# BOWEN ZHANG

Department of CSE, HKUST
bowen.zhang@connect.ust.hk

## EDUCATION

**Hong Kong University of Science and Technology** *August 2021 - Present*
Ph.D. Student in Computer Science and Engineering
Supervised by Prof. Charles Zhang
Research Interests: Program Analysis and Compilers

**Wuhan University** *2017-2021*
B.Eng with honors, Computer Science and Engineering GPA: 3.81/4
Hongyi Honor College. Rank:3/31.
Graduate Thesis: Coroutine Deadlock Detection in Go Language. Instructed by Prof. Xiaoyuan Xie.

**University of California at Berkeley** *June - August 2019*
Exchange Student
Structure and Interpretation of Computer Programs(A+), Game Theory in Social Sciences(B+)

## WORK EXPERIENCE

**Tencent** Shenzhen, China
*Software Testing Engineer Intern at FiT, CDG* *December 2020 - April 2021*

- Testing: perform interface testing(whitebox), END-To-END testing(black box) and smoothness testing (front-end) on several modules of the cell phone recharge business in Wechat Pay
- Python code specification: Fix more than 1 thousand code specification problems in the Python testing code
- Code generation tool: given a protobuf file, it can generate testing code automatically.

**Alibaba Cloud** Hangzhou, China
*Compiler Engineer Intern at C++ Compiler Team* *June - September 2020*

- SPEC Analysis on LLVM vs GCC: This work was aimed to address the performance gap between LLVM and GCC by analyzing the SPEC program deepsjeng. Collected runtime data, compared the assembly code generated by GCC and LLVM, and designed verification experiments. Summed up a report that included 20 opportunities for LLVM in compilation optimizations

## TEACHING

Computer Systems Fundamentals(2020 Fall) - WHU

COMP3021 - Java Programming(2022 Spring) - HKUST

## SKILLS

LLVM: Familiar with LLVM IR; can apply static analysis algorithms on top of LLVM;

Soot: Able to write analyses and perform instrumentation(e.g. coverage tools) on Soot IR.

Programming Languages: C/C++, Go, Scheme, Python, Java, Haskell, SQL, JavaScript, LaTeX

Grammar Tools: Have experiences of using ANTLR, YACC/Lex, and JavaCC.

Performance: Have experiences in locating performance bottlenecks caused by compiler optimization.