

CS101
Fall 2018

Instructor: Ashley Taylor, TA: Shreya Shankar
October 30, 2018

Midterm Exam SOLUTIONS

NAME (LAST, FIRST): _____

SUNET: _____@stanford.edu

Problem	1	2	3	4	TOTAL
Score					
	15	15	12	23	65

Instructions:

- The time for this exam is **1 hour and 20 minutes**, 80 minutes total. There are 65 points total, so you should spend a little more than 1 minute per point, though your pacing may be different on different problems.
- You are only allowed a pencil, pen, and eraser. Any other materials are strictly prohibited. You may not use any digital devices other than a clock for timing. You may ask the proctors for blank scratch paper.
- A **reference sheet** is included as the last page. You may detach it for your reference during the exam.
- For coding problems, we will forgive minor syntax errors (such as missing semicolons, etc.). **We will give partial credit for solutions that are on the right track or correctly solve part of the problem.**
- For any problems involving calculations, we will accept an expression (that could be plugged into a calculator) in lieu of the numerical answer.

Please sign *before* beginning the exam:

I agree to abide by the spirit and letter of the Honor Code, and to follow the instructions above.

(Signature)

(Date)

1. Hardware/Software

- a) (2 points) You're faced with the option of a computer with more RAM or a computer with more disk storage. Which computer would you recommend to a lawyer who needs to store case files containing videos, audio files, and thousands of documents? Please provide at least one sentence of justification.

More disk storage -- disk storage permanently stores data, as opposed to RAM. More persistent storage is ideal to store lots of files.

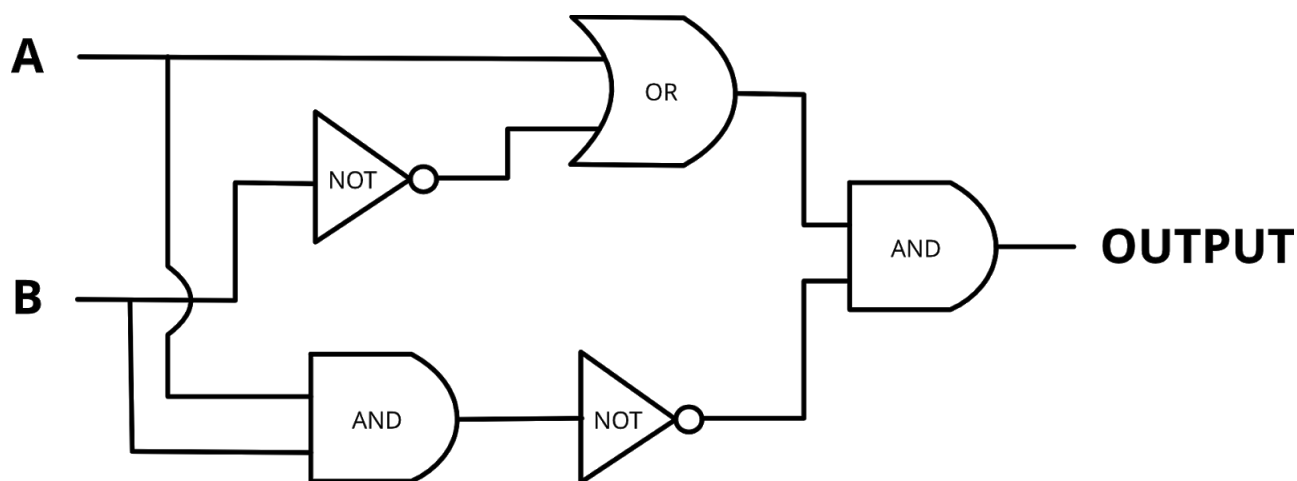
- b) (5 points) Describe the process of running code. Your description should discuss programming languages, architecture, and computer components.

Programmers write code in a language of their choice, such as JavaScript. This code is compiled and converted into assembly and machine code based on the architecture of the computer.

- c) (4 points) Describe two functions of operating systems. Name two operating systems that you currently use or have used in the past.

Operating systems: 1) manage shared resources such as RAM and CPU, 2) start and stop applications. Examples: Android, iOS.

- d) (4 points) Identify which inputs make the following circuit output true, and which inputs make the following circuit output false.



True: A = 0, B = 0; A = 1, B = 0

False: A = 1, B = 1; A = 0, B = 1

2. Potpourri

- a) (3 points) Name and describe an algorithm used in AI. Your description doesn't need to be very detailed; a few sentences will suffice.

Linear regression: finds a line of best fit through data, and uses this line to make predictions for new data points.

Clustering/K-Means: predicts the value for new data points by looking at the closest neighbors. An example of this is predicting where you live based on your three closest friends on Facebook.

- b) (2 point) Describe a pitfall of artificial intelligence. This pitfall can be general or a specific case of AI performing poorly.

Machine learning systems are vulnerable to adversarial examples, or data designed to fool the system (like the patch for computer vision models).

Face classifiers do not work well for several groups of the population.

Reward hacking: AI finding an unwanted / "hacky" solution to a problem.

- c) (3 point) You're choosing between two phones. One has an OLED screen and the other has an LCD display. Which phone would you choose, and why? You can assume that the hardware, size, operating system, and other software/hardware components are identical. We will evaluate your justification, not your choice.

We accepted many answers. Some examples:

- OLED because you can see "true black" pixels
- LCD because it is cheaper

- d) (3 points) Identify the two types of compression discussed in class and describe their distinguishing characteristic. Additionally, give an example of a product that uses a compression technique.

Lossy and lossless compression. Lossy compression “forgets” some of the data as the data is compressed, but lossless compressed data can be perfectly reconstructed to the original data.

