3	Yes

According to the classifier you obtained from two rounds of boosting, should you invest in this startup? *(That is, is it classified as Successful by your ensemble classifier?)* (Circle one)

YES **NO**

$$H = \frac{1}{2}\ln(2)\left[\text{reached kickstarter}\right] + \frac{1}{2}\ln\left(\frac{5}{3}\right)\left[\#\text{ team members} < 3.5\right]$$

$$\downarrow \; -1 \qquad \downarrow \; +1$$

$H < 0$

Part B: Perfect Classifier (12 points)

Suppose you have six training points (P_1, P_2, P_3, P_4, P_5, P_6) and four weak classifiers (h_1, h_2, h_3, h_4), which make the following errors:

Classifier	Misclassified training points					
h_1	P_1		P_3	P_4		P_6
h_2		P_2			P_5	
h_3			P_3			
h_4				P_4		P_6

disjoint errors

B1 (5 points) Ben claims that by combining **three** of the weak classifiers above, he can construct an ensemble classifier H(x) that will correctly classify all the training data. If Ben is correct, list the three weak classifiers and assign them **integer voting powers** (α) to make a perfect ensemble classifier. If Ben is wrong, circle "CAN'T BE DONE" instead.

Weak classifier	Voting power
h_2	$\alpha =$ 1
h_3	$\alpha =$ 1
h_4	$\alpha =$ 1

CAN'T BE DONE

B2 (4 points) Alyssa claims that by combining **two** of the weak classifiers above, she can construct an ensemble classifier H(x) that will correctly classify all the training data. If Alyssa is correct, list the two weak classifiers and assign them **integer voting powers** (α) to make a perfect ensemble classifier. If Alyssa is wrong, circle "CAN'T BE DONE" instead.

If two tests disagree, cannot break tie

Weak classifier	Voting power
	$\alpha =$
	$\alpha =$

CAN'T BE DONE

B3 (3 points) What is the minimum number of rounds of boosting to produce a perfect ensemble classifier? If boosting will loop forever or terminate without producing a perfect classifier, circle CAN'T BE DONE instead.

Number of rounds: 3

CAN'T BE DONE

need at least 3 classifiers to break ties

Quiz 4, Problem 2: Bayesian inference (50 points)

Part A: *Senioritis* relapse (30 points)

Senioritis is a rare and treatable condition in general—however, here at MIT, it is positively epidemic, affecting 50% of the population. Experts have developed a cheap test for senioritis—the HACK scan—which is 80% sensitive and 60% specific. (This means that 80 out of 100 people with senioritis correctly test positive, and 60 out of every 100 people without senioritis correctly test negative. The HACK scan always reports either "positive" or "negative".)

Assume you are a typical member of the MIT population. For notation, we can let D be the variable "You have senioritis" and let T be the variable "You test positive for senioritis". Then the information above is:

$P(D)$	1 out of 2
$P(T \mid D)$	80 out of 100
$P(\overline{T} \mid \overline{D})$	60 out of 100

"nots" show up faintly online

Part A1 (10 points) What is the probability of obtaining a **negative test result**, regardless of whether you have senioritis?

The marginal probability of a negative test result is *approximately* (circle one):

0% 5% 10% 15% 20% 25% 30% 35% 40% 45% 50%

For credit, you must show your work. Write down the equations you intend to solve, if any, and indicate what values you're plugging in. You probably won't need a calculator, because you only need an approximate final answer.

Want the marginal $P(\overline{T})$

$$P(\overline{T}) = P(\overline{T}|D)\cdot P(D) + P(\overline{T}|\overline{D})P(\overline{D})$$

$$= \left(1 - \frac{80}{100}\right)\cdot \frac{1}{2} + \frac{60}{100}\cdot\frac{1}{2}$$

$$= \frac{10}{100} + \frac{30}{100}$$

$$= 40\%$$

Part A2 (10 points) Suppose your HACK scan returns a **negative result**. In this case, the probability that you indeed **don't have senioritis** is *most nearly* (circle one):

50% 55% 60% 65% 70% 75% 80% 85% 90% 95% 100%

<u>For credit, you must show your work.</u> Write down the equations you intend to solve, if any, and indicate what values you're plugging in. You probably won't need a calculator, because you only need an approximate final answer.

