Learning Goals

- Introduction & Representing objects as strings
- Defining computation: Boolean circuits and straightline programs
- Completeness: [illegible] function

Policies

See the course syllabu[illegible]

- **Collaboration:** You can collaborate with other students who are currently enrolled in this course (or, in the case of homework zero, planning to enroll in this course) in brainstorming and thinking through approaches to solutions but you should write the solutions on your own, without sharing notes.
- **Owning your solution:** Always make sure that you "own" your solutions to problem sets. That is, you should always first grapple with the problems on your own, and even if you participate in brainstorming sessions, make sure that you completely understand the ideas and details underlying the solution. This is in your interest as it ensures you have a solid understanding of the course material and will help in the midterms and final. Especially given the generous bonus points policies for homework, **getting 80% of the problem set questions right on your own will be much better to both your understanding and your course grade than getting 100% of the questions through gathering hints from others without true understanding.**
- **Serious violations:** The following are examples of serious violations of the honor code: (1) Sharing questions or solutions with anyone outside this course, including posting on outside websites. (2) Collaborating with anyone except students currently taking this course. (3) Obtaining solutions using material from past years, other websites, or large language models.

- **Submission Format:** The submitted PDF should be typed and in the same format and pagination as ours. Please include the text of the problems and write **Solution X:** before your solution. Please mark in Gradescope the pages where the solution to each question appears. Points will be deducted if you submit in a different format.
- **Late Day Policy:** To give students some flexibility to manage your schedule, you are allowed a net total of **six** late days through the semester, but you may not take more than **two** late days on any single problem set.
- **Attempting problems and "I don't know" policy.** Some problems might be harder than others, so don't despair if they require more time to think or you can't do them all. Just do your best. Also, you should only attempt the bonus questions if you have the time to do so. If you don't have a proof for a certain statement, be upfront about it. You can always explain clearly what you are able to prove and the point at which you were stuck. Also, for a non bonus question, you can always simply write **"I don't know"** and you will get 15 percent of the credit for this problem. If you are stuck on this problem set, you can use Ed to send a private message to all staff.

By writing my name here I affirm that I am aware of all policies and abided by them while working on this problem set:

Name: [[**TODO:** ***Put your name here***]]

HUID: [[**TODO:** ***Put your HUID here***]]

Collaborators: [[**T** ***llaborators here, if any.***]]

1 Question 1 Name 20

You may use a calculator/spreadsheet for both parts of this question.

1.1 10 (2 per part)

The 1977 Apple II pe a processor speed of 1.023 Mhz or about 10^6 operations per second. In 2023 t percomputer (Frontier) performs about 1.2 exaflops per second; i.e. 1.2×10^{18} . (For comparison's sake, it is estimated that evaluating GPT4 to produce a si res 560 terraflops or 5.6×10^{14}.) For each of the following running times (as fu ength n), compute for each of the two computers how large an input it could handle in a week (i.e., $T = 604,800$ seconds) of computation, if it runs an algorithm with that running time: (a) n operations, (b) n^2 operations, (c) $10^6 n \log_2 n$ operations, (d) 2^n operations and (e) Tower(n) operations, where Tower$(0) = 1$ and Tower$(m) = 2^{\text{Tower}(m-1)}$. Your answers can be approximate, up to a factor of two. (Note that an input length is always a non-negative integer.)

Solution 1.1:

[[**TODO:** ***Answer Question 1.1 here.***]]

1.2 10 (2 per part)

Moore's law posits that the number of operations that computer can perform per time unit doubles about every two years, or equivalently, it grows by an order of magnitude every $2 \cdot \log_2 10 \approx 6.6$ years.[1] So in 2100 we may expect computers to perform about 10^{30} operations per second. Under this assumption, how would you compare the largest input that computers can handle in 2100 per unit time vs. what they could handle in 2023 if they run an algorithm that uses the following number of operations: (a) n operations, (b) n^2 operations, (c) $10^6 n \log_2 n$ operations, (d) 2^n operations and (e) Tower(n) operations. For each case show mathematical justification and express your answer as "The largest n in 2100 (roughly) grows/shrinks by an additive/multiplicative factor of X compared to 2023." for the best number X and choice of "additive" and "multiplicative" you can determine.