An example of a product that uses a compression technique is the photos app (compresses as PNG or JPEG). We accepted many answers.

- e) (4 points) We talked about errors in computer addition of both integers and decimal numbers. Give an example of an integer addition problem that would have an error (and describe the error). Give an example of a decimal number addition problem (and describe the error) [hint: we saw an example of errors with decimal numbers in Google Sheets as well as in JavaScript].

Integer addition error: adding two 4-byte ints that causes an “overflow” (ex: Gangnam Style)
Decimal addition problem: adding two numbers with long decimals, such that the resulting sum does not have enough bits to store the entire decimal

3. Code Analysis Questions

a) (3 points) You notice your code isn't performing as you'd like. Describe two different approaches to determine what's wrong.

1. Look for logic or syntax bugs / read code
2. Read error messages
3. Test code piece-by-piece to make sure each part is working
4. Use print statements to debug code
5. Use browser debugger

b) (3 points) How does threshold impact the final image for the edge detection algorithm? Be specific. In your answer, you should describe how increasing the threshold affects the final image.

Increasing threshold decreases the number of edges (more white than edge pixels)

- c) (3 points) You'd like to replace the sky in an image with Paris and notice that only some of the sky was replaced. What change should you make to your code?

Increase green screen threshold

- d) (3 points) You made the change to your green screen code from the previous question, but now you notice that some of the sky remains while some non-sky parts of the image have been replaced. What change should you make to your code?

Limit the area of the image that is replaced with green screen image (using if statement)

4. Code and Spreadsheets

- a) (5 points) Below is a description of columns in a spreadsheet of Fandango ratings. Describe how you would find the **total** number of votes for films with at least 3 stars. Then describe how you would find which star rating has the highest average rating on Fandango.

Column	Definition
FILM	The movie
STARS	Number of stars presented on Fandango.com (between 0 and 5)
RATING	The Fandango ratingValue for the film, as pulled from the HTML of each page. This is the actual average score the movie obtained.
VOTES	number of people who had reviewed the film at the time we pulled it.

We do not expect you to provide actual code or function calls, but you may use any of the following functions in your answer. You should describe which function(s) you'd use and which columns/conditions you'd use.

COUNTUNIQUE(range)

UNIQUE(range)

SUM(range)

COUNTIFS(condition_range, condition)

SUMIFS(range, condition_range, condition)

MINIFS(range, condition_range, condition)

MAXIFS(range, condition_range, condition)

To find the total number of votes for films with at least three stars:

SUMIFS(VOTES, STARS, ">=3")

To find the star rating has the highest average rating on Fandango, look at which of the following has the greatest value:

AVERAGEIFS(RATING, STARS, 0)

AVERAGEIFS(RATING, STARS, 1)

AVERAGEIFS(RATING, STARS, 2)

AVERAGEIFS(RATING, STARS, 3)

AVERAGEIFS(RATING, STARS, 4)

AVERAGEIFS(RATING, STARS, 5)

b) (8 points) Identify one logic and one syntax bug in the following code. You should clearly identify which is which, then correct the bugs you identify.

```
img = new SimpleImage("poppy.jpg");
result = new SimpleImage("poppy-blur.jpg");
result.setAsBig(img);
result.setZoom(0.5);

// my implementation of the blur algorithm from class is below.
// But I think I have some bugs.

for (pixel : img) {
    redSum = 0;
    greenSum = 0;
    blueSum = 0;
    for (neighbor : img.getNeighbors(pixel)) {
        redSum = neighbor.getRed(); // should be redSum = redSum +
neighbor.getRed();
        greenSum = neighbor.getGreen(); // should be greenSum = greenSum +
neighbor.getGreen();
        blueSum = neighbor.getBlue(); // should be blueSum = blueSum +
neighbor.getBlue();
    }
    redAvg = redSum / img.countNeighbors(pixel);
    greenAvg = greenSum / img.countNeighbors(pixel);
    blueAvg = blueSum / img.countNeighbors(pixel);

    newPixel = result.getPixel(pixel.getX(), pixel.getY());
    newPixel.setRGB(redAvg, greenAvg, blueAvg);
}

print(result);
```

- c) (10 points) Write code to change the 3x3 square at the bottom left corner of an image to "turquoise-scale". Recall that grayscale renders the average brightness into gray; instead, we'll only set the blue and green values to have the average brightness value, with the red value as 0. Your code should work on images of any size. For full credit, you should not use any if-statements.

```
img = new SimpleImage("x.png");  
img.setZoom(0.5);  
// write your answer below
```

```
pixel = image.getPixel(1, image.getHeight() - 2);  
  
for (neighbor : image.getNeighbors(pixel)) {  
    avg = (neighbor.getRed() + neighbor.getBlue() +  
          neighbor.getGreen()) / 3;  
    neighbor.setRGB(0, avg, avg);  
}
```

```
    print(img);
```

```
pixel.getX()
pixel.getY()
pixel.getRed()
pixel.getGreen()
pixel.getBlue()
pixel.setRGB(red, green, blue)
pixel.isSimilarTo(red, green, blue, threshold)

image = new SimpleImage("image.png")
image.countNeighbors(pixel)
image.getHeight()
image.getWidth()
image.getPixel(row, column)

for (pixel : image) {
    // your code here
}

for (neighbor : image.getNeighbors(pixel)) {
    // your code here
}

if (condition) {
    // your code here
} else {
    // your code here
}

&& => and
|| => or
! => not
!= => is not equal
== => is equal

1TB = 1000GB
1GB = 1000MB
1MB = 1000KB
```

1KB = 1000 bytes

Bit: 0 is "off", 1 is "on"