ERSP Research Log

Useful Resources
Proposal Planning
ERSP_Revised_Proposal
https://www.overleaf.com/read/nvskpkqhqhcf
https://github.com/mara-downing/NN_Project

Winter 2022

Week 7: 2/14/22 - 2/20/22

Accomplishments

-

Week 6: 2/7/22 - 2/13/22

Accomplishments

- Overcame covid.

Week 5: 1/31/22 - 2/6/22

Accomplishments

- We spent some hours making our presentation for the class and practicing it. After our presentation, we had individual meetings with Professor Mirza.

Week 4: 1/24/22 - 1/30/22

Accomplishments

- If I’m not mistaken, we’ve finished adding var to var comparisons, and according to our tests, it is working properly.

Tuesday 1/25 (2 hours)

Implementing Var to Var Comparisons:

- Here is a sample of some of the code we have written (that Mara checked on Wednesday):
  - ```cpp
    // check x
    string num2 = lexed[2].substr(3);
    int intnum2 = stoi(num2);
    assert(intnum2 < (int)inputSize);
    num2 = lexed[2].substr(2);
    if(num2[0] == 'x'){
        num2 = "sym_" + num1;
    }

    if(lexed[1] == "C:<"){
        outputconstraints = outputconstraints && c.int_const(num1.c_str()) < c.int_const(num2.c_str());
    }
    else if(lexed[1] == "C:<="){
        outputconstraints = outputconstraints && c.int_const(num1.c_str()) <= c.int_const(num2.c_str());
    }
    else if(lexed[1] == "C:>"){
        outputconstraints = outputconstraints && c.int_const(num1.c_str()) > c.int_const(num2.c_str());
    }
    else if(lexed[1] == "C:>="){
        outputconstraints = outputconstraints && c.int_const(num1.c_str()) >= c.int_const(num2.c_str());
    }
    else if(lexed[1] == "C:=="){
        outputconstraints = outputconstraints && c.int_const(num1.c_str()) == c.int_const(num2.c_str());
    }
    else{
        outputconstraints = outputconstraints && !(c.int_const(num1.c_str()) == c.int_const(num2.c_str()));
    }
    ```
- Simply put, for each comparison, we checked the first variable, x or y, the comparison such as <=, and the second variable, x or y. We then added them in the appropriate format to the z3 objects that Mara had already created previously in the function ParseConstraints.

Week 3: 1/17/22 - 1/23/22

Accomplishments

- Essentially, this week, we’ve come to a better understanding of Mara’s code so far. We still don’t have everything figured out, but we do understand some of the relevant code to our upcoming tasks, specifically adding variable to variable comparisons to the z3 constraints.
    ```

Wednesday 1/19 (2 hours)

Looking at the Code

- We met to look at all of the code that Mara was written so far and that we've pulled from the Github repo. There is a lot to look at and break down, so we tried our best to understand all of the relevant functions that Mara explained in the startup document: ERSP Startup Info. Still, it was very helpful to have met with Mara so that she could go into more detail about the function implementations.

Week 2: 1/10/22 - 1/16/22

Accomplishments

- We ran into an issue with syncing with the Github repo that Mara had set up. Thankfully, Mara found a solution through Docker that allowed us to finish this setup. We also have all downloaded z3 and ABC. Because this took a few days, though, we're off to a bit of a slow start this quarter and are already feeling a little behind. What we should start working on soon is the ability to compare the input variables to each other, as opposed to just comparing constants.

Fall 2021

Week 11: 12/6/21 - 12/12/21

Accomplishments

- We finished our proposal! We were all super stressed this week with finals but were able to meet for a few hours and grind out the finishing changes.

Thursday 12/9 (1.5 hours)

Grad Student Interview

- Grad Student Interview

Wednesday 12/8 (4 hours)

Work on Final Draft of Proposal

- ERSP_Revised_Proposal

Week 10: 11/29/21 - 12/5/21

Goals:

- [ ] Go through the five Marabou examples, turn them into Python expressions and determine if they are satisfiable
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~
- [x] ~~Finish Research Proposal Paper~~
- [x] ~~Prepare final presentation~~

Accomplishments

- We translated our proposal into a presentation. However, it is still lacking good flow, visuals, and relevant details; we will add these in later. In addition, we have made more progress on our proposal. It's all starting to feel like a legitimate paper. We need to meet once more with Mara before submitting a final draft next week.

