**THIS WAS A WRITE UP FOR A JOB APP. ITS KIND OF OFF TOPIC, BUT MAY HELP WITH LETTER**

**Career Goals: Describe your career goals and how this program would help you fulfill them**

The Two Career Goals that I hope to achieve over the next decade and a half is to “Perform Original Research”, both in Academia and in Industry.

In the context of Academia, “Performing Original Research” amounts to writing and defending a dissertation to a PhD committee. Assuming I do go on to a PhD program in math, I successfully pass the qualifying examinations, and I start writing a dissertation, the focus around the dissertation would consist of writing mathematical proofs. This is noteworthy, because as opposed to many other fields where the focus of the dissertation is around data acquired from “real world” research and the conclusions that can be made from it, the content of a math dissertation requires forming conclusions, but also supplying proofs to verify their validity. Proof writing is an essential skill for mathematicians of all levels since along with a thorough understanding of the relevant topic, creativity and stubbornness are essential to discovering the logic that verifies a proof. This is a skill that requires significant practice, and whether it be writing proofs, implementing software, or using machine learning algorithms to make conclusions from data sets, the more time I spend “struggling” with difficult topics, the better I become at discovering the logic behind proofs. So, whether my potential future work I’ll do through NRIEP will be closely related to combinatorics (my favorite field in math), or in implementing machine learning / artificial intelligence algorithms, if it pushes to me struggle with, and understand new difficult material, the practice I get would be directly beneficial in helping me perform Research in Academia and eventually defend my dissertation.

As opposed to research in Academia, Performing Original Research in Industry would be more along the lines of designing and implementing something that is on the cutting edge of current technology. This isn’t to say that academic research has not directly resulted in the creation of cutting-edge technology, just like this isn’t to say that research in industry has not directly resulted in original ideas and understanding of complicated topics. The reason why I differentiate between the two is that the sole goal of research in industry is to physically create something that is tangible, and can solve some purpose in the real world, while the sole goal of research in academia is to add to the overall human knowledge. But the skills required to discover proofs are very similar to the skills required to implement cutting edge technology, the consequence of my goal to improve my abilities to write proofs is the improvement of my abilities to innovate and create something that hasn’t been created before. So, I still believe that it doesn’t matter what I do, I want to do it if it's difficult and it pushes me to understand the material at a higher level.

**Academic Interests: Describe any special interests you have that would make you a valuable participant**

I have two main fields of academic interest that I have pursued in the past and want to continue to pursue in the future, mathematics and computer science.

Even though splitting my focus between both subjects means I must be picky about what topics I want to devote my spare time to, I rarely worry because it always seems that my understanding and skills built from working in each field is directly beneficial to my understanding and skills in the other.

As a computer science, having experience with coding makes me predisposed to working with algorithmics and understanding logic, which assists me in quickly wrapping my head around difficult to understand topics. The benefits these skills give me as a mathematician are undeniable, the same skills required to wrap my head around how a piece of code work are the same skills required to understand a mathematical process or proof. But the largest advantage to knowing software engineering as a mathematician is being able to throw together scripts to perform calculations and to analyze data, which makes creating conclusions and seeing patterns in mathematics trivial, compared to making the calculations by hand.

As a mathematician, having experience with understanding and verifying the theory being various results helps me integrate results themselves, which applies to the field of computer science when implementing algorithms. It can be difficult to implement them but understanding why they work makes the process of implementation easier.

**Research Experience: Describe any research experience you may have in relation to this program, including any STEM afterschool programs, camps, or competitions in which you have participated․**

Recently, I have been doing undergraduate research in the field of group theory. As opposed to most fields in pure mathematics, Group Theory is directly intertwined with computer science. That is, we use software to help gain insight abouts aspects of group theory, which perfectly aligns with my interests. For my specific research project, I have been working on a necessary and sufficient condition for a well-known equality pertaining to permutations and reframing that condition in terms of hypercubes. I’ve been able to read research papers to gain a understanding of the relevant topic, and then implement functions to exhaustive search over permutations and calculate whether the equality holds or not.

**Technical Skills: Describe any technical skills you may have which are related to the science and engineering disciplines**

C and Python are the programming languages im the most comfortable with. Python is currently my favorite programming language because since it abstracts so much away from the user, it makes it very straightforward to implement advanced mathematics and complicated algorithms, since the user can think more about the mathematics rather than the task of implementing the software. On the other hand, I’ve had lots of experience with C, whose main feature is the lack of abstraction, meaning the user is assumed to be mature enough to have control of very low-level aspects of software development.

LaTeX is a mathematical typesetting software that I have gotten quite familiar with over the past few years. Having the ability to type up, discuss, and present information about math is obviously very important for mathematicians, and the LaTeX software is the unequivocal “industry standard” which allows mathematicians go about communicating mathematics. LaTeX trivializes typesetting mathematics. It also is highly customizable, with packages like TikZ which help rendering more complicated objects, like plots / charts / graphs etc.

