Apoorva Bhagwat

aabhagwa@andrew.cmu.edu
+1 513 641 7503

Work Experience

Google (Software Engineering Intern) *Summer 2017*
Developed a performance analysis tool for the Google Cloud Storage (Performance and Reliability) team.

Google (Software Engineering Intern) *Summer 2016*
Devloped a tool for automated analysis and visualization of integration test results for the Borg team to interpret integration test results. *Written with C++ and Python.*

LibertyX (Software Engineering Intern) *Summer 2015*
- LibertyX is a cash on-ramp for the US bitcoin economy. As my primary project, I developed an asynchronous client-server framework based on WebSockets for smooth integration with LibertyX's partner bitcoin services. *Written with Node.js and Python*

Carnegie Mellon (Teaching Assistant) *2015-2017*
- Theoretical Computer Science (15-251), 5 semesters, Head TA
- Imperative Programming (15-122), 1 semester
  (Grading, teaching recitation and holding office hours)

Education

2014 – Present **Bachelor of Science**
Computer Science
with minors in Math and Chinese
Carnegie Mellon University
GPA : 4.0/4.0

Skills

Programming	C, C++, Python, Java, OCaml, x86 assembly, Numpy, SageMath, LaTeX, Git JavaScript, HTML/CSS3
Math	linear algebra, probability, abstract algebra, analysis, topology, computational complexity
Languages	English, Hindi, Marathi, Sanskrit (basic), Mandarin (basic)

Selected Coursework

- Compiler Design
- Computer Systems
- Machine Learning
- Artificial Intelligence
- Abstract Algebra
- Automata, Algebra and Logic (a.k.a. CDM)
- Complexity Theory
- Real Analysis
- Parallel Data Structures and Algorithms
- Theory of Programming Languages

Projects (for fun and for work)

- Parametric polymorphism for C0 *Fall 2016*
  After writing a compiler for the C0 programming language (which is a well-defined, type-safe subset of C), we extended the language with parametric polymorphism (i.e. generic types), and implemented a compiler for this extension. The result was a low-level C-like language that supports generic programming. *Compiler written in OCaml*

- Graph Clustering (Research) *Spring 2017*
  Under the guidance of Prof Avrim Blum, I investigated a novel notion of graph clustering called (α, β)-clustering that is useful in community detection. We proved that a natural problem related to this formulation is NP-hard.

- Surroundify *Fall 2015*
  At HackCMU 2015, we developed a web app that lets users utilize multiple machines to play music synchronously and create a surround-sound effect. The system supports surround-sound effects specified by the user. *Backend written in Python, using Flask.*

- Tool for automated grade entry *2016-2017*
  TODO This is an ongoing project to automatically digitize grades from homework submissions, using tools from computer vision (expected to be useful for TAs of large classes). *Written in Python using OpenCV.*

- Universal Register Machines *Fall 2015*
  As a final project for a theory of computation class, I implemented an optimizing register machine interpreter (that speeds up execution of nested register machine loops from exponential to constant time), and a universal register machine (i.e. a register machine program that runs other register machine programs). *Compiler written in Python.*

- Programming assignment design *Spring 2016*
  Designed a programming assignment for a CS theory class at CMU and managed its autograding infrastructure. This assignment involved proving the Turing-completeness of register machines by incrementally building primitives to simulate Turing machines.

- Linguistics olympiad problem design *2014-2016*
  I have been contributing to India's national linguistics olympiad as a problem designer and tester. Here is one of our past papers.

Awards

2015	Among the top 480 in the William Lowell Putnam mathematical competition
2014	Represented India at the 12th International Linguistics Olympiad (Beijing, China)
2015	Invited to FooBar (Google's programming challenge)