Saturday 11/4 (2.5 hours)

Work on Final Draft of Proposal

- ERSP_Revised_Proposal

Thursday 11/2 (2.5 hours)

Work on Final Draft of Proposal

- ERSP_Revised_Proposal

Tuesday 11/30 (2.5 hours)

Prepare Final Presentation

- ERSP Presentation

Week 9: 11/22/21 - 11/28/21

Goals:

- [ ] Go through the five Marabou examples, turn them into Python expressions and determine if they are satisfiable
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~
- [x] ~~Look at peer reviews and update proposal~~

Accomplishments

- We've completed a solid draft of our proposal with help from our peers and grad mentor, Mara. We've also made all these updates to our overleaf document and submitted it for more feedback from Professor Mara and Chinmay.

Friday 11/26 (3 hours)

Proposal Work:

- Proposal Planning

Wednesday 11/24 (2.5 hours)

Met with Mara and Reviewed Proposal Feedback:

- Proposal Planning

Tuesday 11/23 (3 hours)

Proposal Feedback Revising:

- Proposal Planning

Week 8: 11/15/21 - 11/21/21

Goals:

- [ ] Go through the five Marabou examples, turn them into Python expressions and determine if they are satisfiable
- [x] ~~Continue work on proposal, convert it to overleaf document~~
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~
- [x] ~~Peer review Team Bultan's proposal~~
- [ ] Look at peer reviews and update proposal (have not receive reviews yet this week)

Accomplishments

- This week, we continued work on our research proposal and transferred our introduction, background, proposition, and cited work bodies into a collaborative overleaf document. I am very happy with the progress we've made so far this quarter.

Sunday 11/21 (2.5 hours)

Professor Bultan's ERSP Group Proposal:

- Peer Review Feedback Form.docx

Wednesday 11/17 (3 hours)

Proposal planning:

- Proposal Planning and https://www.overleaf.com/project/61934cc8a2079776cf3c1921

Tuesday 11/16 (2 hours)

Proposal planning:

- Proposal Planning

Week 7: 11/8/21 - 11/14/21

Goals:

- [x] ~~Read the z3 article sent by Mara (and a z3 program)~~
- [ ] Go through the five Marabou examples, turn them into Python expressions and determine if they are satisfiable
- [x] ~~Continue work on research proposal~~
    - [x] ~~Find more related background papers~~
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~

Accomplishments

- I have a much better understanding of SMT solvers after looking through z3 articles and notation. However, I still have not practiced using them and thus don't feel confident quite yet in utilizing them in our team's research.

Sunday 11/14 (2 hours)

More work on our Research Proposal

- Proposal Planning

Friday 11/12 (2 hours)

Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications:

- https://arxiv.org/pdf/1910.01624.pdf#:~:text=In%20the%20rest%20of%20this,no%20adversarial%20ex-%20amples%20exist.

Tuesday 11/9 (2 hours)

z3

- https://web.archive.org/web/20210119175613/https://rise4fun.com/Z3/tutorial/guide
- Z3 is used to check the satisfiability of logical formulas
- We say that a Z3 stack is satisfiable if there is an interpretation that makes all asserted formulas true
    - If the formulas are satisfiable, then Z3 returns sat. If not, Z3 returns unsat
    - Z3 may also return unknown when it can't determine whether a formula is sat or not
    - When the common check-sat returns sat, the common get-model can be used to retrieve an interpretation that makes all formulas on the Z3 internal stack true
- Lots of commands such as pop, push, display, simplify
- The basic building blocks of SMT formulas are constants and functions
    - Constants are just functions that take no arguments, so really everything is simply a function
- Z3 has builtin support for integer and real constants (not to be confused with machine integers which are 32-bit and 64-bit) and floating point numbers.
- Bitvectors
    - Z3 supports Bitvectors of arbitrary size, which may be defined using binary, decimal, and hexadecimal notation (#b010 in binary is a bitvector of size 3, #x0a0 in hexadecimal is a bitvector of size 12)
- Lots of information on Arrays in Z3
- Datatypes, scalars, recursive datatypes
- Quantifiers
    - Quantified bit-vector formula (QBVF) is a many sorted first-sorted logic formula where the sort of every variable is a bit-vector sort. The QBVF satisfiability problem, is the problem of deciding whether a QBVF is satisfiable modulo the theory of bit-vectors.
