Name:

Massachusetts Institute of Technology

6.034 Artificial Intelligence

Final Examination **Date:**

Please indicate your section, or if you do not remember, put a tick mark on the last line.

Time	Instructor	
10-11	Davis	
11-12	Davis	
10-11	Szolovits	
11-12	Szolovits	
1-2	Shrobe	
1-2	Berwick	
2-3	Berwick	
2-3	Lozano-Perez	
3-4	Lozano-Perez	

Please indicate the TA you are assigned to.

Calista Tait	
Dennis Ruhl	
Irene Wilson	
Jonathan Loflin	
Ozlem Uzuner	
Serkan Arslanalp	

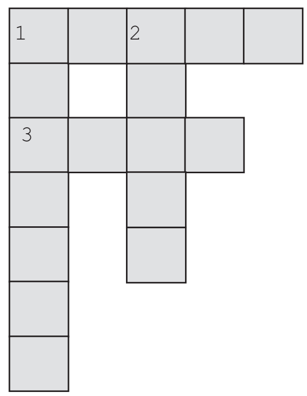
Problem 1	
Problem 2	
Problem 3	
Problem 4	
Problem 5	
Total	

Special Notes

- On the first two examinations, many students lost many points because of misinterpretations caused by failing to read carefully. **Many even missed warnings printed in bold face.**
- Avoid getting stuck on particular problems. You will probably get a higher grade if you do some work on all problems, rather than better work on fewer problems.

Problem 1 Constraint Satisfaction (20 points)

Consider the word problem below, which is something like a crossword puzzle (though perhaps easier). You are given a grid into which the words must fit:

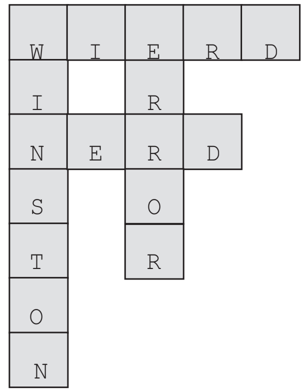


And, you are given a list of words to use. Not all words have to be used; words can be repeated.

Words to use

```
AI
MIT
VEST
EXAM
NERD
WIERD
ERROR
COURSE
LECTURE
WINSTON
ARTIFICIAL
RECITATION
INSTITUTION
INTRODUCTION
INTELLIGENCE
MASSACHUSETTS
```

One solution to this problem is shown here:



Part A

We'd like you to view this problem as a constraint satisfaction problem.

A.1 (2 points)

What are the variables for this problem? Choose the best possibility from the list below:

a) the words given in the word list

b) each individual box in the grid

c) the set of all the words inserted into the grid

d) each vertically or horizontally contiguous sequence of boxes in the grid (eg the sequence 1-down or 3-across).

e) the letters of the alphabet

A.2 (2 points)

What are the values for the variables? Again choose the best possibility from the list below:

a) the words given in the word list

b) each individual box in the grid

c) the set of all the words inserted into the grid

d) each vertically or horizontally contiguous sequence of boxes in the grid (eg the sequence 1-down or 3-across).

e) the letters of the alphabet

A.3 (4 points)

What are the constraints for this problem. For each possible type of constraint, (a) write a sentence that describes the constraint and (b) give an example from the problem in Figure 1 above. For example, in the map coloring problem, a constaint is that no two countries that share a common boundary can have the same color, and an example of that constraint is that if the United States is colored red, then Canada cannot be colored red.

Use a different line for each type of constraint in the **crossword puzzle problem**; we have given you more than enough lines.

1a: Constraint

1b: Example

2a: Constraint

2b: Example

3a: Constraint

3b: Example

4a: Constraint

4b: Example

Part B

Important: Part B of this problem is concerned with map coloring, a different CSP. All of the questions below refer to map coloring.

In each sub-problem you are given the same map and four colors with which to color it: blue, green, red, and yellow. We want you to show how to use several different constraint satisfaction algorithms to the map coloring problem.

You should show your work by annotating the tables given with each map. When you select a value, circle it; when you eliminate a value, draw a line through it. Each time you select or eliminate a value, *indicate when* that event happened by putting a number next to that circle or slash. (If you select and then later eliminate a value you will first circle it and then slash it.) As a hypothetical (and random) example, a table could be annotated to show:

Country

I	II	III	IV
Blue	~~Blue~~ 2	$\boxed{\text{Blue}}$ 1	Blue
Green	Green	Green	Green
Red	Red	Red	Red
Yellow	Yellow	Yellow	Yellow

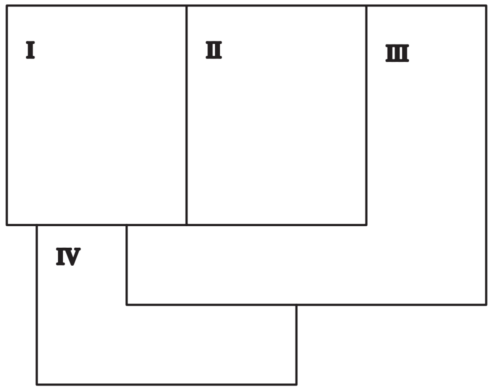
This indicates that we began by selecting Blue for country III, next eliminated Blue for country II, etc. Note that selecting Blue for country III implicitly eliminates the other color choices for that country; to keep the table from becoming too marked up, you need *not* indicate those eliminations.

