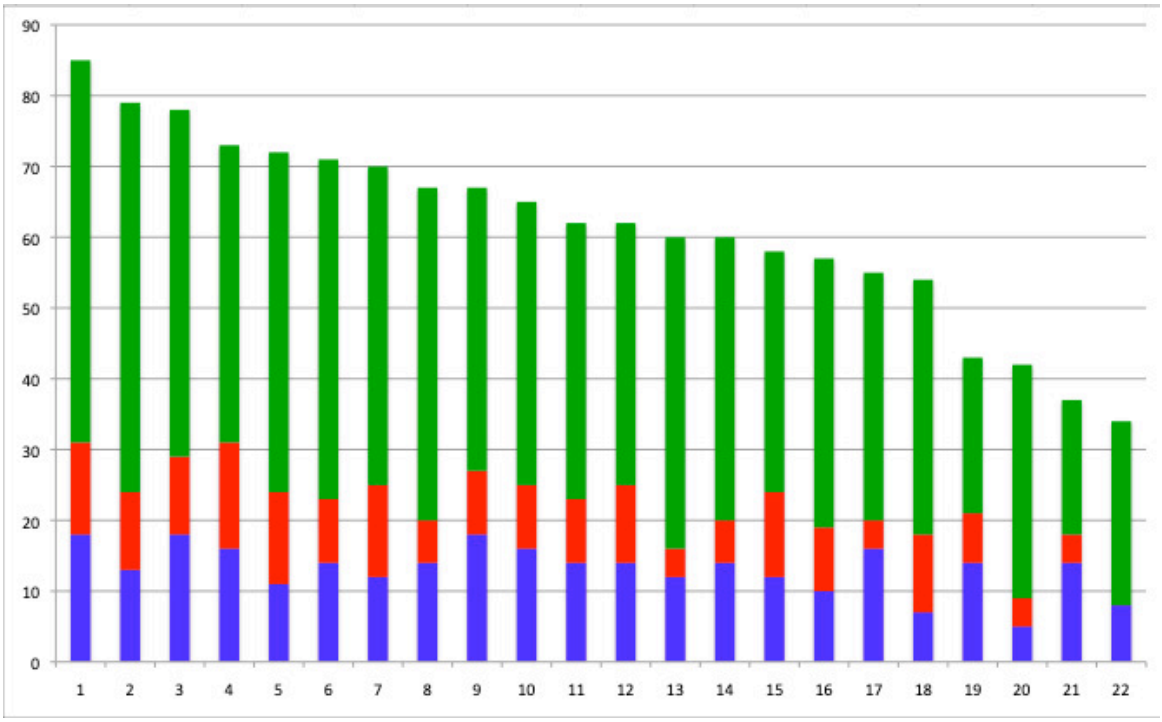
COS 109 Midterm Exam, Fall 2020 Due 9:00 PM ET, Friday, October 9

I graded this myself. Thanks to everyone for soldiering through under adverse circumstances both in taking the exam and in returning the results.

The class is about half the size of last year and the circumstances are wildly different, so I'm not sure whether there is anything to infer from a comparison, but for whatever it's worth, last year's median, lower quartile and upper quartile were 63, 50 and 70. This year the corresponding numbers were 62, 57, and 71. The kinds of things that people had trouble with are similar, however: how to represent things in bits and bytes, notations like hexadecimal, and generally anything that involves arithmetic in bases other than 10 (and some problems with base 10 too). Many are unsure about the Toy machine. The three values in each column below are for parts 1, 2 and 3, reading from the bottom.



1. (20 points, 2 each) Short Answers. Circle the right answer or write it in the space provided.

(a) Unix systems distinguish upper case letters from lower case letters in filenames: pset3.pdf, Pset3.pdf and PSET3.PDF are different names. How many different ways are there to write the filename **`pset3.pdf`** in mixtures of upper and lower case letters?

128 = 2^7. Each letter can be independently upper or lower case. Not well handled on average.

(b) The first character of the Unicode code chart for emoticons (emoji) has code **`1F600`** and the last character has code **`1F64F`**. How many characters are there in the emoji code chart?

80. The first is 00 and the last is 4F (79 decimal). You have to include them all; failing to do that led to 79, the common wrong answer. Pretty similar to questions on previous midterms, and covered in the Q/A.

(c) In the following list of 24-bit RGB colors, expressed in hexadecimal, which one has the least amount of green?

(The character **0** is a zero.)

ACCEDE **BEADED** **B0BBED** **C0FFEE** **DEC0DE** **EFFACE** **FACADE**

BEADED. The green component starts with A; all others are B or larger. No arithmetic needed. I hope those who attended the Q/A recognized this question.

(d) The speeds of supercomputers are measured in floating-point operations per second, or "flops". Which one of these would be the most representative speed for the fastest of today's supercomputers?

