6.034 Quiz 2

19 October 2018

Name	
Email	

For 1 extra credit point: Circle the TA whose recitations you attend so that we can more easily enter your score in our records and return your quiz to you promptly.

Suri Bandler

Sanchit Bhattacharjee

Alex Charidis

Samir Dutta

Marie Feng

Ariel Jacobs

Victoria Longe

Smriti Pramanick

Kifle Woldu

Matt Wu

Richard Yip

Problem number	Maximum	Score	Grader
1 - kNN	27		
2 - ID trees	33		
3 - Constraints	40		
Total	100		

4 - SRN	6		

There are 18 pages in this quiz, including this page, but not including tear-off sheets. Tear-off sheets with duplicate drawings and data are located after the final page of the quiz. **We do not collect tear-off sheets, so show your work on the quiz pages, not the tear-off sheets.**

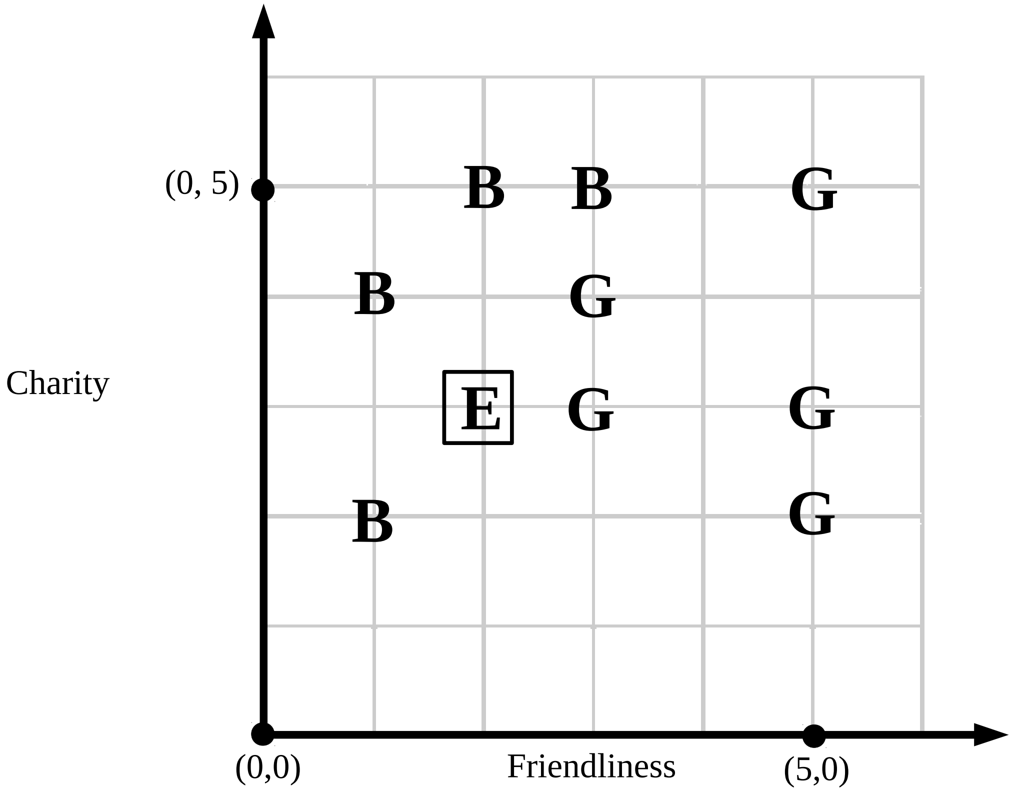
As always, open book, open notes, open just about everything, including a calculator, but no computers.

This page is intentionally blank.

Problem 1: k-Nearest Neighbors (27 points)

Part A: Are Eleanor and Tahani Good? (12 points)