Important: Whenever you have a choice among countries, select the country with the lowest number first; whenever you have a choice among colors you should select them in the (alphabetical) order shown above.

We have given an extra copy of the table, to make it easy if you want to change an answer. If you wish, rather than erasing you may *cross out* the first table and annotate the second copy.

B.1 (3 points)

Solve this map coloring problem using pure backtracking; don't forget to include not only circles and slashes, but also numbers, as described above.



Country				Country			
I	II	III	IV	I	II	III	IV
Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
Green	Green	Green	Green	Green	Green	Green	Green
Red	Red	Red	Red	Red	Red	Red	Red
Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow

B.2 (3 points)

Solve this problem using pure propagation of constraints, assuming country I is Blue.

Country				Country			
I	II	III	IV	I	II	III	IV
Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
Green	Green	Green	Green	Green	Green	Green	Green
Red	Red	Red	Red	Red	Red	Red	Red
Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow

B.3 (3 points)

Solve this problem using backtrack with forward checking.

Country				Country			
I	II	III	IV	I	II	III	IV
Blue	Blue	Blue	Blue	Blue	Blue	Blue	Blue
Green	Green	Green	Green	Green	Green	Green	Green
Red	Red	Red	Red	Red	Red	Red	Red
Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow

B.4 (3 points)

When doing constraint satisfaction we can stop with the first successful solution or can continue on, finding all possible solutions. When doing constraint satisfaction via the pure backtracking approach, the search is done depth first.

This gives Ben Bitdiddle an idea: he suggests using breadth-first search through the space of possible assignments. He claims that this too will find all possible solutions, will avoid having to back up, and will be time and space efficient. He's so happy he decides to publish this as the Bitiddle Algorithm. Luckily, just before sending it off to a journal, he asks you for assistance:

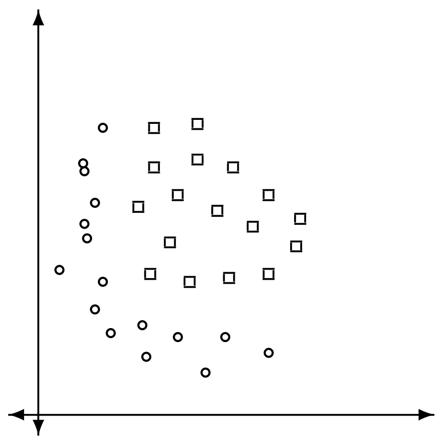
a) Is it true that breadth first will also find all of the solutions? Answer yes or no:

b) If *no*, explain why. If *yes* explain any serious drawbacks to using this approach.

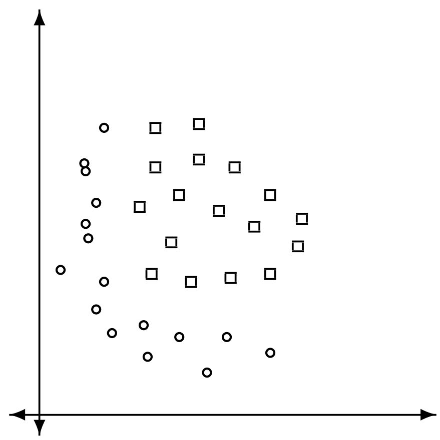
Problem 2 (15 points)

Part A Decision Boundaries (8 points)

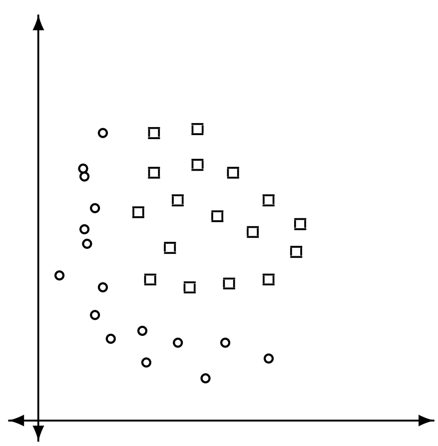
On the graphs below draw in the *approximate* decision boundaries that would be determined by three different learning algorithms: Nearest Neighbor, Identification Tree, and a Single Unit Perceptron. (Recall that a perceptron is a neural net in which the threshold functions are step functions, not sigmoids.) In each case assume that the squares are positive and the circles are negative. You need not find the precisely correct decision boundary, but your answer should be close and it should show that you know the type of decision boundaries that each algorithm can construct. If you feel you cannot draw the decision boundary sufficiently clearly, you may add a sentence of description to disambiguate your drawing (though it is not necessary).



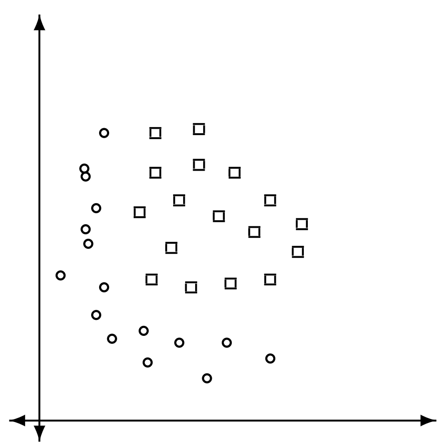
On this graph draw an approximate decision boundary for a nearest neighbor classifier trained on this data.



