Overview

- This course studies **design** and **analysis** of algorithms
  - **Design**: Methods used to create new algorithms to solve problems (e.g. greedy, dynamic programming, divide and conquer)
  - **Analysis**: Mathematical (as opposed to empirical) assessment of an algorithm’s **correctness** and **efficiency**

Correctness and Efficiency

- **Correctness**: Does the algorithm do what it is supposed to do on all inputs?
  - Could be an infinite or exponential number of inputs, so cannot typically do this empirically
- **Efficiency**: Measuring the algorithm’s running time
  - Count number of basic operations (e.g. number of comparisons in sorting)

Efficiency

- Typically focus on the **worst-case, asymptotic** performance
- E.g. an algorithm with an input of size $n$ takes $O(n^2)$ time steps on all inputs
- Other analyses, such as **average case**, can be done but are not as common