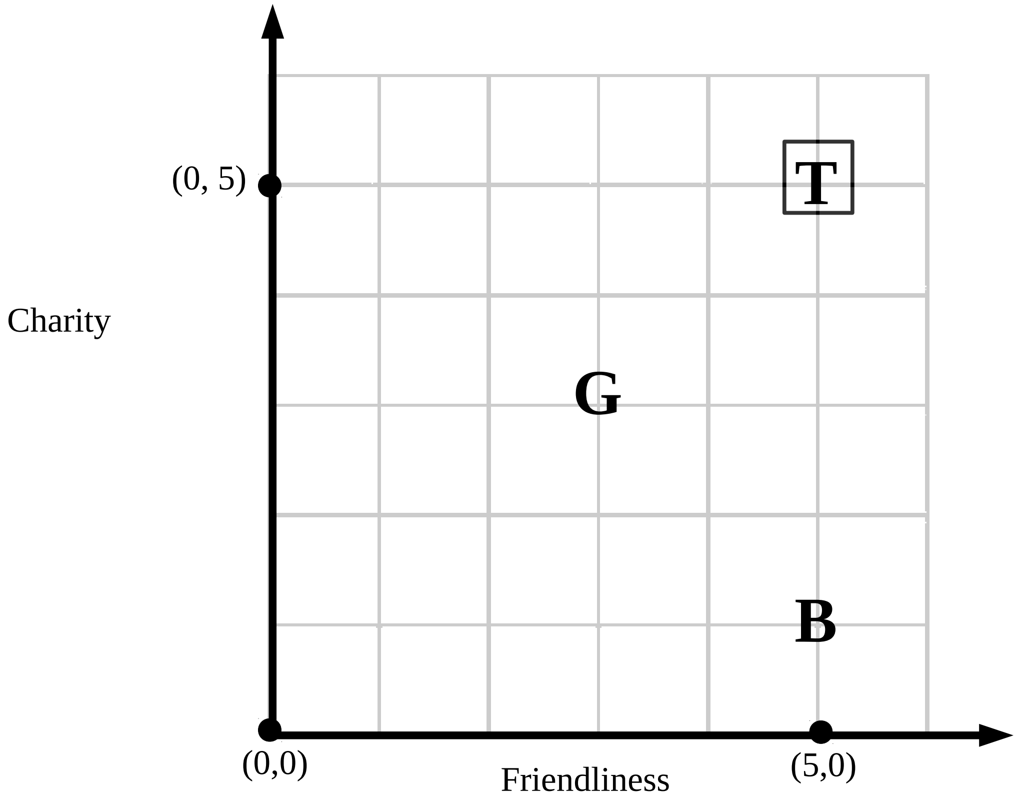
In the television show *The Good Place*, Michael discovers that Eleanor has arrived in his community accidentally, and he wants to find any reason to justify her staying there. He decides to use the k-Nearest Neighbors algorithm to classify test point Eleanor based on a data set of good (G) and bad (B) people. The data set is graphed using two dimensions, friendliness and charity. The graph is below. Eleanor's position is shown as $\boxed{\textbf{E}}$.



A1 (4 points) Using the graph above, for each different k value below, circle the one best classification for Eleanor $\boxed{\text{E}}$ using k-Nearest Neighbors and **Euclidean distance**:

k=1	Good	Bad	Can't Tell
k=5	Good	Bad	Can't Tell
k=8	Good	Bad	Can't Tell
k=9	Good	Bad	Can't Tell

A2 (8 points) Michael also wants to see if Tahani will be classified as good (G) or bad (B). Below is a graph showing test point Tahani and two close neighbors, one good (G) and one bad (B). Tahini's position is shown as $\boxed{\text{T}}$.



Using 1-Nearest Neighbor, classify test point Tahani $\boxed{\text{T}}$ using each of the following distance metrics:

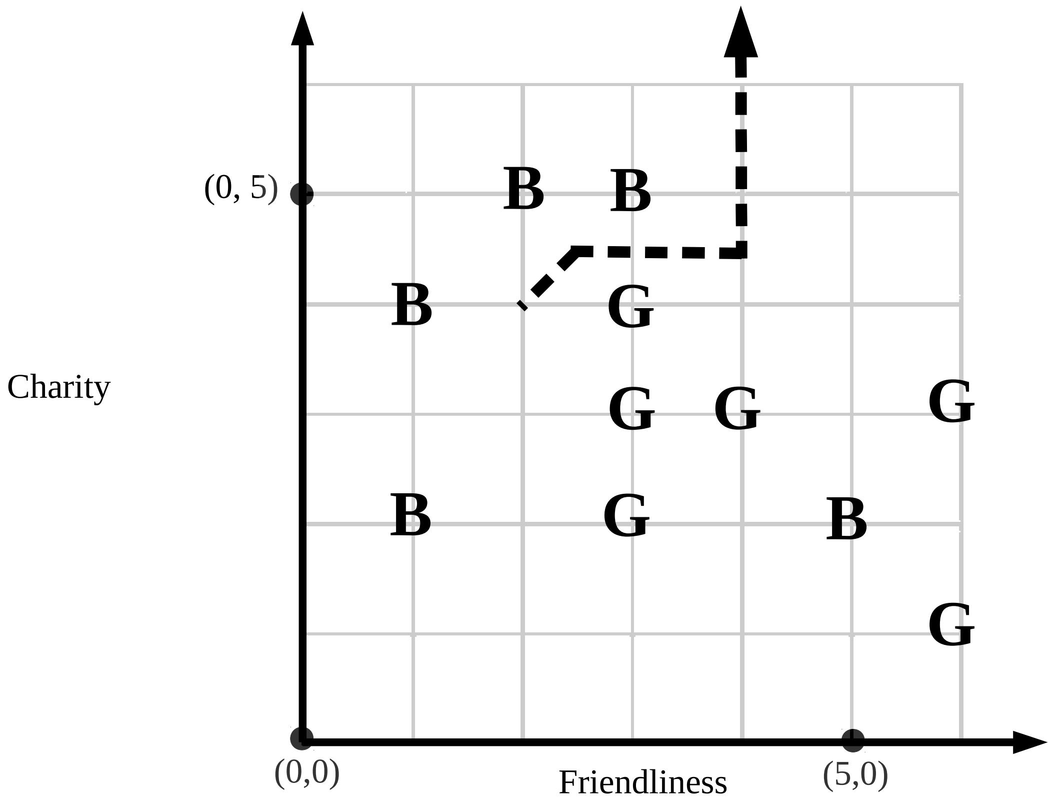
Euclidean	Good	Bad	Can't Tell
Manhattan	Good	Bad	Can't Tell
Hamming	Good	Bad	Can't Tell
Cosine	Good	Bad	Can't Tell

Part B: Who's Good, Anyway? (15 points)

Eleanor decides to use the 1-Nearest Neighbor algorithm with Euclidean distance on the graph below to classify people as good (G) or bad (B). Her Oracle provides her with a partial boundary, which is shown as the dashed line on the graph below.