On this graph draw an approximate decision boundary for an Identification Tree trained on this data.



On this graph draw an approximate decision boundary for a single perceptron unit which minimizes the misclassification error.



This is a practice graph, just in case.

If you were asked to draw the decision boundary for a multi-layer sigmoidal neural network which minimizes the misclassification error, the correct response would be "I can't do that unless you tell me more information". What sort of information do you need?

Part B Distance to your Nearest Neighbor (7 points)

You are given a set of one thousand points which are distributed in a 500 dimensional space (i.e. each can be described as a 500 dimensional vector $\vec{x_i}$). The points are uniformly distributed in a 500 dimensional sphere centered at the origin with radius 1. We will label points as *positive* if the first dimension is greater than 0: $\vec{x_i}[1] > 0$. We will label it *negative* if the first dimension is less than zero: $\vec{x_i}[1] < 0$. (Note that since the sphere of points is centered at the origin roughly half of the points will be positive and half negative).

Would simple nearest neighbor, when trained with the 1000 classified points, be a useful technique for classifying a new random point correctly? Explain. If you believe that simple nearest neighbors would not work well, what modification would allow for correct classification of new random points most of the time? Explain.

Could a perceptron, when trained with the 1000 classified points, successfully classify a new random point correctly? Explain. If you believe that a perceptron would not work well, what modification would allow for correct classification of new random points most of the time? Explain.

Would an identification tree, when trained with the 1000 classified points, be a useful technique for classifying a new random point correctly? Explain. If you believe that an identification tree would not work well, what modification would allow for correct classification of new random points most of the time? Explain.

Problem 3 :Crazy Frank (25 Points)

Several years ago, when genetic algorithms were first becoming popular, some of the 6.034 recitation sections were run by Professor Frank N. Stein. The other professors referred to him as “Crazy Frank”. It took a while for the authorities to discover why. This is his story.

Prof. Stein was worried about his students’ grades, and decided to improve them using a genetic algorithm. Stein had contacts at Clones’r’Us, a local biotechnology firm with a new technique for emulating sexual reproduction by fusing cells from two humans (regardless of their gender), mutating the ‘child’ or ‘spawn’, and growing the result into a sophomore-sized person within a couple of days.

So, each Friday, all the students in Prof. Stein’s section took a quiz to see what they had learned that week. Students who did well were selected for reproduction: tissue samples from these students would be taken (painlessly) and sent to Clones’r’Us. Every Monday morning a crate of newly-spawned students would arrive. To keep the number of people in his recitations constant, Stein arranged for students who did badly on the quiz to be flown across the Atlantic to work in British universities (this, he later explained, was roughly the same as killing them, only it was legal). Stein’s idea was that, in the week before the final quiz, the students would do a review of the whole 6.034 course and in that week they would learn everything they needed to know – this idea had outlasted Professor Stein by many years.

Shortly before he was arrested, Prof. Stein gave a seminar where he explained his methods. He had four recitation sections, $R1$, $R2$, $R3$, and $R4$. A preliminary quiz at the start of the course showed that, in each section, the spread of student abilities was the same.

Each week, with section $R1$, Stein took tissue samples from the top two students only: he sent the samples to Clones’r’Us with instructions that, while it was OK to do crossover to combine good genes from the two parents, he didn’t want these good genes messed up, so there should be no mutation. And he ordered enough ‘spawn’ to replace all the students in the section.

With section $R2$, Stein did the same as with $R1$, except he wanted to try more variation, so he instructed Clones’r’Us to use mutation rates of approximately one mutation per gene: this is a rate so high that it was possible that every gene in a mutant spawn could differ from the genes of the original.

With sections $R3$ and $R4$, Stein used a medium mutation rate, and every week he ordered enough spawn to replace the entire sections. Each spawn was generated from two tissue samples. The two samples were taken from students in the current week’s section in such a way that everyone in the section had a chance of providing a sample, but people who’d done better in the quiz had more chance of being selected to provide tissue. Any one student could give multiple samples, and there was nothing to stop a spawn being created from two samples of the same student.

In $R3$, every time a tissue sample was needed, a ‘parent’ student was selected from the current class using the ‘standard’ (roulette-selection) stochastic method where the probability of a student being selected was given by that student’s proportion of the total fitness for the $R3$ population.

In $R4$, every time a tissue sample was needed, a ‘parent’ student was selected from the current class using a rank-based stochastic method. The students’ scores were ranked worst-to-best. That is, in a group of n students, the worst-scoring student(s) have a rank of 1 and the best-scoring student(s) have a rank of n. The probability of a student being selected was calculated by dividing the student’s rank number by the sum of all the rank numbers in the $R4$ section. *(Note: this differs from the rank-selection method in Winston’s “Artificial Intelligence” book).*

Part A : What might go wrong? (4 points)

In each recitation section, an evolutionary search is in progress:

- Which, of $R1$ to $R4$, is most likely to get stuck early on a local maximum? (1 point)

- Which, of $R1$ to $R4$, is unlikely to do much better than random search? (1 point)

Part B : What's the difference? (6 points)

Coincidently, there were six students in section $R3$ and six in $R4$, and on some weeks the scores of the students in $R3$ were, surprisingly, the same as the scores of the students in $R4$. In weeks 2, 3, and 11, the $R3$ and $R4$ recitations had the same distribution of scores. In week 2, the scores (ordered worst-to-best) were: 4%, 4%, 5%, 6%, 7%, and 8%. In week 3, the scores are 5%, 5%, 6%, 9%, 9%, and 67% – the one high scorer was a lucky mutant spawn. In week 11, the scores are: 64%, 64%, 67%, 73%, 75%, and 78%. Complete the following tables by writing each student's probability of being selected for reproduction, either as fractions or as real values rounded to 2 decimal places.