- Z3 programs seem somewhat understandable. It would be nice to have Mara walk us through a z3 program that she herself has written.-

---

Week 6: 11/1/21 - 11/7/21

Goals:

- [x] ~~Read network article that Erin sent~~
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~
- [x] ~~Meet with Olivia~~
- [x] ~~Read through slides from Olivia~~
- [ ] Read the z3 article sent by Mara (and a z3 program)
- [ ] Go through the five Marabou examples, turn them into Python expressions and determine if they are satisfiable

Accomplishments

- This week I feel we all gained a much much better understanding of SMT solvers and symbolic execution. Specifically we have a better idea of where the inputs come from and how the outputs are generated. In addition, our research topic is becoming more clear and we have started on our first proposal draft. Our current research problem involves the lack of scalability of verification methods of networks through SMT solvers. We have a couple of ideas with help from Mara on how to approach this. For one, we can make some adjustments to the SMT input (how it is read in). We could also compare a model count to solutions being found in the SMT process to determine when we can stop and thus save time.

Sunday 11/7 (2 hours)

Continued work on Proposal

- Proposal Planning

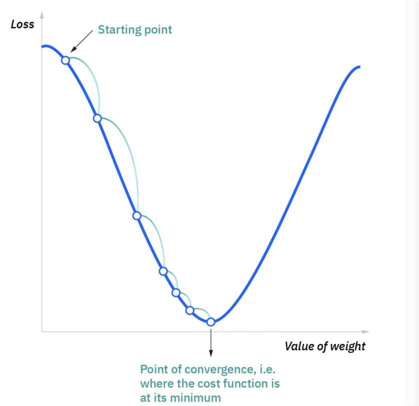
Saturday 11/6 (2 hours)

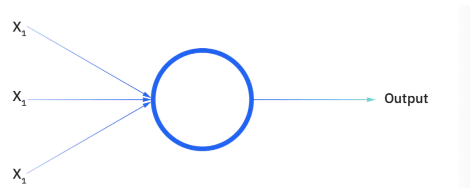
Slides from Olivia

- Stuff that might help
- Z3 guide - https://ericpony.github.io/z3py-tutorial/guide-examples.htm

IBM - Neural Networks

- Each individual node can be thought of as being composed of input data, weights, a bias, and an output
- $$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$
- $$output = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases}$$
  I believe f(x) in this example represents the activation function ReLU
- Weights are assigned after an input layer is run through the activation function. These weights determine how important any given variable is; the larger the weight, the more significant the node is. This process repeats, passing data from one layer to the next.
- $$Cost\ Function = MSE = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2$$
- The goal is to minimize the cost function. The lower the cost, the high the accuracy and better the fit for any input to output. y-hat represents the predicted outcome, y is the actual value, m is the number of samples, and i represents the index of the sample.
- This process, in which the algorithm adjusts its weights, is called gradient descent and allows the model to determine the direction to take in order to reduce errors (minimizing the cost function)

- With each training example, the weights of the model adjust to gradually converge at the minimum
- 

- The simplest network is the single neuron:
    - 

- Feedforward neural networks (multi-layer perceptrons) are comprised of an input layer, hidden layers, and an output layer.
- Convolutional neural networks are similar to feedforward networks and utilize matrix multiplication to identify patterns.
- The "deep" in deep neural networks refers to the depth of layers in the network (generally consisting of more than three layers)

Thursday 11/4 (2 hours)

Meeting Notes

- Some notes from our meeting with Olivia (previous Bultan ERSP group)
    - For a program that has a lot of paths, something like KLEE will determine every possible node (end of branch, possible pathway) and its exact constraints (conditions) that are met or not met to get there. An smt solver such as z3 will take these constraints as input and give a concrete solution (that meets all of these constraints) as output
    - A program with a bunch of conditionals is input into KLEE and it gives you the possible pathways. However, the smt solver is not very scalable (it doesn't work for very large inputs). So, complex programs that lead to very complicated constraints (output from KLEE) will overwhelm the smt solver.
    - KLEE reads in a program, and gives a bunch of output files (that each contain a long long list of the conditions)
    - The smt solver, z3 for instance, would read in these KLEE output files and give a concrete solution, potential ranges of solutions, whether it's satisfiable or not