B1 (3 points) Looking at the partial boundary, Eleanor notices that for the boundary to have that shape, a point must be missing! Where must the missing point be? If the point is for a good person, mark the location on the graph with a **G**. If the person is a bad person, mark the location with a **B**.

B2 (12 points) Complete the 1-Nearest Neighbors decision boundary on the graph below. Indicate with arrows where the decision boundary continues beyond the graph.



Problem 2: Identification Trees (33 points)

Part A: To Sport or Not To Sport (26 points)

For the longest time, your friend group has argued over whether or not Curling is a sport. With 6.034, you believe that you can use identification trees to end the debate!

You identify seven (7) games that you label as Sport or Not Sport. You then collect data about the games' **ACTIVITY** level**, TEAM** nature, and if they have **DIRECT** opponents. The data is shown below and on a tear-off sheet.

Game	Classification	ACTIVITY	TEAM	DIRECT
Baseball	Sport	Medium	Yes	Yes
Basketball	Sport	High	Yes	Yes
Football	Sport	High	Yes	Yes
Bobsled	Not Sport	Low	Yes	No
Golf	Not Sport	Medium	No	No
Archery	Not Sport	Low	No	Yes
Sync. Swim	Not Sport	Medium	Yes	No

A1 (12 Points) Calculate the average test disorder of each of the following feature tests. Use the table of logarithms on the next page to express your answer as **sums and products of decimals and fractions only.** Your final answer should contain **no logarithms.** *Space is provided on the next page for you to show your work.*

Feature Test	Disorder (as sums and products of decimals and fractions)
ACTIVITY	
TEAM	
DIRECT	

For partial credit, show your disorder calculation work here.

$-\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1$	$-\left[\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right] \approx 0.97$	$-\left[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right] \approx 0.9$
$-\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right] \approx 0.8$	$-\left[\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right] \approx 0.72$	$-\left[\frac{1}{6}\log_2\frac{1}{6} + \frac{5}{6}\log_2\frac{5}{6}\right] \approx 0.65$

A copy of this table is available on a tear-off sheet.

A2 (12 points) Based on the data above, construct a greedy, disorder-minimizing identification tree to correctly classify all the games according to their classification of **Sport** or **Not Sport**. In constructing your identification tree, choose from the following feature tests:

ACTIVITY, TEAM, DIRECT

Break ties in left-to-right order (ACTIVITY < TEAM < DIRECT).

Draw your ID tree here.

A3 (2 points) You and your friends think Curling is a low activity, team, direct sport.

Game	Classification	ACTIVITY	TEAM	DIRECT
Curling	**???**	Low	Yes	Yes

How would your ID tree (from part A2) classify Curling? (Circle one.)

Sport **Not Sport** **Can't tell**

Part B: A Reclassification (7 points)

B1 (5 points) Your friends decide that **Archery should actually be classified as Sport.** If you were to construct a new greedy, disorder-minimizing ID tree **using your original data (i.e., not including Curling)**, what would be the first test you would choose? (Circle one.)

ACTIVITY **TEAM** **DIRECT**

Note: You can do this problem without calculating disorder, by looking directly at the data.

For partial credit, show your work in the box at the bottom of this page.

B2 (2 points) How would your new ID tree classify Curling?

Sport **Not Sport** **Can't Tell**

For B2 partial credit, show your work here.

Problem 3: Constraints My Ninja Way! (40 points)

Five friends are students of the famous ninja Kakashi Hatake. To demonstrate their survival skills, Kakashi challenges the students to take a bell from his hand. The students want to approach their teacher as a group, and they decide to plan out their starting positions using 6.034 techniques.

The students' task is to assign each student to one of five positions in a line. The available positions are designated by the numbers **1 through 5**.