Want $P(\overline{D} \mid \overline{T}) = \dfrac{P(\overline{T} \mid \overline{D}) P(\overline{D})}{P(\overline{T})}$

$= \dfrac{\frac{60}{100} \cdot \frac{1}{2}}{\frac{40}{100}}$ ← from A1

$= 3/4$

$= 75\%$

Part A3 (10 points) Out of a random sample of 100 MIT students, about how many of them are expected to be false negatives—that is, how many of them will both <u>have senioritis</u> and also <u>test negative</u>?

0 10 20 30 40 50 60 70 80 90 100

<u>For credit, you must show your work.</u> Write down the equations you intend to solve, if any, and indicate what values you're plugging in. You probably won't need a calculator, since you only need an approximate final answer.

Want $P(D, \overline{T}) = P(\overline{T} \mid D) P(D)$

$= \left(1 - \frac{80}{100}\right) \cdot 1/2$

$= \frac{10}{100}$

$= 10\%$

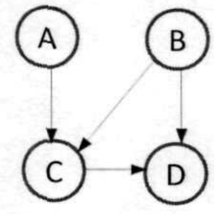
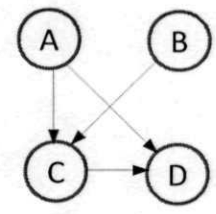
Equivalently: $P(D, \overline{T}) = P(D \mid \overline{T}) P(\overline{T})$

$= (1 - .75) \cdot .4$

$= .1$

Part B: This again, but different (16 points)
In the figure below, there are two Bayes nets and some independence statements. For each of the statements below and each Bayes net, circle TRUE if the statement is true for the net, and FALSE if the statement is false for the net.

__Note:__ Assume that the only independence statements that are true are the ones enforced by the shape of the network.

		
A is independent of C.	TRUE FALSE	TRUE FALSE
Given C, A is independent of D.	TRUE FALSE	TRUE FALSE
P(B|DAC) = P(B|AC) aka $B \perp\!\!\!\perp D \mid AC$	TRUE FALSE	TRUE FALSE
Assuming all of the variables are boolean, how many parameters does each Bayes net have? *(The number of parameters is the total number of entries in all probability tables.)*		
# of parameters	$1+1+4+4=\boxed{10}$ A B C D	$1+1+4+4=\boxed{10}$ A B C D

Part C: What are the parameters in a binary net? (4 points)
Suppose you have training data, each with one feature X, and a classification Y. Both X and Y are boolean variables, meaning they can be either true or false. Consider the Naive Bayes classifier for this problem—which of the following probabilities are the parameters of the Naive Bayes model? (Circle __ALL__ answers that apply, or circle "NONE OF THESE" instead.)

