Last Updated on 26th July 2022

Zeki Gurbuz (Gürbüz)

github.com/zekigurbuz | zekigurbuz@utexas.edu / zekidgurbuz@gmail.com | zekigurbuz.github.io

EDUCATION

UNIVERSITY OF TEXAS AT AUSTIN

BS, Mathematics
BS, Computer Science Honors, Turing Scholars Program
Projected: May 2024 | Austin, TX
GPA: 3.8390

SELECTED COURSEWORK

Data Structures Honors
Computer Architecture, Systems Honors
Operating Systems Honors
Discrete Mathematics Honors
Quantum Computing I, II
Competitive Programming
Probability, Statistics
Linear Algebra Honors
Algebraic Structures
Number Theory
Stochastic Processes
Real Analysis
Multivariable Calculus
Introduction to Research

ACTIVITIES & AWARDS

PROGRAMMING, MATHEMATICS

UT ICPC Regionals Team
USACO Platinum - Top ~5% US
AIME Qualifier 2021 - 120/150 AMC 12
2200+ Rated LeetCode - Top <1%
UIL CS Individual 2021 - 1st Place TX
UIL CS Team 2021 - 1st Place TX
UT Dallas BoTB 2021 - 1st Place TX
HP CodeWars 2020, 2021 - 1st Place US

CAPTURE THE FLAG

Atlassian CTF 2022 - 3rd Place
National Cyber Scholar 2021
rgbCTF 2020 - 22nd Place
csictf 2020 - 22nd Place
UIUCTF 2020 - 28th Place

ORGANIZATIONS

UTPC | CORE PROBLEM AUTHOR

Created numerous algorithmic and mathematics-based coding puzzles
Helped to organize biweekly ICPC-style competitions frequently garnering >100 contestants across the US
Provided in-depth solution presentations

DENT CYBERSECURITY | MEMBER

Competed in numerous US-wide "capture the flag" style competitions
Solved various cryptography and reverse engineering challenges
55th out of 17,875 US teams in 2020

EXPERIENCE

FOREFLIGHT | Software Engineering Intern

iOS Mobile Application Development in Swift
May 2022 - August 2022 | Houston, TX

- Coded in Swift using the SwiftUI framework and PencilKit to implement an entirely new flight plan signing feature for FDPx+, allowing pilots to sign off on an OFP before takeoff
- Implemented key features of FDPDesignKit for the FDPx+ team, consolidating UI components into an easily modifiable and interactive library for developers
- Drastically reduced the amount of time needed to draft UI features by maintaining a very concise dependency graph (allowing the efficient use of Xcode previews), and increasing guarantees of conformance to spec by breaking down UI to the "atomic level"
- Learned about the MVVM pattern, dependency injection, and Objective-C networking

A-PLUS COMPUTER SCIENCE | Software Engineering Intern

Online IDE Development in PHP/JS and Test Problem Creation
May 2021 - August 2021 | Houston, TX

- Worked on full stack development of a website used by teachers/students across Texas
- Developed a versatile online IDE using the LAMP stack to handle Java/Python coding
- Employed AJAX techniques to save and retrieve files efficiently through PHP/SQL
- Helped to catch and fix security vulnerabilities such as PyJails in the online IDE itself
- Fixed numerous structural flaws in the codebase and thoroughly cleaned up JS code
- Created algorithmic contest problems for high school students that covered topics such as OOP, data structures, and more advanced algorithms

SELF-EMPLOYED | Competitive Programming Tutor

Algorithms and Data Structures in C++, Java, and Python
July 2021 - Present | Remote

- Taught students one-on-one in an interactive lecture style with homework assignments
- Covered topics ranging from binary search to tree algorithms, dynamic programming

PROJECTS

3-D RENDERER

- Compiled Rust to WASM in order to support a browser-based 3-D rendering engine
- Utilized linear algebra to compute projections and support various shaders

CO-ROUTINES

- Designed and implemented co-routines in C as a pseudo-native language feature
- Utilized x86 Assembly, C, and in-line x86 Assembly to achieve desired behavior

ANONYBOT

- Created a Discord bot using discord.py, allowing users to send anonymous messages
- Hosted an instance on a Raspberry Pi which has processed thousands of messages

AUTOMATED TRADING BOT

- Created a Python bot for Jane Street's etc trading competition during the FTTP event
- Employed strategies such as pennying and arbitrage on a variety of securities to make money with low risk exposure in comparison to standard directional investing
- Interacted with numerous other bots as well as non-playing directional traders

GAME BOY ADVANCE EMULATOR

- Used C++ and ARM7TDMI Assembly to create an emulator and vast testing suite
- Worked on the implementation of bitmap graphics and instruction emulation

MALLOC AND FREE

- Redesigned and implemented C's "malloc" and "free" in C without additional memory
- Used the first-fit algorithm and a doubly linked list for constant-time node merges/splits

BB84 QUANTUM KEY DISTRIBUTION

- Used Python to implement the BB84 algorithm while mimicking arbitrary gate noisiness
- Utilized strategic quantum measurements to provide robust detection of adversaries

PIPELINED PROCESSOR

- Used the HDL Verilog to create an efficient pipelined AArch64 processor
- Supported a reduced instruction set with large speedups over a standard design