    - Model counting → input would be a constraint file (such as from KLEE) and the output would be the number of solutions (not any concrete solutions)
- Some notes from our meeting with Mara
    - Model counting can be used in place of smt solvers
    - Model counting returns the number of possible solutions to a symbolic execution tree. If we know that a tree has 15 outcomes, and the first 4 branches sum to 15, then we can infer that the rest of the tree is unsatisfiable, and they can be skipped.

Week 5: 10/25/21 - 10/31/21

Goals:

- [x] ~~Literature Search Part 1: The Lay of the Land~~
- [x] ~~Literature Search Part 2: Narrowing Down~~
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~

Sunday 10/31 (3.5 hours)

Literature Search Part 1: The Lay of the Land

- Intriguing Properties of Neural Networks - Notes
- Literature Search

Wednesday 10/27 (1 hours)

Identifying the Components of a Research Paper

- Our group read through the VMotion research paper and identified different structural components: thesis, big picture, specific knowledge gap, and specific contributions. We also practiced sifting through related works/cited papers that are strongly related.

Tuesday 10/26 (2.5 hours)

Literature Search Part 1: The Lay of the Land

- Literature Search

Week 4: 10/18/21 - 10/24/21

Goals:

- [x] ~~Independent Learning and Teaching Assignment~~
- [x] ~~Attend meetings with Professor Bultan, Mara, and the group~~

- [x] ~~Identifying Related Research Problems~~

Sunday 10/24 (2 hours)

Identifying Related Research Problems:

- Notes from Mara:
  - Will - “model counter for string constraints called ABC” as well as “access control policies for cloud services.” These control policies aim to boost security on shared cloud servers.
  - Seemanta - working to approximate the difficulty of reaching a statement in code through random testing. The aim is to estimate “the percentage of total inputs that could hit a statement.”
  - Burak - “working on mitigating side channels for IoT devices,” using machine learning to “find any fix side channel leaks in agricultural IoT devices.”
  - Laboni - using the symbolic execution tool ANGR
  - Yilmaz - looking at how to quickly transfer a running program between devices if one of them is compromised
  - Chaofan - working on a fuzzer (currently have no idea what this is)
- Problems our group discussed with Mara:
  - Currently, the SMT formulas in the symbolic execution tree will sometimes overwhelm the solver. How could this be solved?
  - Is there an easy way to deal with user input? Mara says that while we know exactly what format the symbolic execution tree constraints will be in, we need to define a format/rules for the user constraints. Is there an easy way to read them into the program? Also, are there any ways to “integrate them into the constraints” in the tree to speed up execution?
  - I feel that seeing a real neural network in action would help me, and my group, better identify some issues that they are experiencing. We have learned about how computationally expensive solvers are as well as about the tradeoff of precision for efficiency. However, we have absolutely no idea how to approach a problem such as this. We would need to talk much more with Mara.
- We will discuss possible solutions to these problems with Mara this Thursday

Monday 10/18, Tuesday 10/19 (3.5 hours)

Learning and Teaching Assignment: Linear Programming:

- Learning and Teaching: Linear Programming
- Notes on Anoushka and Erin’s presentations:
  - Notes were taken on in-class handouts

Week 3: 10/11/21 - 10/17/21

Goals:

- ☑ ~~Monday 10/11 class discussion~~
- ☑ ~~Attend meetings with Prof Bultan, Mara, and the group~~
- ☑ ~~Read the abstracts of the other important papers sent by Mara~~

Friday, 10/15 (1 hour)

Read the Abstracts of the Other Important Paper sent by Mara:

- The Marabou Framework for Verification and Analysis of Deep Neural Networks
  https://link.springer.com/content/pdf/10.1007%2F978-3-030-25540-4_26.pdf
  - Marabou is an SMT-based tool that can "answer queries about a network's properties" and convert these queries into constraint satisfaction problems. Marabou also makes use of "parallel execution" for improved scalability.
- Scalable Quantitative Verification For Deep Neural Networks
  https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9402111&casa_token=M90orY8TKvAAAAAA:x0Ces27OPNdnRHYk5pP3qGMzvwIOkCdGzT6O6xHuwhz3KA6oMr8PbS7GVwWVIWxXN-p1TOIUkA&tag=1