- Calculate the probability of being a parent for the lowest-scoring and highest-scoring individuals in $R3$ and $R4$ in weeks 2, 3, and 11. (4 points)

	Wk 2: lo	Wk 2: hi	Wk 3: lo	Wk 3:hi	Wk 11: lo	Wk 11: hi
Score	4%	8%	5%	67%	64%	78%
$R3$ Prob.						
$R4$ Prob.						

- Assume that diversity is the same in both $R3$ and $R4$ before the end of week 3. From week 4 onwards, is the diversity in $R3$ likely to be less or more than in $R4$ (2 points)?

Part C : Clones'r'Us (4 points)

Stein also gave an example of the process used by Clones'r'Us. Two DNA strands were extracted – one from each tissue sample. They then went through a highly secret chemical process that turned them into short strings of characters from the binary alphabet {`0,1`}. Let n be the length of the strings. The Clones'r'Us mutation operator took a string, randomly selected one location in the string, 'flipped' the bit at that location (i.e., turned a `1` into a `0` or turned a `0` into a `1`), and returned the result. Clones'r'Us also used a 'one-point' crossover operator that took two 'parent' strings and returned a 'spawn' string: it worked by randomly selecting a position $p : p \in \{1, 2, \ldots, n\}$, and forming the spawn by concatenating characters 1 to p of one parent's string with characters p+1 to n of the other parent's string. The choice of which parent is used for the first part and which is used for the second is made randomly.

Let $\mathcal{M}(S)$ represent the result of applying one-bit mutation to the string S, and let $\mathcal{C}(S1, S2)$ represent the result of crossover between strings $S1$ and $S2$ – the order of $S1$ and $S2$ doesn't matter: the crossover process randomly chooses between copying a substring from the front of $S1$ followed by a substring from the rear of $S2$, or vice versa.

In Stein's example, $n = 8$ and A was `11101110` and B was `01110111`. For each of the following strings marked '*?*', and the above definitions of crossover and mutation, circle ONE of (a), (b), (c), (d), or (e) that best describes the '*?*' string:

1	1	1	0	1	1	1	0	A
0	1	1	1	0	1	1	1	B
1	0	0	0	1	1	1	0	?

(a) $\mathcal{M}(A)$; (b) $\mathcal{M}(B)$; (c) $\mathcal{C}(A, B)$; (d) $\mathcal{M}(\mathcal{C}(A, B))$; (e) None of these.

1	1	1	0	1	1	1	0	A
0	1	1	1	0	1	1	1	B
0	1	1	1	1	1	1	0	?

(a) $\mathcal{M}(A)$; (b) $\mathcal{M}(B)$; (c) $\mathcal{C}(A, B)$; (d) $\mathcal{M}(\mathcal{C}(A, B))$; (e) None of these.

1	1	1	0	1	1	1	0	A
0	1	1	1	0	1	1	1	B
1	0	1	0	1	1	1	0	?

(a) $\mathcal{M}(A)$; (b) $\mathcal{M}(B)$; (c) $\mathcal{C}(A, B)$; (d) $\mathcal{M}(\mathcal{C}(A, B))$; (e) None of these.

1	1	1	0	1	1	1	0	A
0	1	1	1	0	1	1	1	B
1	1	1	0	0	1	1	0	?

(a) $\mathcal{M}(A)$; (b) $\mathcal{M}(B)$; (c) $\mathcal{C}(A, B)$; (d) $\mathcal{M}(\mathcal{C}(A, B))$; (e) None of these.

Part D : Stein's last quiz (4 points)

At the end Stein's fateful last seminar, he announced that he was going to use his genetic algorithm to improve the intelligence of the MIT community, starting with the seminar audience. He said that he would give the audience a two-question quiz: people who did well would be spawned, while people who did badly would be put on the next boat for England. At this point, he was led away by Campus Police.

However, you have to answer his quiz, reproduced below:

HERE ARE THREE 'POPULATIONS', $P1$, $P2$, AND $P3$, EACH WITH 6 INDIVIDUALS.