1	2	3	4	5

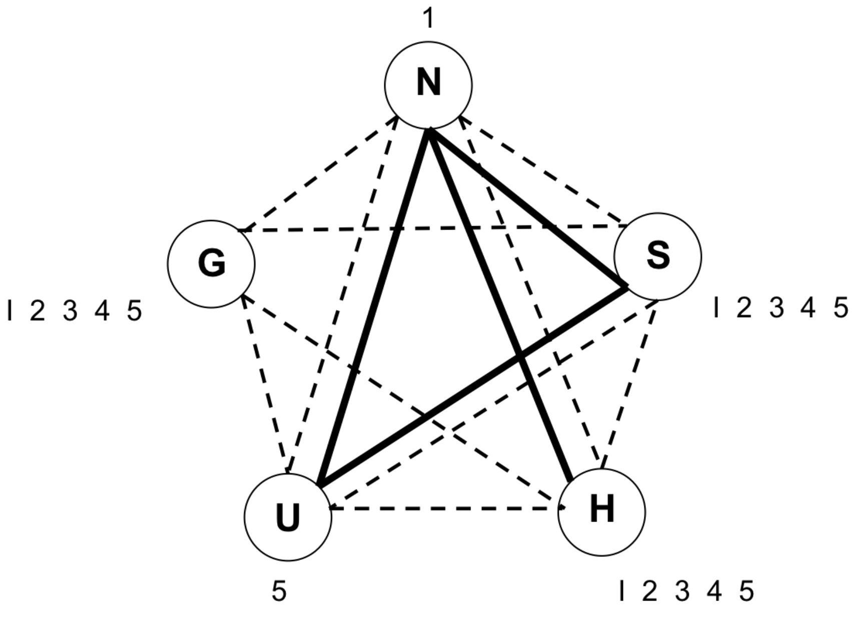
The five students are designated by an initial: **N, G, H, U, S**.

The students have the following **constraints**:

- **N** must be in position 1 because he thinks he is the best.
- **U** must be in position 5 to counter his rival **N**.
- **N**, **U**, and **S** must not be adjacent to each other, because they don't work well together.
- **H** must not be adjacent to **N**, because **H** has a secret crush on **N**.
- No two people can be in the same position.

The constraint graph is below.

- Solid lines represent the constraints for people who can't be adjacent to each other.
- Dashed lines represent the constraint that no two people can be in the same position.
- The domains have been initialized for you on the graph and in the accompanying table.



	Domain
N	1
G	1 2 3 4 5
H	1 2 3 4 5
U	5
S	1 2 3 4 5

Copies of the graph and domain table are available on tear-off sheets.

Part A: Forward Check or Fail Survival Test (22 points)

The team decides to try depth-first search with forward checking. They'll assign positions in the following order: **N, G, H, U, S**.

A1 (18 points) Perform **Depth-First Search with Forward Checking** (but no propagation) to find a solution.

★ For credit, show your work on the next page by both

- drawing the search tree **and**
- filling out the domain worksheet**.**

Domain worksheet instructions:

1. Every time you **assign a variable** or **remove a variable from the propagation queue** (if applicable), fill out a new row in the table. There may be more rows than you need.
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** (e.g. X). In the second column, list the **values that were just eliminated from neighboring variables** as a result (or "NONE" if no values were eliminated). Do not eliminate values from variables that have already been assigned.
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. With a propagation queue, if you add several variables to your queue at once, break ties by adding variables in alphabetical order (e.g., A before B).

Example row showing an assigned variable (with backtracking)

