What is an algorithm? What is a data structure?

**Definition**

- An **algorithm** is a step-by-step procedure to solve a problem in a finite amount of time.
- A **data structure** is a systematic way of storing, organizing, modifying, and accessing data.
  (Good data structures help in designing good algorithms.)

Input data → **Algorithm** → Output data
Why learn algorithms? (Applications)

<table>
<thead>
<tr>
<th>Field/application</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Search</td>
<td>Ranking/relevance algorithms</td>
</tr>
<tr>
<td>Internet routing/Google Maps</td>
<td>Shortest path algorithms</td>
</tr>
<tr>
<td>Secure emails/ecommerce</td>
<td>Cryptographic algorithms</td>
</tr>
<tr>
<td>Pharmaceuticals</td>
<td>Computational biology algorithms</td>
</tr>
<tr>
<td>Movie graphics</td>
<td>Geometric algorithms</td>
</tr>
<tr>
<td>Stock market prediction</td>
<td>Machine learning algorithms</td>
</tr>
<tr>
<td>Image processing</td>
<td>Computer vision algorithms</td>
</tr>
<tr>
<td>Amazon/Netflix/Facebook</td>
<td>Recommendation algorithms</td>
</tr>
<tr>
<td>IBM Watson</td>
<td>Artificial intelligence algorithms</td>
</tr>
<tr>
<td>Companies/banks/hospitals</td>
<td>Database algorithms</td>
</tr>
<tr>
<td>Physics/chemistry/biology</td>
<td>Simulation algorithms</td>
</tr>
<tr>
<td>Compilers</td>
<td>Automata algorithms</td>
</tr>
<tr>
<td>Evolution</td>
<td>Evolutionary algorithms</td>
</tr>
<tr>
<td>Cooking</td>
<td>Cooking algorithms (recipes)</td>
</tr>
</tbody>
</table>
Why learn algorithms? (Understanding universe)

- Humanities
- Psychology
- Biology
- Chemistry
- Physics
- Mathematics
- Computation
Why learn algorithms? (Designing software)
Good computer program = Good algorithm
+ Good data structure
+ Good programming language
+ Good coding
What is the measure of goodness of algorithms?

- Correctness
- Running time
- Space-efficiency
- Clarity
- Functionality
- Modularity
- Generality
- Reliability
- Maintainability
- Robustness
- Testability
- Debuggability
- Usability
- Energy-efficiency
Data structures

- No Searching
  - List
    - ArrayList (dynamic array)
    - Stack (circular dynamic array)
    - Queue (circular dynamic array)
    - Deque (circular dynamic array)
  - FIFO
  - WIFO
  - WIPO
    - PriorityQueue (array-based heap)

- Searching
  - Hashtables
    - HashSet (hashtable + chaining)
    - HashMap (hashtable + chaining)
    - Trees
      - TreeSet (red-black trees)
      - TreeMap (red-black trees)
    - Skip lists
      - SkipListSet (skip list)
      - SkipListMap (skip list)

- Sorting
  - No Sorting
  - Sorting