$P1$	$P2$	$P3$
11110111	00000100	11010100
10111111	01000000	10100011
11111111	00000010	11111111
11111101	00000000	00101011
10111111	00010000	00000000
11110111	00100000	01011100

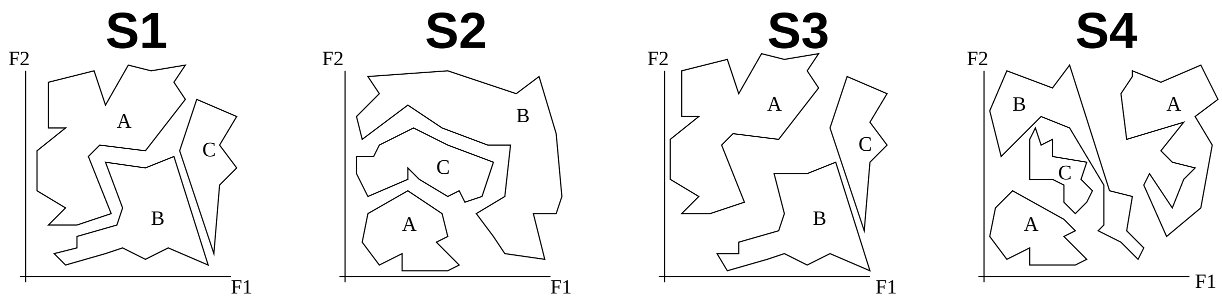
- WHICH POPULATION IS THE MOST DIVERSE?

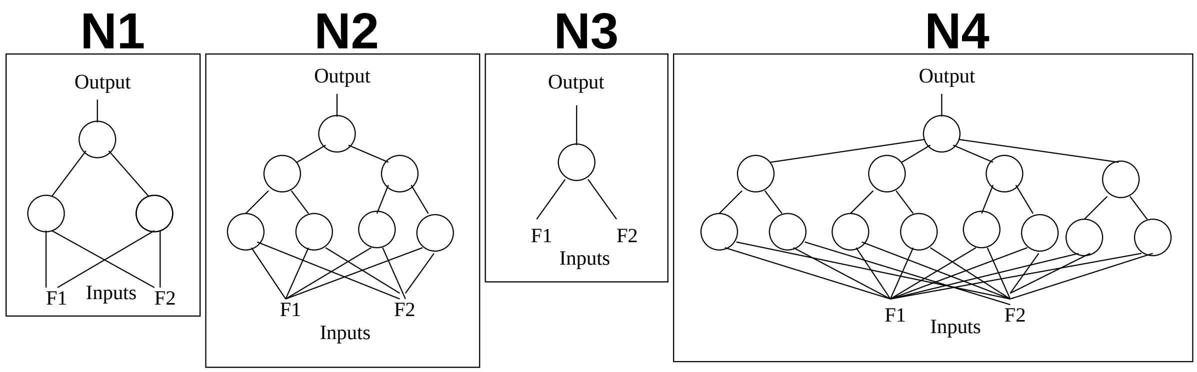
- WHICH POPULATION IS THE LEAST DIVERSE?

Part E : Mr Percy Eptron (4 points)

When the investigators were going through Prof. Stein's papers, they found that the TA who did the grading on the weekly quiz, Mr Percy Eptron, was in fact a neural network program written several years previously by one of Stein's UROP students. Stein had been experimenting with altering the structure of 'Percy' to improve the performance. The Percy network was built from perceptron units that applied a threshold function to the weighted sum of their inputs, and returned either 0.0 or 1.0. The network was supposed to return 1.0 for an A-grade student, and 0.0 for anyone who got grades B or C.

It was clear that Stein had been studying the effect the number of units, and their arrangement into layers, had on the ability to learn. Recall that a perceptron unit gives a linear decision boundary in the input feature-space. In the following figure, the top row shows four figures of 2-dimensional feature-spaces, S1 to S4. Each feature-space has regions marked A, B, and C. The bottom row shows four perceptron networks, N1 to N4.





For each feature-space-region figure, circle the label of the *simplest* perceptron network that could distinguish A-grade students from those with lower (not-A) grades without any misclassifications. *(Note: if you want to, you can use the same network more than once).*

- **S1:** (a)N1 (b)N2 (c)N3 (d)N4
- **S2:** (a)N1 (b)N2 (c)N3 (d)N4
- **S3:** (a)N1 (b)N2 (c)N3 (d)N4
- **S4:** (a)N1 (b)N2 (c)N3 (d)N4

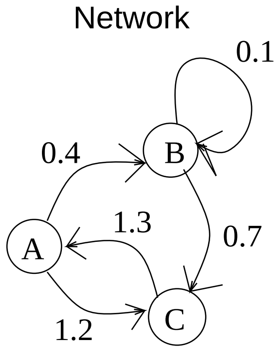
Part F : Encodings (3 points)

On the chalk-board in Stein's office there were sketches of his plans to improve the Percy network by applying a genetic algorithm. Stein had been experimenting with a system where the initial values for the connection weights were encoded as 'genotypes'.

Each genotype in the population was a string of real numbers. The connection weights were encoded by treating each row of the network's connection matrix as a string of real numbers, and concatenating these strings for all the rows in the matrix to form one big string. The following figure shows a simple example:

Stein was using this method for encoding the connection matrices of feed-forward backpropagation networks. *(Feed-forward networks are introduced on page 444 of Winston's "Artificial Intelligence" book).*

For a network with n units, this encoding scheme requires n^2 real numbers in each genotype. Suggest a different encoding method that takes advantage of the fact that Stein was only interested in evolving



Connection Matrix

	A	B	C
A	0.0	0.4	1.2
B	0.0	0.1	0.7
C	1.3	0.0	0.0

Genotype

0.0 0.4 1.2 0.0 0.1 0.7 1.3 0.0 0.0

connections for feed-forward backpropagation networks. In comparison to Stein's method, the new encoding method should never require more than half as many connections-values to be specified on each genotype, even if the feed-forward network is fully connected.