500 Mflops **500 Gflops** **500 Tflops** **500 Pflops** **500 Eflops** **500 Zflops** **500 Yflops**

500 Pflops.

(e) Modern computers can efficiently process integers of several sizes, usually 1, 2, 4, 8, and sometimes 16 bytes long. Which of these is the least number of bytes that could be used for storing a binary number that represents the population of the European Union?

1 **2** **4** **8** **16** **none are big enough**

4. The EU is about 700M people? It doesn't matter; 2 bytes is not enough and 4 is sufficient for the whole world.

(f) Famous Stanford computer scientist Don Knuth said, "I have just celebrated my 10000th birthday (in base 3)." How old is Don (in decimal)?

81 = 3^4. Stanford people are special, for sure, but it seems unlikely that a famous CS professor there would be only 9 years old.

(g) "All the devices are individually identified. There will be no duplication unless you have 8 million kids within 300 feet." (from a newspaper article describing a device to keep track of kids at amusement parks.) How many bits is it likely that the device identifier uses?

23. 20 bits is a million; this is 8 times as big, so 3 more bits.

(h) What is the decimal value of the binary number **1010.11**?

10.75. Fractions are negative powers of 2, like 2^-1 and 2^-2.

(i) A secretive spy agency stores the name, address, phone number and social security number or equivalent for every person in the world. *Very roughly*, how many gigabytes would be required to hold all this information?

1000. 100B per person x 10^10 people? There was a worrying amount of confusion about bits and bytes in some answers, and occasionally way too much precision.

(j) Computer pioneer Grace Hopper (1906-1992) said, "The instruction code should use symbols which are easily learned and identified with the operations by already existing mental associations: 'a' for add, etc. Replacing a sequence of binary numbers with a single letter to represent an operation [simplifies] the coding process and makes it much more intuitive for users." What kind or level of programming language is Hopper describing? One or two words is enough.

assembly language. Hopper was indeed involved with Cobol, a high level language, but this is all about assemblers; note words like "instruction" and "operation".

2. (15 points) Machines

Here is a program in the Toy assembly language, with reminders about what the instructions do.

```
Foo   GET           get a number from keyboard into accumulator
      IFZERO Bar    if accumulator is zero, go to Bar
      LOAD   1      load 1 into accumulator
      ADD    Sum    add value in location Sum to value in accumulator
```

```
        STORE   Sum     store accumulator value in location Sum
        GOTO    Foo     go to instruction labeled Foo
Bar     LOAD    Sum     load value in location Sum into accumulator
        PRINT           print contents of accumulator
        STOP
Sum     0    reserve a memory location called Sum, set its initial value to 0
```

(a) If this program is given the sequence of inputs **3 –1 4 1 –5 9 2 –7 0** ***exactly*** what does it print?

8. It's just counting the inputs.

(b) It is possible to simulate a GOTO instruction with a sequence of two other Toy instructions. Write out one such sequence.

LOAD 0; IFZERO …

IFPOS …; IFNEG …

And variants.

(c) Imagine that Alan Turing and John von Neumann are having an argument. Turing says "I can simulate *any* of your computers on my Turing machine." Von Neumann replies "So what? I can simulate your silly Turing machine on *all* of my computers." Who is right?

neither one **only Turing** **only von Neumann** **both**

both.

(d) In his 1946 paper, John von Neumann said "We are therefore forced to recognize the possibility of constructing a hierarchy of _____, each of which has greater capacity than the preceding but which is less quickly accessible." Which of the following is the proper word to fill in the blank

accumulators **controls** **instructions** **memories** **orders** **organs** **processors**

memories. A gift; it was on a previous midterm.

(e) Von Neumann also said "It is convenient to group the binary digits into tetrads, groups of 4 binary digits." What synonym or alternative terminology might be used today instead of tetrads?

hexadecimal

3. (55 points, 5 each) Miscellaneous

(a) A *NY Times* story on 8/5/08 about horse-doping drugs referred to a non-existent unit of mass called a "petragram." Perhaps they meant picograms; a petagram would be a lot of drugs.

(i) How many picograms are there in a petagram, as a power of 10?

10^27 (12+15). For those unfamiliar with pico, it's in the table on the first page of the glossary.

(ii) What power of two is nearest to this power of ten?

2^90

(b) Last month I did some experiments with a new algorithm. The measured running times were 15.0 milliseconds to process 4 items, 30.1 msec for 5 items, 59.9 msec for 6 items, and 120.2 msec for 7 items.

(i) Given these times, what is the likely running time for 10 items?

960.

(ii) Which of these is the most likely description of how the algorithm's running time grows in proportion to the number of items?

logarithmic **linear** **n log n** **quadratic** **cubic** **exponential** **none of these**

exponential. It's doubling each time.

(c) The hexadecimal value **FF0100** can be interpreted as an RGB color, but it is really just a 24-bit integer. Suppose that we subtract 1 from this integer value, that is, compute **FF0100 – 1**.