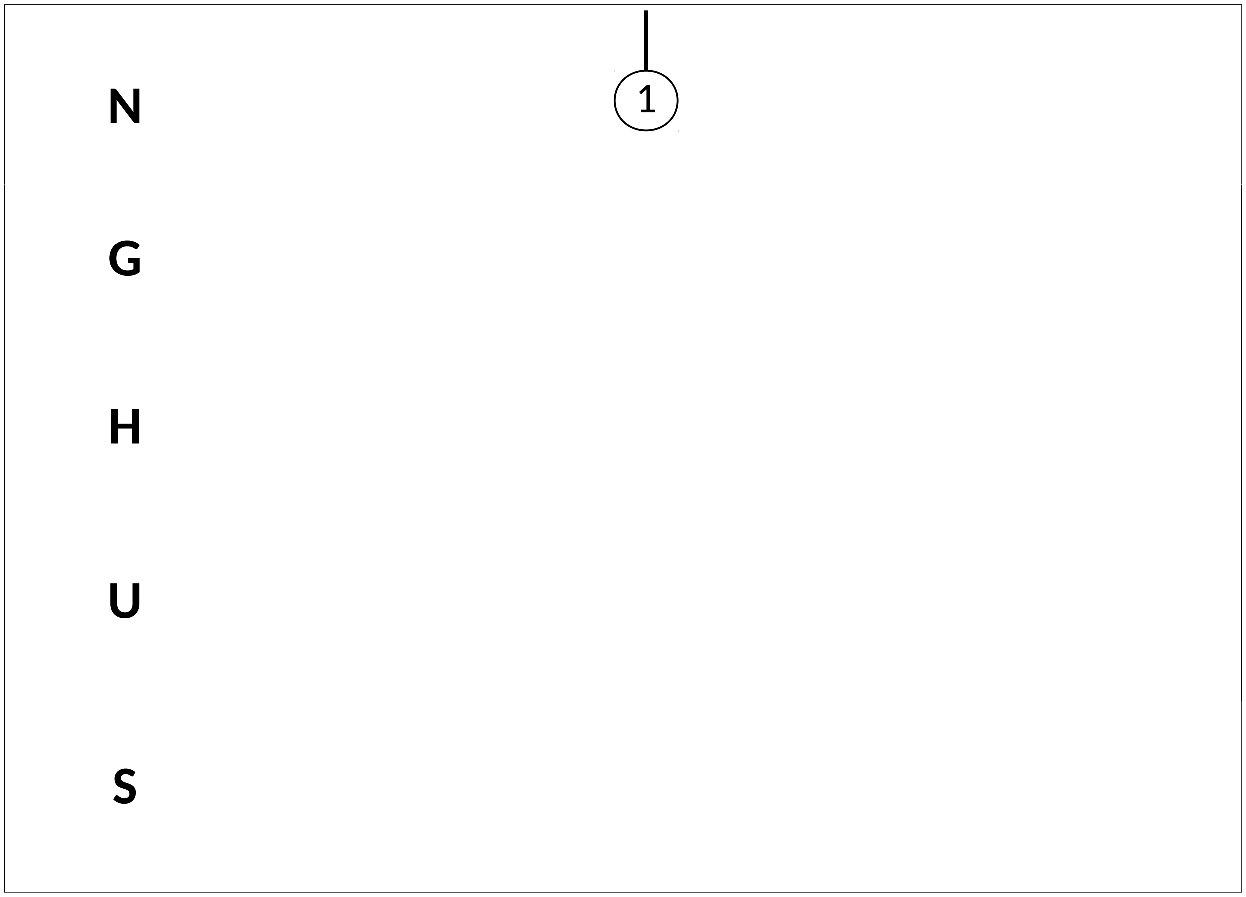
ex	X = 3	Y ≠ 3, 4 Z ≠ 3 (example)	☑

Example row showing a de-queued (propagated) variable

ex	X	W ≠ 1, 4 (example)	□

For your convenience, a copy of these instructions is provided on a tear-off sheet at the end of the quiz.

Draw your search tree for **Depth-First Search with Forward Checking** (DFS + FC) here.
If you want to start over, use next page. If you write on both pages, clearly mark which page we are to grade.

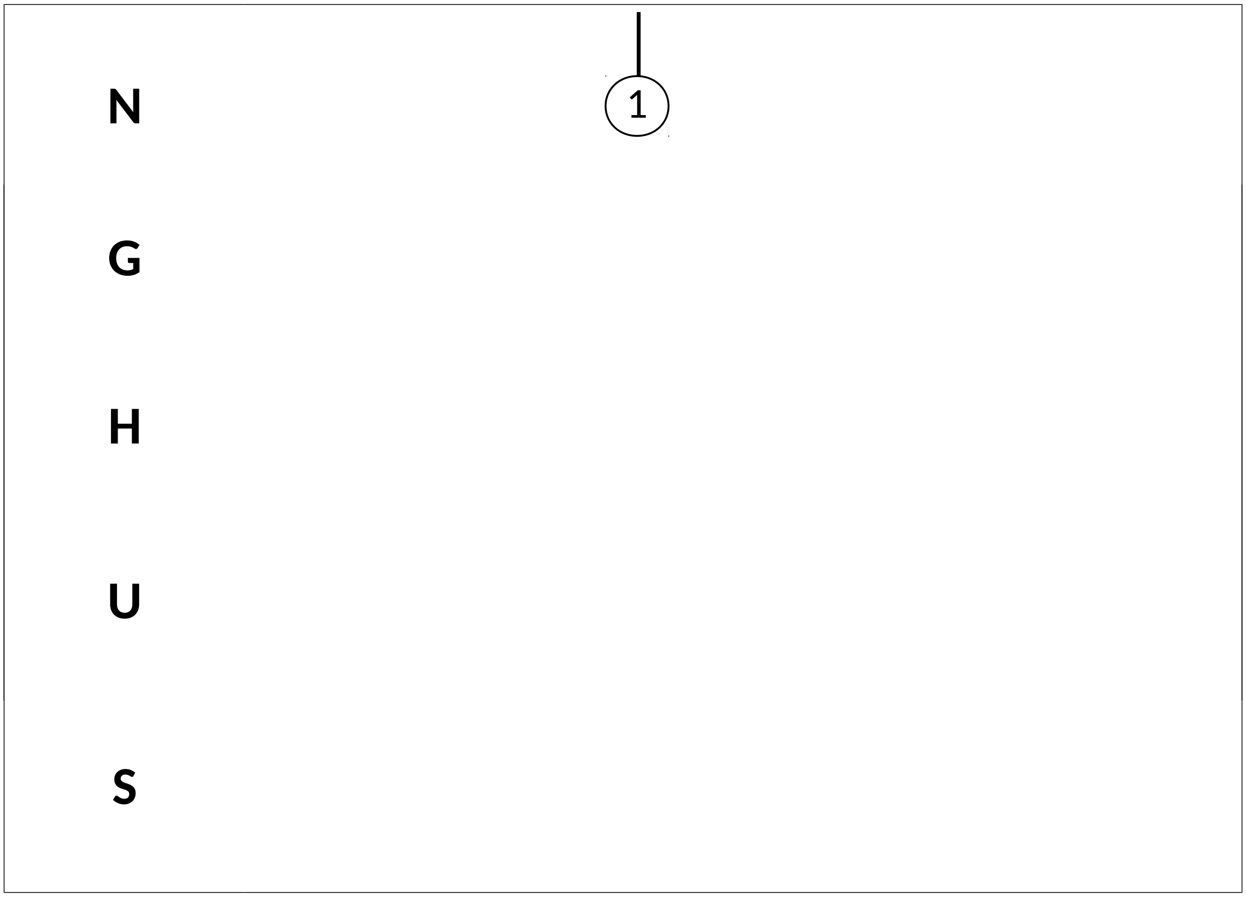


Domain Worksheet

	Variable assigned	List all values just eliminated from neighboring variables or NONE	Backtrack?
1			□
2			□
3			□
4			□
5			□
6			□
7			□
8			□
9			□
10			□