Although Computer Graphics isn’t my specialty, I’ve had plenty of opportunities to become familiar with it. Building a project in unity has allowed me to be familiar with how one goes about allowing 3d objects to move and to create effects on a screen in response to user input. I’ve also had experience with various python tools (Django, tkinter, pygame) which are used to implement user interfaces. In my studies, I’ve also become familiar with the rendering pipeline and the background it takes to go about taking code and using it to render pixels on a screen.

As a computer scientist, I am an avid user of GitHub. I’ve had lots of experience with creating GitHub repositories and formatting Read Me’s to concisely dispense information. As a computer scientist, I was taught that documentation of code was essential, but I really learned its importance when I was tasked to modify and extend code not written by computer scientists. Attempting to decode inherently broken code that lacks any repository lever documentation is miserable, so I’ve had lots of practice in building code in a way that makes it easy for others to understand it.

Research

I created Python implementation of standard permutation algorithms (converting between 1-line, disjoint cycles, transpositions) and implementations of metrics on the symmetric group of permutations. There is an inequality defined in terms of those permutation statistics / metrics, and recently a necessary and sufficient condition was found for when equality holds. We extended it to signed permutations / hypercubes and i am currently analyzing and making a conjecture about a necessary and sufficient condition for equality of the equation of metrics over signed permutations.]

SOP

Five years ago, my younger self would have been in disbelief if he knew that I wanted to be a mathematician. That’s not to say I was inherently bad at mathematics, it’s just that my younger self lacked self-confidence in my math abilities. Music was my career choice at the time, but the choice was short-lived because I knew music was not a sustainable career, and my worsening asthma symptoms made it impossible for me to play my instrument. For the sake of choosing a reasonable career, I then chose to major in computer science. Now that wasn’t a choice I regret since implementing algorithms is something I’d love to spend a lifetime doing. But I chose computer science out of necessity. The path that led me to choose mathematics was drastically different from the others. After dipping my toes into university mathematics, I discovered an applied math major that worked perfectly with computer science major. It immediately piqued my interest, and I chose to commit to it, even though I didn’t know of its existence before then. Once that happened, switching to general mathematics was simply inevitable.

As I studied and improved my mathematical maturity, proof focused classes quickly became my favorite as the process of proof writing went from some terrifying ordeal of working with abstract nonsense, to being ubiquitous as working through the reasoning to solve simple puzzles. Proofs are my favorite way of working with complex mathematical topics and integrating understanding of the surrounding material. It was when I took a proof-based Graph Theory class where my career goal changed from “being a Software Developer” to “maximizing the amount of math I can do”.

My path to becoming a math tutor was analogous to the way I chose mathematics, my belief that I couldn’t succeed gave me a late start. I was recommended to be a tutor after taking a difficult physics class, but I didn’t accept the position because I believe I wouldn’t be any good at tutoring. Even when applying to my current tutoring position, I started and then quit the application process, only applying when I learned that more tutors were still needed. But once I started tutoring, it became evident that my lack of confidence was unfounded. Although there was the expected learning curve since I had never taught mathematics, the experience has been invaluable since it’s been the perfect opportunity to help others and myself in improving our mathematics ability.

My journey to undergraduate research was also analogous to my journey to choosing mathematics, even though I started as soon as I knew of its existence, I wish I had started earlier. The research project me and my mentor have been working on started with an inequality defined in terms of certain statistics of permutations, and a theorem for when equality holds. I started by implementing those various statistics, along with a permutation class which supports basic permutation methods. Then, we modified the definitions in terms of signed permutations representing hypercubes, and then modified the implementation to get data for the new inequality. Our current task is to find and prove when equality holds over the inequality for signed permutations. The project has been important to me because I’ve been able to implement mathematics in python and am currently attempting to discover and prove a mathematical result using the data I generated. This perfectly combines my goals of working with pure mathematics, and how it applies to software and computers. It’s these aspects that I want to continue to have in my future research projects. I want to create results about mathematics that is directly applicable to computers. To go about this, I think Combinatorics is the perfect research area for me. I enjoy bijective combinatorics the most because the task of creating a bijective function from one set to another has been some of the most creative and ingenious constructions I’ve ever seen or used in math myself. Another aspect of Combinatorics I enjoy, and I think lots of research can be performed, is connecting the field of combinatorics with other fields of mathematics that are more related to pure or continuous mathematics. When I was digging into graph theory, I got so excited when linear algebra popped up via degree and adjacency matrices. And just this week I read that Fourier analysis has direct applications to combinatorics, and im curious to figure out how exactly this is, and whether my hunch is correct that they are used like Taylor Series in generating functions.

Because I had a later start to mathematics than I would have preferred, I have so much mathematics to do before I can be confident enough to call myself a mathematician, rather than just an aspiring mathematician. But my current experiences in teaching and performing research have shown me that I have not nearly done enough mathematics to satisfy my mathematical curiosity. Whether my focused area of research ends up being combinatorics, or an area of mathematics that has nothing to do with computers, my goal will still be to maximize my mathematical maturity, to expand the knowledge of mathematics in myself and others, and to increase the amount of original research that it is out there relating to the topics in math that I enjoy.