(i) What is the resulting value in hexadecimal?

FF00FF. Much confusion about what hex numbers mean here, and how to do arithmetic.

(ii) If the resulting value is interpreted as RGB, which of these colors is it closest to?

red **green** **blue** **yellow** **cyan** **magenta** **black** **white**

magenta

(d) Suppose that two adjacent bytes in RAM contain the binary values 01010101 and 01101111. For each of the following, indicate whether those two bytes could represent the given item, or could not possibly represent it.

two ASCII characters	**could**	**could not**
a Princeton University student id number	**could**	**could not**
a 16-bit integer with value less than 512	**could**	**could not**
part of a JPEG image of your best friend	**could**	**could not**
part of an instruction in the Safari browser program	**could**	**could not**

could be ascii, and indeed are – look at the table at the end for 55 and 6F.

could not. An id is 9 decimal digits, so roughly 10^9, which doesn't fit in 16 bits.

could not. The number would be at least 2^14 because that bit is on (in both bytes so order doesn't matter).

could. Any bit pattern is possible in JPEG**.**

could. Same for instructions.

(e) Let's play the guessing game "I'm thinking of a number between 1 and N": I think of a number and you try to guess it; after each guess, I tell you whether your answer was too high, too low, or correct. Suppose I cheat: I give you answers that will ultimately prove true, but I don't actually select my number until I'm forced to, when the range has been narrowed down to only a single value.

(i) If you use the best possible strategy to choose your guess each time, about how many guesses are needed if N is 1 million?

20. It doesn't matter what your opponent does; optimal play still divides the possible range in half with each guess.

(ii) How does the maximum number of guesses vary in relation to N?

log N

(f) Suppose that you want to encode certain information about current Princeton undergrads *in as few bits as possible*. The information for each person is: living on-campus or not, birthday (like October 9), age (assume everyone is between 17 and 24 inclusive), and class year (2021 through 2024). What is the *minimum* number of bits you need per person, and why?

15 = 1 + 9 + 3 + 2. Again, frequent confusion about how to encode information in bits.

(g) A blog post says that there were 100 million Alexas at the start of 2019 and 200 million at the start of 2020. Assuming (most improbably!) that this represents a smooth exponential growth that will continue into the future,

(i) What is the percentage rate of increase of the number of Alexas each month?

6% (doubling in 12 months; use the rule of 72)

(ii) During what year will the number of Alexas first exceed the number of people in the world?

2026. It's doubling every year, so it has to get over about 8B. It's a bit unclear about the stopping criterion so graded generously.

(h) Many years ago, Pat Programmer wrote a C program for a computer that now no longer exists. She still has both the C source code and the compiled code for the ancient computer. She wants to work on the program again, but will have to run it on her brand new laptop. For each of the following, circle the most appropriate answer.

She could run the original compiled code, unchanged, on her laptop	**likely**	**unlikely**
She could compile the C program and run that compiled code on the laptop	**likely**	**unlikely**
She could write a simulator in C for the old computer, and run it on the laptop	**likely**	**unlikely**
She could run the original compiled code on this simulator on the laptop	**likely**	**unlikely**
The simulated computer could run faster than the old physical computer it simulates	**likely**	**unlikely**

The new laptop will have a new instruction set so the old bits won't work. Compiling for the new machine will work, as will writing a simulator. And new machines are way faster than ancient ones.

(i) The home page for COS 109 contains half a dozen occurrences of a small bitmap, each referenced by the tag **`<img src="new.gif">`** at the appropriate place on the page.

(i) How many times is the image likely to be downloaded when you first visit the page, and why?

1. It would be cached on first access.

(ii) If you revisit the page five minutes later, what is the most likely additional number of times the image will be downloaded, and why?

0. That's what caches are for.

(j) Random quickies:

Alan Turing got his PhD at the University of Cambridge	**true**	**false**
John von Neumann is buried in Princeton	**true**	**false**
Tony Hoare received a Turing Award for inventing binary search	**true**	**false**
An assembler has to be written in assembly language	**true**	**false**
Babbage debated Leibnitz about whether to use binary arithmetic for computation	**true**	**false**

We did refer to Turing *38 a fair number of times, even with pictures of his entry at the Grad College.

I showed pictures of von Neumann's headstone in Princeton Cemetery.

Hoare did not invent binary search; it's ancient.

Among other things, the Toy machine, an assembler of sorts, is in JavaScript.

Leibnitz died about 75 years before Babbage was born, which precludes personal communications.

(k) The picture on the left is a close-up of a seriously geeky t-shirt from Thinkgeek.com. ***Exactly*** what does it say?

```
0100100001100101
0110110001101100
0110111100100000
0101011101101111
0111001001101100
0110010000100001
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

`Hello World!` Case matters; note that "***exactly***" is in underlined bold italic.