This page is a duplicate copy for Part A1 (DFS+FC). If you want this copy graded instead, check the box:

□ **I want to start over; grade this page.**



Domain Worksheet

	Variable assigned	List all values just eliminated from neighboring variables or NONE	Backtrack?
1			□
2			□
3			□
4			□
5			□
6			□
7			□
8			□
9			□
10			□

A2 (2 points) How many variable assignments did you perform using the students' ordering of N, G, H, U, S?

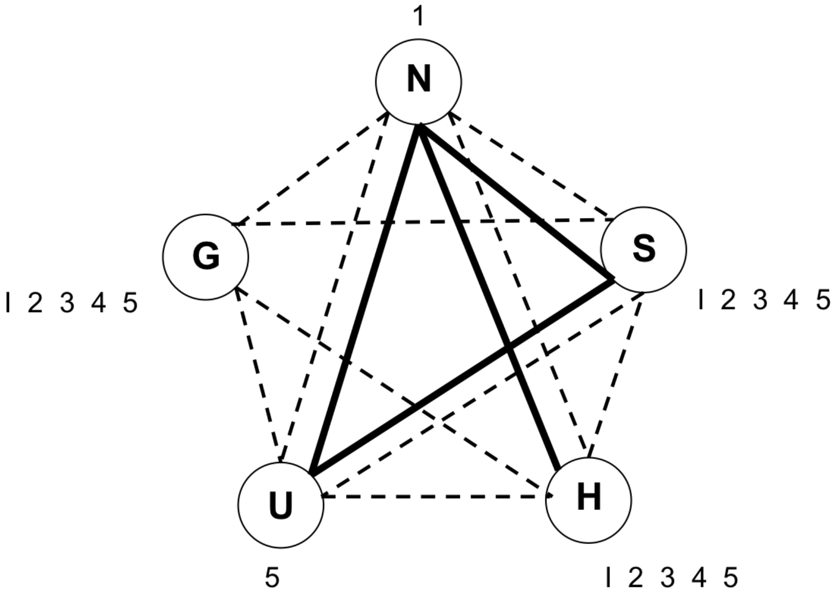
A3 (2 points) The students wonder if an ordering other than N, G, H, U, S would result in fewer assignments when performing the algorithm. List an ordering that would result in fewer assignments or write NONE if no such ordering is possible.

Part B: Enter Domain Reduction! (18 points)

The students think that DFS+FC took too long. They wonder if domain reduction before search would improve performance.

Perform **Domain Reduction** on the constraint graph by filling out the worksheet below. Initialize the propagation queue with variables in alphabetical order: **G, H, N, S, U.** Propagate through any variable once its domain has been reduced.

For your convenience the constraint graph and domain table are repeated here and on a tear-off sheet at the end of the quiz.



	Domain
N	1
G	1 2 3 4 5
H	1 2 3 4 5
U	5
S	1 2 3 4 5

Fill in this worksheet with your **Domain Reduction** (DR) steps, initializing your propagation queue with variables **in alphabetical order**: **G, H, N, S, U.** There may be more rows than you need.

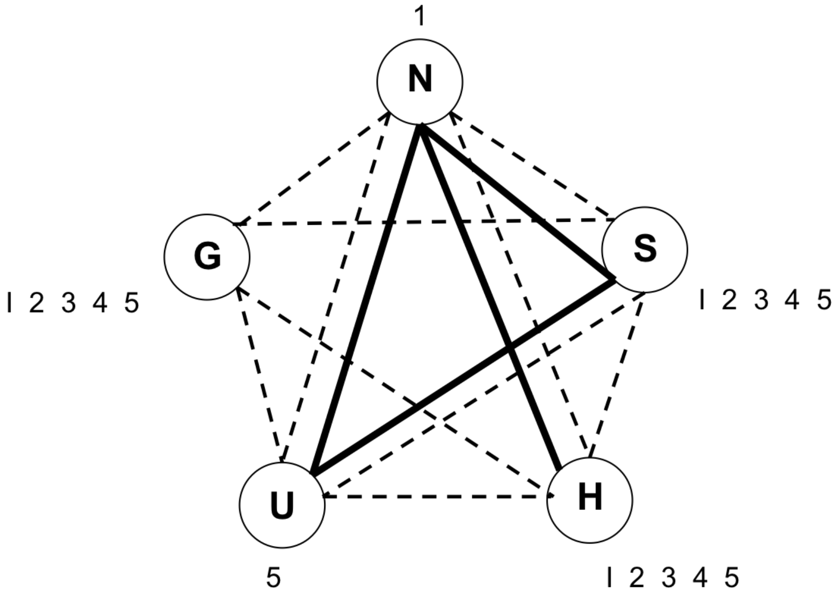
Domain Worksheet

If you want to start over, use next page. If you write on both pages, clearly mark which page we are to grade.