Problem 4 Grading Trees (20 pts)

All of the faculty and students of the AI Laboratory have left MIT to start a new company that makes PC games. Unfortunately, they have left before the end of the term. Now, it is up to you to assign grades for 6.034. You want to do this consistently with the policy from previous years, so you decide to use grade data from previous years to train a learning system to assign grades for this year. The training data consists of records like the following:

Name	Course	Year	Q1	Q2	Final	PS	Grade
J. Foo	6	2	100	100	100	85	B

The PS grade is the summary problem set grade, the sum of ten problem set scores each scored out of 10 points. The field labeled Grade is the desired "class", not one of the attributes.

For simplicity, assume we are focusing on trying to classify the input samples into two classes: those that will receive an A grade and those that won't.

Part A (2 points)

If you ran the standard identification-tree builder (such as you saw in Chapter 21) on the full data set given above, which of the attributes (drawn from Name, Course, Year, Q1, Q2, Final, PS) would always end up at the top node. Explain.

Hint: You do not need to see the data to answer this question.

Part B (2 points)

In the rest of the problem, we will be using the following formula for computing disorder of a set split into two classes, where x is the fraction of the elements in one of the two classes:

$$D(x) = 1 - 2|x - 0.5|$$

This takes the place of the usual disorder measure (entropy):

$$E(x) = -x\log(x) - (1-x)\log(1-x)$$

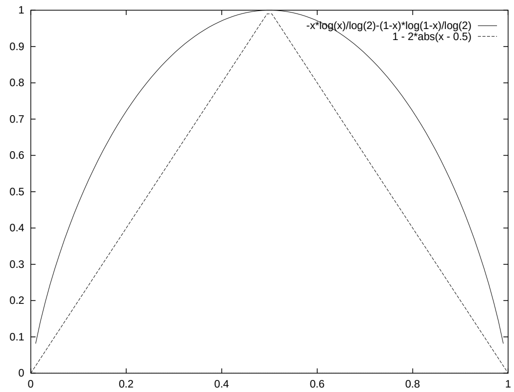


Figure 1: Two disorder measures.

Consider using this disorder formula within the following implementation of an identification-tree builder (a slightly refined version of the one in Chapter 21, page 431):

1. Until each leaf is populated by a homogeneous sample set:
   (a) Select a leaf node N with an inhomogeneous sample set.
   (b) Construct a test node T that divides the inhomogeneous sample set into minimally disordered sets, according to some measure of average disorder.
   (c) If the average disorder of the new test node T is less than the disorder of the leaf node N, then replace T by N.

In general, which of the following properties of a disorder measure is the single one required to guarantee that the tree-building procedure above always terminates? You may assume that an identification tree with homogeneous leaves does exists.

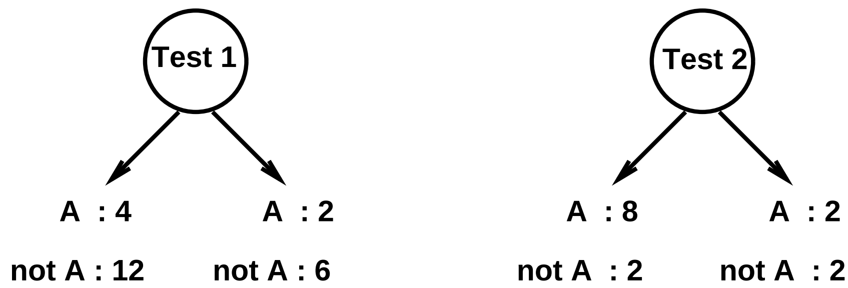
Circle only one:

1. The max value is 1.
2. The max value is at $x = 0.5$.
3. The min value is 0.
4. The min values are at $x = 0$ and $x = 1$.
5. The function is symmetric about $x = 0.5$.
6. None of the above.

Part C (4 points)

The figure below shows the results of two tests. The numbers show how the samples passing the test (on the left) and failing the test (on the right) are distributed into the two classes: those getting an A grade and the others (not A).

Which of the following nodes would be placed at the top of an identification tree, assuming we measure disorder using $D(x)$ as given above. **Show how you arrived at your answer.**



Part D (2 points)

Assume that one of the grading policies last year was that if someone scored 85 or higher in the final then they got an A regardless of their other scores. Note that only a fraction of the A's would have been awarded under this policy.

Is there a good chance that such a policy would show up directly in the identification tree built from the data (ignoring the attribute chosen in Part A)? That is, would you expect to see a node in the tree whose test checks the Final grade against 85 (or a very similar number). Explain your answer. Hint: Think about the expected distribution of data and how the tree is built.

Part E (5 points)

How about if, instead, the policy had been that if any two of the three exams (Final, Q1 or Q2) were above 85, then the grade was an A regardless of their other scores. Note that not all the A's would be awarded under this policy.

Draw a tree that implements this policy. Make sure you label the tree with tests, that you label the branches with Y or N and that you label the leaves with the intended class. Leaves whose class is not determined by the policy should be left blank. There is not a unique tree that implements this policy; draw the one that looks at Final first and then Q1 before Q2.