  - This paper proposes a "scalable quantitative verification framework" for neural networks. It works with "provable guarantees," specifically with large amounts of tests until a probabilistic property can be certain.
- Framework for Binarized Neural Networks:
  https://arxiv.org/pdf/2103.07224.pdf
  - This paper studies how to transform BNNs into binary decision diagrams which can then be used for precise analysis.
  - Unfamiliar terms: BDD4BNN
- Exploiting Verified Neural Networks via Floating Point Numerical Error
  https://arxiv.org/pdf/2003.03021.pdf
  - Many verifiers inaccurately model floating point arithmetic, leading to imprecise results, but do not address this issue. This paper presents a method to efficiently check the incorrectness of robust claims made by a verifier.

Monday, 10/11 (1 hour)

Monday 10/11 class discussion:

- Quantized is fast but less precise, idealized is more precise but slower
- The author wants to make the quantized version more scalable (usable for larger applications)
- The solutions: 1) dead branch removal, minimal bit allocation for each bit-vector variable, elimination of redundant branches
- The author proves his work with tables comparing standard method results with his methods
- Major things that we were all confused about:
  - ReLU, QF_BV2
- Questions for Mara:

- What is an attack radius?
- Ask Mara how to better understand the tables

Week 2: 10/4/21 - 10/10/21

Goals:

- [x] ~~Read "How to Read a Research Paper," by William Griswold~~
- [x] ~~Schedule Group Meeting Time~~
- [x] ~~Read at least the main paper shared by Mara, and an extra if I feel confident~~
- [x] ~~Read Scalable Verification of Quantized Neural Networks and Record Notes~~
- [x] ~~Meet with Mara, then team~~
- [x] ~~Watch first 2 3Blue1Brown neural networks videos~~
- [x] ~~Look into all areas of confusion from Part 1 of the reading~~
- [x] ~~Read through Part 2 of the reading a couple of times to understand it~~

Sunday, 10/10 (5.5 hours)

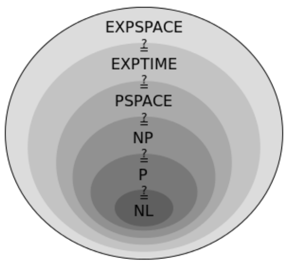
3Blue1Brown Neural Network Videos Notes:

- https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi&index=1
- It would be incredibly hard to translate the graphics in the video to written word, but pretty much, a neural network is a system that takes in lots and lots of input, to give one or a few outputs
  - From this perspective, a network is essentially just a function, takes inputs, gives outputs; the videos provide a great way to visualize neural networks
- Much of neural networks rely heavily on linear algebra concepts

Part 2: Scalable Verification of Quantized Neural Networks: Thomas A. Henzinger, Mathias Lechner, Đorde Žikelić

https://arxiv.org/pdf/2012.08185.pdf

- 2.1: Researching my points of confusion
  - SMT-based verifications - https://en.wikipedia.org/wiki/Satisfiability_modulo_theories
    - Satisfiability modulo theories are constraint satisfaction problems and deal with determining whether a formula is satisfiable. For instance, it may check linear inequalities such as 2x + 4y <= -z
  - Linear Programming (LP) - https://en.wikipedia.org/wiki/Linear_programming
    - Linear programming, also known as linear optimization, is a method to achieve the most ideal outcome through linear relationships.
  - Boolean Satisfiability (SAT) - https://en.wikipedia.org/wiki/Boolean_satisfiability_problem

        - The problem of determining whether there exists an answer to a boolean formula. It checks whether variables in a function can be swapped for TRUE in a way that allows the entire formula to evaluate to true. For example, (a AND NOT a) = TRUE is not satisfiable because a cannot be both TRUE and FALSE
    - NP-complete - https://en.wikipedia.org/wiki/NP-completeness
        - Nondeterministic polynomial-time complete involves brute force algorithms. It is a problem that can be "verified quickly" and an actual solution can be found through a brute force approach checking all possible solutions.
    - NP-hard - https://en.wikipedia.org/wiki/NP-hardness
        - Nondeterministic polynomial-time hardness means "at least as hard as the class of problems in NP."
    - PSPACE-hard - https://en.wikipedia.org/wiki/PSPACE
        - 

PSPACE, according to Mara, is like polynomial time but really really complex and can have a really really long runtime