	Variable de-queued	List all values just eliminated from neighboring variables or NONE
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

This page is a duplicate copy for Part B1 (DR). If you want this copy graded instead, check the box:

□ I want to start over; grade this page.



	Domain
N	1
G	1 2 3 4 5
H	1 2 3 4 5
U	5
S	1 2 3 4 5

Fill in this worksheet with your **Domain Reduction** steps, initializing your propagation queue with variables **in alphabetical order**: **G, H, N, S, U.** There may be more rows than you need.

Domain Worksheet

	Variable de-queued	List all values just eliminated from neighboring variables or NONE
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

Problem 4: Spiritual and Right Now (6 points)

For each question, **write in the box** provided the letter corresponding to the **one** best answer **and circle** the answer. There is **no penalty for wrong answers**, so it pays to guess in the absence of knowledge.

□ 1. Raibert exhibited a robot that, amazingly:

- **a.** Operates a jack hammer.
- **b.** Mixes a martini.
- **c.** Builds a copy of itself.
- **d.** Changes diapers.
- **e.** Is not listed here; it's a secret.

□ 2. Boyden demonstrated that you can use the following material to expand ("blow up") the brain:

- **a.** Nano-particles
- **b.** Elmer's glue
- **c.** Mozzarella cheese
- **d.** Long chained fatty acids
- **e.** Diaper polymers

□ 3. Winston argued that convolution for deep neural networks should be referred to as:

- **a.** Fourier Transform, because you are combining signals.
- **b.** Rotating, because you rotate the regions of focus in a matrix.
- **c.** Windowing, because you focus on a specific window, or subregion, of the matrix.
- **d.** Combining, because you combine values in the matrix.

□ 4. Winston summarized AlphaGo search as a balance between:

- **a.** Heuristics and brute force.
- **b.** Evaluation and exploration.
- **c.** Black box calculation and explainability.
- **d.** Symbolic reasoning and statistics.

□ 5. Winston claimed that the best way to learn the difference between *bah* and *gah*, or more formally, to learn how to pronounce sounds in a new language is to:

- **a.** Listen to audiotapes, because you can easily and repeatedly listen to specific pronunciations.
- **b.** Read books in the language so that you can see how words are spelled out.
- **c.** Watch a native speaker speak so that you can see how they move their lips.
- **d.** None of the above; there is no good way to learn as an adult.

□ 6. Marr's approach to problem solving emphasized:

- **a.** Applying successful methods to as wide of a variety of problems as possible for efficient and prolific results.
- **b.** Ignoring implementation so the solution remains as general and far-reaching as possible.
- **c.** Understanding the problem in order to develop a method that best fits the problem you are trying to solve.
- **d.** User studies, as they are the only way to determine if your solution makes sense in the real world.

This page is intentionally blank.

Tear-off sheet

We do not collect tear-off sheets, so show your work on the quiz pages, not the tear-off sheets.

Problem 2 (ID Trees)

Game	Classification	ACTIVITY	TEAM	DIRECT
Baseball	Sport	Medium	Yes	Yes
Basketball	Sport	High	Yes	Yes
Football	Sport	High	Yes	Yes
Bobsled	Not Sport	Low	Yes	No
Golf	Not Sport	Medium	No	No
Archery	Not Sport	Low	No	Yes
Sync. Swim	Not Sport	Medium	Yes	No

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

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

Tear-off sheet

We do not collect tear-off sheets, so show your work on the quiz pages, not the tear-off sheets.

Problem 3 (Constraints)

Domain worksheet instructions:

1. Every time you **assign a variable** or **remove a variable from the propagation queue** (if applicable), fill out a new row in the table. There may be more rows than you need.
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** (e.g. X). In the second column, list the **values that were just eliminated from neighboring variables** as a result (or "NONE" if no values were eliminated). Do not eliminate values from variables that have already been assigned.
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. With a propagation queue, if you add several variables to your queue at once, break ties by adding variables in alphabetical order (e.g., A before B).

Example row showing an assigned variable (with backtracking)