Would this policy be (a) more, (b) equally or (c) less likely than the policy in Part D to show up in the tree built from the data (ignoring the attribute chosen in Part A). That is, how likely (relative to Part D) do you think it is that this subtree (or one of its reorderings and possibly with small changes in the numbers) shows up in the answer. Explain your answer. Hint: Think about the expected distribution of data and how the tree is built.

Part F (3 points)

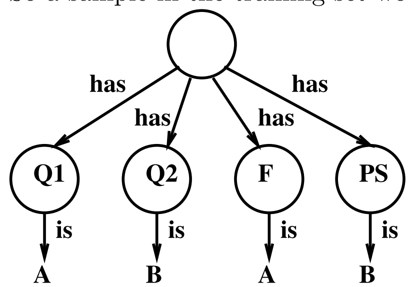
Unfortunately, if you were to apply any tree that you derive from last year's grades directly to the grades for this year, you would face a student revolt.

1. Explain what the difficulty is.

2. How could you process the data so that it would be fair to apply results from last year to this year?

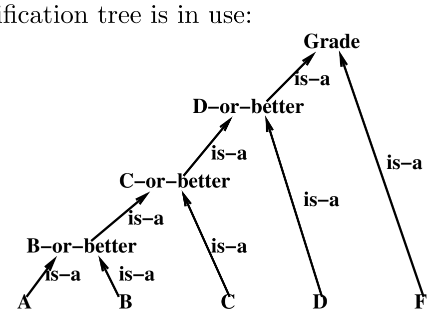
Part G (2 points)

Imagine that we want to use Procedure W you saw in lecture to do the grading task instead. Since Procedure W does not deal with numerical data, let's assume we process the individual exam and problem set scores into letter grades, e.g. A, B, C, etc. So a sample in the training set would be represented as follows:



An example for Procedure W.

Let's assume that Procedure W only uses the **require-link, forbid-link, climb-tree** and **drop-link** operations. Assume the following classification tree is in use:



The classification tree for grades.

Which of the following series of samples will teach Procedure W the concept: *An A in the final and a B-or-better on everything else is* **required** *to get an A in the course*? Circle all the ones that would work. The first column indicates whether the sample is positive (an example of the concept) or negative (a counterexample).

1.

Pos or Neg	Q1	Q2	Final	PS
Neg	A	A	B	A
Pos	A	A	A	A
Pos	B	B	A	B

2.

Pos or Neg	Q1	Q2	Final	PS
Pos	A	A	A	A
Neg	A	A	B	A
Pos	B	B	A	B

3.

Pos or Neg	Q1	Q2	Final	PS
Pos	A	A	A	A
Pos	B	B	A	B
Neg	A	A	B	A

4.

Pos or Neg	Q1	Q2	Final	PS
Pos	A	A	A	A
Neg	A	A	B	A
Pos	B	A	A	A
Pos	A	B	A	A
Pos	A	A	A	B

5.

Pos or Neg	Q1	Q2	Final	PS
Neg	A	A	B	A
Pos	A	A	A	A
Pos	B	A	A	A
Pos	A	B	A	A
Pos	A	A	A	B

6.

Pos or Neg	Q1	Q2	Final	PS
Pos	A	A	A	A
Pos	B	A	A	A
Pos	A	B	A	A
Pos	A	A	A	B
Neg	A	A	B	A

Problem 5 Rules & Probabilistic Reasoning (5 points)

You are given the following set of rules to identify animals, a slight variation of the animals rules you've seen previously.

```
R1:    If      x gives milk
       then    x is a mammal

R2     If      x is warm-blooded
       then    x is a mammal

R3     If      x is a mammal
               x eats meat
       then    x is a carnivore

R4     If      x is a mammal
               x has pointed teeth
               x has forward-looking eyes
       then    x is a carnivore

R5     If      x has whiskers
               x has claws
       then    x is a feline

R6     If      x has light fur
               x has dark vertical bands
       then    x has stripes

R7     If      x is a carnivore
               x is a feline
               x has stripes
       then    x is a tiger
```

In this part, you are given a **backward-chaining** rule interpreter. The rule interpreter's job is to identify a particular animal, Fuzzy. The interpreter starts with the hypothesis that Fuzzy is a tiger. You are also told that the interpreter selects the rules in the order they appear in the rule set, and works through them depth first. Similarly, you are told that the interpreter works on the clauses in a given rule in the order that the clauses occur in that rule.

You are told as well that the system starts with no initial assertions in the database, but that once a question is answered (or a rule fired), the answer (or inference) is recorded in the database and used by the system (rather than asking the question again).