    - NEXP - Another really complex space
    - TQBF - https://en.wikipedia.org/wiki/True_quantified_Boolean_formula
        - True quantified boolean formulas are formulas in which each variable is bound by existential or universal quantifiers.
- 2.2: Griswold's questions
    - Part 2: Scalable Verification of Quantized Neural Networks Griswold Questions

Tuesday, 10/5 (4 hours)

Part 1: Scalable Verification of Quantized Neural Networks: Thomas A. Henzinger, Mathias Lechner, Đorđe Žikelić

https://arxiv.org/pdf/2012.08185.pdf

- Part 1: Scalable Verification of Quantized Neural Networks Notes

Saturday, 10/2 (1.5 hours)

Research Paper Thoughts:

- Scalable Verification of Quantized Neural Networks:
    - I am having an incredibly difficult time trying to understand this.
    - From my understanding, something is happening around performing calculations with integers instead of floats because floats take up 32 bits whereas ints take up only 8. While I'm not sure if this is 100%, doing this improves space usage as well as time efficiency at the expense of accuracy.

    - I have no idea what the background and related work functions mean
    - There are dozens of terms that I do not recognize such as NP-complete, PSPACE-hard, SMT, QBF formula, ReLu-N operations, etc.
    - I don't have enough background knowledge to comprehend the tables and function references throughout the entire paper.
- The Marabou Framework for Verification and Analysis of Deep Neural Networks
    - This paper uses more simplified language than the first but still, I am not reaching an understanding of what a deep neural network really is.

Thursday, 9/30 (0.5 hours)

How to Read a Research Paper Thoughts:

- Griswold emphasizes asking yourself a lot of questions while reading. Asking questions can help you identify areas of confusion or help you find the focus of the paper.
- I remember learning in WRIT 2 that it can be helpful to carefully read the abstract then skim the paper to gain an understanding of the paper's subject before scrutinizing the paper's body. Thus, you'll be able to read the paper more thoughtfully because you know what to focus on.

Week 1: 9/27/21 - 10/3/21

Goals:

- [x] ~~Attend research group meeting and record attendance~~
- [x] ~~Set up my research log~~
- [x] ~~Complete reflection 1: identity and ERSP preliminary thoughts~~
- [x] ~~ERSP contract~~
- [x] ~~Reflect on research logs~~

Wednesday, 9/29 (class time, 1 hour)

ERSP Alumni Quick Advice:

- Work ahead of schedule, it can be very easy to fall behind. Don't let your teammates, professor, or yourself down.
- Ask a lot of questions; it's okay to not feel confident in yourself. The professors and other researchers are super experienced and thoughtful. They will all be willing to help you and love to know that you are super into your project.
- Tyler suggests breaking the project down into sub-sections so that each ERSP member can take charge of their own portion. This allows each student to take on a sort of leadership role in the group.
- Olivia says that she and her group members frequently felt lost during their meetings with their professor and PhD students. She said not to worry about this.

- Ari says that if you put the time into the program, you will see great results in your ability to comprehend research papers as well as your project.
- Set small and realistic goals for each day. Something as little as, “I want to understand this single line of code or what this paragraph means.”

Tuesday, 9/28 (3.5 hours)

Reflection 1: Identity and ERSP Preliminary Thoughts:

- Reflect on your identity as a researcher or someone who could become a researcher.
    - I think that despite my lack of experience as a researcher, I have what it takes to be good at it. For one, I feel I have the curious mindset of one. In addition, I find that when something puzzles me, or I can't find the right solution to a problem, I can't put the pencil down. I feel that the challenge of research especially would give me more motivation to learn and work.
- Reflect on your identity as a computer scientist, data scientist, computer engineer or electrical engineer (select the identity that applies most closely to your studies and/or research).