Solution 1.2:

[[**TODO:** ***Answer Question 1.2 here.***]]

[1] For those comparing carefully with the previous problem: the Apple II was not the fastest computer of 1977.

2 10

2.1 5

A company named Su [illegible] asses of products. Products of Class A (of which there are at most 32) are d [illegible] oduct code, Class B by a 4-bit code, and Classes C and D each by 2-bit code [illegible] e coding scheme $E : \{A, B, C, D\} \to \{0, 1\}^*$ such that, if you prepend the cl [illegible] e product code, the result is a 6-bit encoding of every product sold by Supe [illegible] ustify that all products have distinct encodings. i.e. for your class encoding fu [illegible] p, $E(Class_p)[Product\ code\ of\ p]$ are distinct)

Solution 2.1:
[[**TODO:** ***Answer Question 2.1 here.***]]

2.2 5

Prove that if E' is a class-name coding scheme satisfying the requirements of the previous problem part, then E' is a prefix-free code. (You may use the fact that for every class there is a product associated with every product code within that class. e.g., there is a Class A product with code 00010, and there is a Class C product with code 10, a Class D product with code 10, etc.)

Solution 2.2:
[[**TODO:** ***Answer Question 2.2 here.***]]

3

15

For $n \geq 1$, let $\mathrm{Odd}_n : \{0,1\}^{2n} \to \{0,1\}$ be the function such that $\mathrm{Odd}_n(x_0, \dots, x_{2n-1}) = 1$ if there exists an odd $i \in [2n]$ such that $x_i = 1$ and $x_j = 0$ for every $j < i$. Otherwise $\mathrm{Odd}_n(x_0, \dots, x_{2n-1}) = 0$. Show that Odd_n ha [illegible] circuits. For partial credit, you can show that Odd_n has circuits of size $O(n$ lo [illegible]

Hint: *You might fi* [illegible] *t the function* $\mathrm{OR}_n(x_0, \dots, x_{n-1})$ *has* $O(n)$ *sized circuits, where* OR_n *is* [illegible] $\dots, x_{n-1}) = 1$ *if there exists* i *such that* $x_i = 1$ *and* $\mathrm{OR}_n(x_0, \dots, x_{n-1}) =$ [illegible] *e a recursive way to compute* Odd_n *in terms of* Odd_{n-1} *and some* $\mathrm{OR}_?$ *functi* [illegible]

Solution 3.0:

[[**TODO:** ***Answer Question 3.0 here.***]]

4

20 + BONUS 10

4.1

20

Define a $\{MAJ, MIN$ [illegible] be one that uses the following gates:

- $MAJ : \{0,1\}^3 -$ [illegible] inputs $a, b, c \in \{0,1\}$ if and only if the majority of a, b, c are equal to 1 ([illegible]
- $MIN : \{0,1\}^3 -$ [illegible] inputs $a, b, c \in \{0,1\}$ if and only if the majority of a, b, c are equal to 0 ([illegible]
- $ZERO$: constant function with zero inputs and output 0

Prove that for every function $f : \{0,1\}^n \to \{0,1\}$, there is a $\{MAJ, MIN, ZERO\}$ circuit that computes f.

Solution 4.1:

[[**TODO:** ***Answer Question 4.1 here.***]]

4.2

BONUS 10

Define a $\{MAJ, AND, OR\}$ circuit to be one that uses the following gates: $MAJ : \{0,1\}^3 \to \{0,1\}$ (defined as above), and $OR : \{0,1\}^2 \to \{0,1\}$, $AND : \{0,1\}^2 \to \{0,1\}$ defined as in the text. Prove that there does not exist a $\{MAJ, AND, OR\}$ circuit that computes the function $MIN : \{0,1\}^3 \to \{0,1\}$ (defined as above).

Solution 4.2:

[[**TODO:** ***Answer Question 4.2 here.***]]