You are to assume that the user answers questions by looking at the list provided below: any question corresponding to one of those assertions is to be answered yes; other questions are to be answered no.

```
Fuzzy eats meat
Fuzzy gives milk
Fuzzy has pointed teeth
Fuzzy is warm-blooded
Fuzzy has forward-looking eyes
Fuzzy has claws
Fuzzy has whiskers
Fuzzy has stripes
```

A backward chaining interpreter produces an AND/OR tree. Simulate the behavior of this interpreter by completing the AND/OR tree below (we have done the first step for you to show you what we mean) and by showing the order in which the depth-first search backward chaining would ask questions of the user (we have given you more than enough lines). Use the standard arc across the branches of AND nodes to distinguish them from OR nodes.

```
              tiger
                |
       ---------|--------
       |        |       |
   carnivore  feline  stripes
```

Questions asked of the user (we have given you more than enough lines):

1)

2)

3)

4)

5)

6)

7)

8)

9)

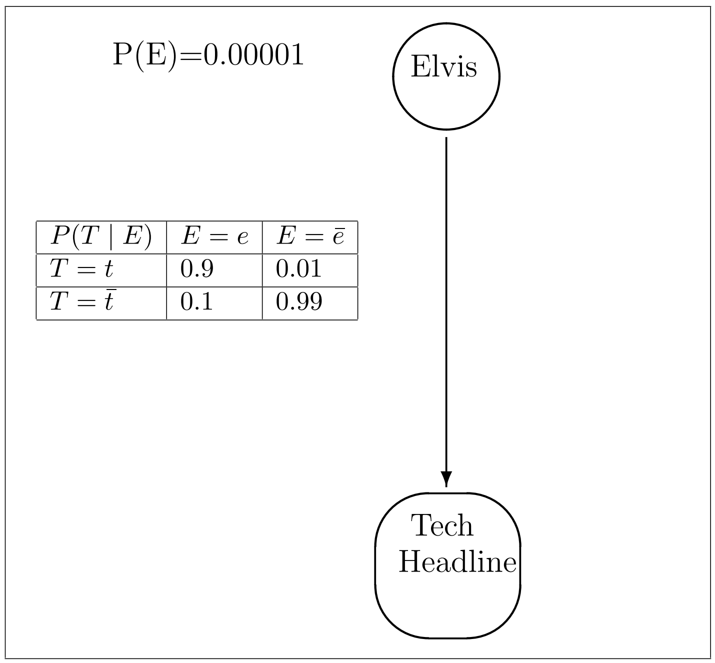
10)

11)

12)

Problem 6 Elvis Lives! (15 points)

Alyssa P. Hacker, the tireless undergraduate, has decided to use her 6.034 knowledge to construct a Bayes net to model the probability of Elvis making an appearance at MIT. She reasons that if Elvis were to appear, the Tech would almost certainly make this the headline story (with probability 0.9). The Tech is not perfect, however, so the Tech may claim that Elvis is coming even though he is not, as a joke (with probability 0.01). Alyssa believes that the probability that Elvis will appear at MIT is very low: 0.00001. Alyssa decides that the she will create a Bayes net in order to reason about this situation.



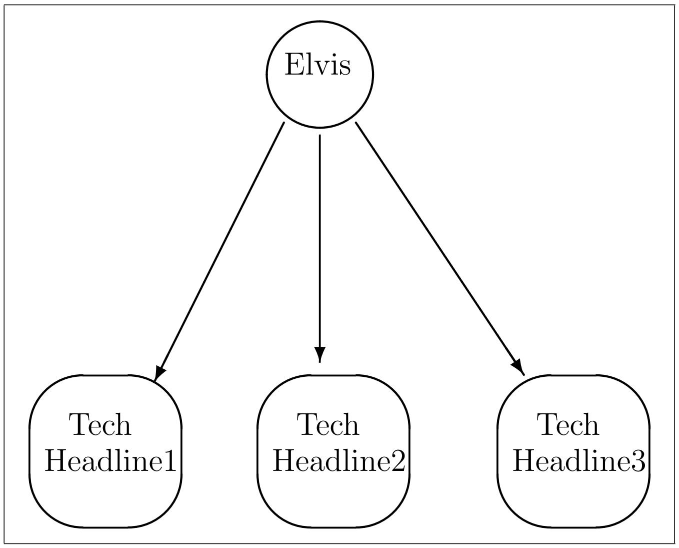
$P(T \mid E)$	$E = e$	$E = \bar{e}$
$T = t$	0.9	0.01
$T = \bar{t}$	0.1	0.99

Part A (3 points)

One day Alyssa is walking around campus when she sees a copy of the Tech lying on the ground. This copy of the Tech has the headline "Elvis appears at MIT". Using this model what is the probability that Elvis has appeared?

Part B (3 points)

Later on she sees two additional copies of the Tech with the same headline (i.e. three copies of the same issue). After some thought she decides to model this situation by making copies of the "Tech Headline" nodes. In order to keep things simple she keeps the conditional probability of Elvis given each Headline node the same as before (i.e. $P(T_3|E) = P(T_2|E) = P(T|E)$):



Under this model what is the probability of Elvis appearing given that the three papers have the headline "Elvis appears at MIT"?

Part C (3 points)

Later on Alyssa sees 500 more copies of Tech each with the same headline: "Elvis appears at MIT." Without thinking she simply extends her model from before so that there are now 500 "Tech Headline" nodes. In this model what is the probability that Elvis is going to appear?

Part D (3 points)

Alyssa realizes that something is very strange about this scheme. Explain in a sentence or two what the problem is.

Part E (3 points)

She mentions this problem to Ben Bitdiddle who, after some thought, says "I think you can add a single node to this net and then rearrange the connections. That should fix your problem." In the space below draw a simplified version of this *improved net*. You need not modify the 501 node net. You can start from the net with 4 nodes shown above. The resulting net will have 5 nodes.