__Hint:__ It may help to draw the Bayes net that corresponds to the Naive Bayes classifier for this problem.

$P(X)$ $P(X|Y)$ $P(X|\bar{Y})$ $P(X|Y)$ $P(X|Y)$ $P(X,Y)$ $P(X,Y)$

$P(Y)$ $P(Y|X)$ $P(Y|X)$ $P(Y|X)$ $P(Y|X)$ $P(X,\bar{Y})$ $P(\bar{X},Y)$

Bayes net

NONE OF THESE

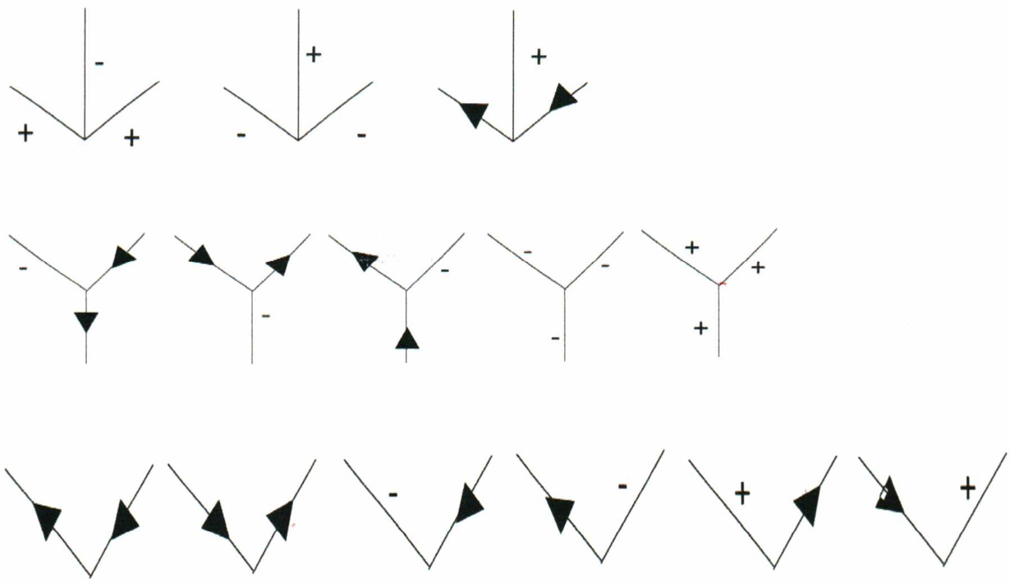
$$P(Y|X)=\frac{P(X|Y)P(Y)}{P(X)} \propto P(X|Y)P(Y)$$

$$P(\bar{Y}|X)=\frac{P(X|\bar{Y})P(\bar{Y})}{P(X)} \propto P(X|\bar{Y})(1-P(Y))$$

SRN, Quiz 2, Constraint propagation

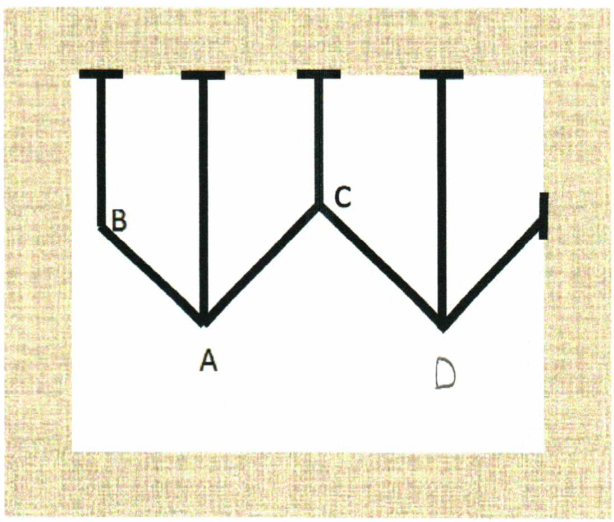
In this question, you get 2 points for the first correct answer and 1 point for each additional correct answer.

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



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

Through a window you see the following fragment. T junctions provide no constraint.



Note: No solutions for SRN multiple choice.

Part A

You are to perform pure constraint propagation. 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. (This is like "domain reduction before search" with no propagation.)

Of the 3 arrow labels in the junction library, how many are left at A after constraint propagation?

3

All work:

Of the 6 L labels in the junction library, how many are left at B after constraint propagation?

4

These work:

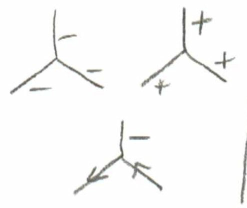
These don't work:

...because A can't be

Of the 5 fork labels, how many are left at C in the drawing after constraint propagation? (for any "?")

3

These work:



These don't work:

because A can't be

because D can't be

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

Four consistent solutions: