Oliver Iliffe

Email | +44 7944 727309 | GitHub | Stack Overflow | LinkedIn | Website

Education

Imperial College London – Advanced Computing MSc **(Sep 2024 - Sep 2025)**
Focus: Systems & Performance Engineering

King's College London – Computer Science BSc **(Sep 2021 - Jun 2024)**
Grade: First-Class Honours

Experience

Algorithms Research Intern (Haskell & Scala) **(Jun 2024 – Sep 2024)**

Supervised undergraduate research position at King's College London

- A team of three, lead by a professor with research interests in theorem provers and compilers
- Research novel matching algorithms for regular expressions
- Build a testing framework for verifying the correctness of extended regular expression matchers
- Leverage **Haskell** and **Scala** to produce elegant algorithms that could later be formalised in **Isabelle**

Teaching Assistant (Java, Scala & C++) **(Sep 2023 – Present)**

TA for various modules at KCL. Most involved in the **Operating Systems & Concurrency** module.

- Taught small groups, supervised coursework and demonstrated during lectures.
- I create additional content for the students, both module-specific (e.g. revision sessions) and related content.
- For example, an in-depth look at **Linux's scheduler** including a visual demonstration of the impact of nice values.
- As well as additional sessions about **ELF**, **x64 page tables** and **interrupt handlers**.
- Some feedback: *"very engaging, and you could tell that he has a thorough and in-depth knowledge of the content."*

Projects

Machine Learning Classifier for Acute Kidney Injury (C) view repository

A ML Classifier for Acute Kidney Injury (AKI) with tight performance constraints.

- Implement (from **absolute scratch**) a random-forest classifier for detecting AKI in C. (`libc` is used).
- Model achieves an F_3-score of 0.985, correctly classifying 1180/1200 people with AKI.
- Performs inference (including data parsing) in $< 2\mu s$ and trains in $< 150ms$ on 13,000 data entries.
- Use novel methods to compress training data. An article is available on my website here (easy read).

Associative Array Research (C & C++) view repository

Researched a variety of associative array implementations, in an effort to produce a very fast one.

- Create a testing and benchmark suite for hashtables for any language that supports the C ABI.
- Implement many different hashtable designs (robin-hood hashing, quadratic-probing, chaining etc. **SIMD-lookup**).
- Current best effort shows up to a 500% improvement on `std::unordered_map`.
- Formally describe the complexity of probing hashtable lookups (as well as other articles about my research).

libhopeful – Tracing Heap Allocations (Rust & C) view repository

Build inspectable graphs of the allocations active Unix processes. 'Tracing' is used here in the 'tracing GC' sense.

- Consume DWARF debug info, such that we can attempt to link any `T` to a representation.
- **Lock-free** data-structure using `std::sync::atomic` (this is practically identical to **C++ atomics**) to look up meta-data for interior pointers.
- Considerable investigation into the operational semantics surrounding allocation in Rust (ask me about it!).

Compiler for a Functional Language (Rust) view repository

Implemented a small purely functional language. The entire list of features is documented on the GitHub page.

- CLI build tool for compiling and running programs.
- Clear and precise error messages with syntax highlighting and exact error location.
- Statically typed – lowers to **LLVM-IR**

Skills

tokio; axum; wgpu; wgsl; bevy; Linux; Windows; C; C++; C#; .NET; Unity; Lua; Python; Django; TypeScript; JavaScript; axios; React; HTML; CSS; Java; Scala; Redux; Prisma; REST API; OOP; Functional Programming; Web-Dev; TCP/IP; Serialization; gdb; Haskell;