

# Guanghao Ye

http://yeguanghai.xyz/

Last updated on January 18, 2019

Email : yeguanghai1999@gmail.com

Mobile : +1-206-228-7256

## PROGRAMMING SKILLS

---

- Proficient: C/C++, Python, Java, SQL, Standard ML, Algorithm Design and Analysis
- Familiar:  $\text{\LaTeX}$ , Golang, Racket, Haskell, Shell, Amazon AWS, Microsoft Azure, Apache Spark

## EDUCATION

---

- **University of Washington** Seattle, WA  
*Bachelor of Science; Computer Science, Mathematics; GPA: 3.86* *Aug. 2017 – Exp Jun. 2021*  
**Undergraduate Coursework:**
  - Software Design and Implementation
  - The Hardware/Software Interface
  - Systems Programming
  - Programming Languages
  - Database
  - Introduction to Algorithms**Graduate Coursework:**
  - Computational Complexity
  - Design and Analysis of Algorithms
  - Algorithms through Geometric Lens
  - Theory of Convex Optimization
  - Probabilistic Combinatorics



## EXPERIENCE

---

- **University of Washington** Seattle, WA  
*Teaching Assistant* *Mar. 2018 - Present*  
Assisted with course planning and development, graded students' homework, and held weekly office hours.
  - CSE 421: Introduction to Algorithm 2018 Spring Instructor: Yin Tat Lee
  - CSE 421: Introduction to Algorithm 2018 Autumn Instructor: Yin Tat Lee
  - CSE P521: Applied Algorithms 2019 Winter Instructor: Anna Karlin
- **Chengdu No.7 High School** Chengdu, China  
*Software Engineer* *Mar. 2017 - Aug. 2017*  
Wrote an Online Judge System based on Python, Django, and Docker to help students to test their programs of competitive programming problems and to help coaches to organize contests online.

## PROJECTS

---

- **File System Search Engine** *Jun. 2018*  
Wrote a file system search engine using C++. It builds on-disk indices and has a POSIX multi-thread web server built on C++11 and Boost which support thread pool. This is a course project for *CSE 333 System Programming*.  Github
- **Top Trees C++ Implementation** *Aug. 2016*  
Wrote a C++ implementation of *Tarjan's self-adjusting top trees* which can maintain information in a fully dynamic forest. Here fully dynamic means that edges may be both inserted and deleted.  Github

## HONORS

---

- **Recipient:** Microsoft Endowed Scholarship *Sept. 2018 – Jun. 2019*
- **Rank 5:** International Collegiate Programming Contest, Pacific Northwest *Nov. 2018*
- **Rank 5:** International Collegiate Programming Contest, Pacific Northwest *Nov. 2017*
- **Rank 7:** Internet Problem Solving Contest 2017, High School Division *Jul. 2017*
- **Rank 1:** USA Computing Olympiad 2016-2017 First Contest *Dec. 2016*
- **Dean's List:** University of Washington Seattle Annual Dean's List *Sept. 2017 – Present*