ex	X = 3	Y ≠ 3, 4 Z ≠ 3 (example)	☑

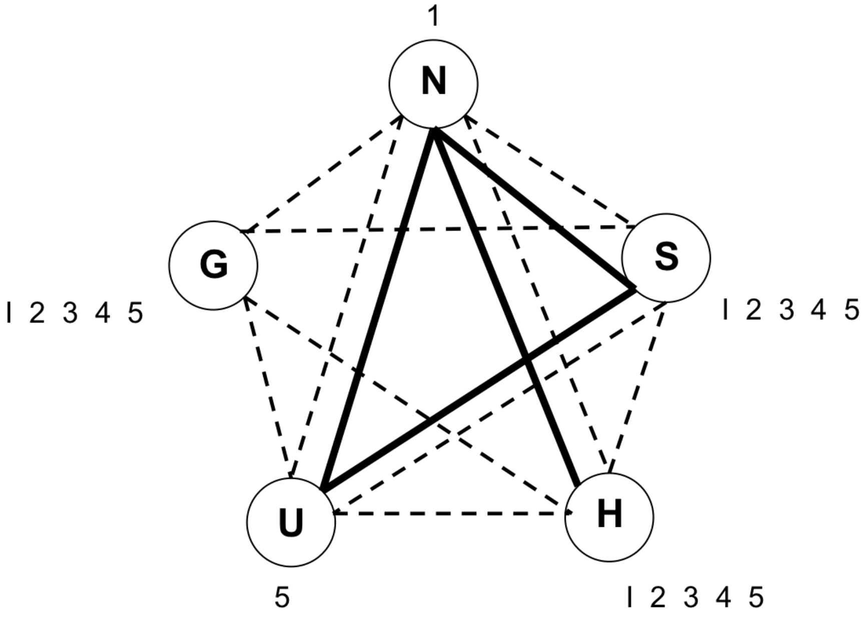
Example row showing a de-queued (propagated) variable

ex	X	W ≠ 1, 4 (example)	□

Constraint graph:

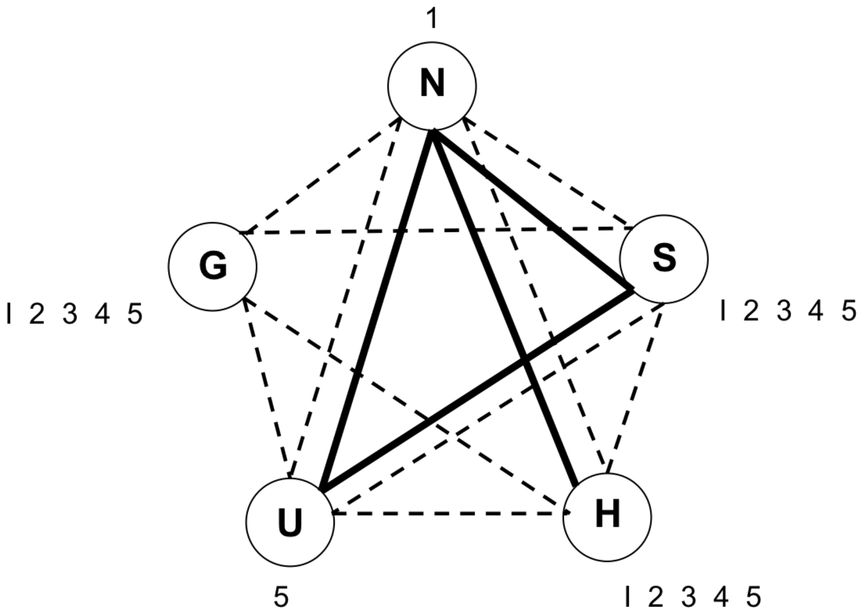
The constraint graph is below.

- Solid lines represent the constraints for people who can't be adjacent to each other.
- Dashed lines represent the constraint that no two people can be in the same position.
- The domains have been initialized for you on the graph and in the accompanying table.

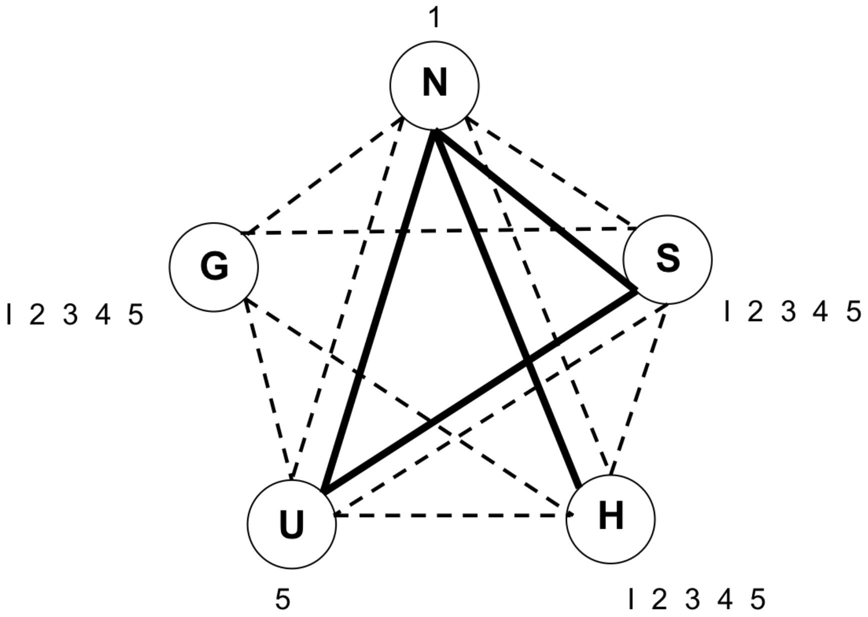


	Domain
N	1
G	1 2 3 4 5
H	1 2 3 4 5
U	5
S	1 2 3 4 5

Additional copies of the graph and table are on the next page.



	Domain
N	1
G	1 2 3 4 5
H	1 2 3 4 5
U	5
S	1 2 3 4 5



	Domain
N	1
G	1 2 3 4 5
H	1 2 3 4 5
U	5
S	1 2 3 4 5