    - I have no experience with research and, despite an entire year in computer science studies, I hardly feel I am knowledgeable whatsoever in this discipline. Thus, I certainly do not feel like a computer scientist, but I hope I will once ERSP is through with me!
- From your current perspective, what does it mean to do research in computer science, data science, computer engineering or electrical engineering? (Again, select the term that most closely applies to your studies and work).
    - Currently, my idea of research in computer science involves searching for methods that improve efficiency in software. I'm sure despite how broad this statement is, it still fails to encapsulate all that is actually done in computer science research. I also imagine that much of research in computer science today includes studying AI and a computer's ability to self-learn.
- Initial thoughts about ERSP: What are you most excited about in ERSP, and why?
    - As a second-year computer science student, who spent the entirety of his first year online during the pandemic, I have had minimal exposure to this vast field that I am studying. For all of this past year, I have been hoping to be a part of and contribute to more meaningful projects. ERSP is the portal through which my current goals will become reality. I will be able to dive deep into a specific topic in computer science and actually put what I've learned in the classroom to work. I find it super exciting to know that I will be contributing to something that could one day improve many peoples' lives.
- Initial thoughts about ERSP: What are you most nervous about in ERSP, and why?
    - Again, as a second-year with practically no project experience, I am a little nervous about my ability to contribute to the project. After our first lecture today, however, I feel a little reassured that I likely share this feeling with many others. And, after all, this feeling of being a helpless beginner is why I am participating in ERSP! I'm grateful for this opportunity to improve as a computer scientist.

Log Reflection:

- How did the logs differ in style (not just in content)? What advantages do you see in one style over another?
  - First, I will briefly describe the style of each of three logs: Miranda's, Tony's, and Adrian's. Miranda's log, which is contained on a google site-specific to her project, greatly utilizes bullet points to provide detailed descriptions of her accomplishments every day. In addition, Miranda uses colored headers and horizontal lines to clearly differentiate between each week while also providing a list of the log's contents with corresponding links to each section. Tony's log, which is a GitHub page, makes similar use of horizontal lines and links. However, his page lacks the detail that Miranda includes, and I feel he provides little insight into his project. Finally, Adrian's log appears to be inspired by Miranda's, especially considering how he states that he appreciates Miranda's work. Similar to her, he utilizes colored headers and provides detailed accounts of his accomplishments each week.

    Like Adrian, I hope my log will emulate Miranda's. Her use of bullet points, colored headers, horizontal lines, and links made her log incredibly organized and easy to navigate. It's simple yet very detailed.
- How do you think the logs were useful, both to the researcher as well as those working with the researcher?
  - I think there are two great uses for the logs. For one, even after a frustrating day, one can write in their log all that they did and feel at least some satisfaction in knowing that they accomplished something. Secondly, the logs are great records of all that has been done. They serve to remind not only of the progress that has been made but also of the researcher's main goals.

    Miranda's log structure displays how the log can be helpful in breaking down the main goals into smaller tasks to tackle each week and even each day. The logs seem useful in that they help provide clear direction on what to do each session.
- Did the students keeping their logs seem to meet their goals? Did they get better at meeting their goals over time?
  - Yes, all the students appeared to be consistently meeting their goals. Over time, I'm not sure that they got better at meeting their goals; however, I do feel that their goals often became more significant and possibly demanding. Overall, the goals seem like a great method of maintaining a good pace in the project.