Link youtube of this course: <https://www.youtube.com/watch?v=2WtPyqwTLKM&list=PLhQjrBD2T383q7Vn8QnTsVgSvyLpsqL_R&index=3>

**Lecture 1:**

1. Binary:

Computers only understand electronic format signals: on/off, so all data is encoded as bits - the smallest form in a computer, equivalent to a value of 0 or 1.

1. ASCII:

Responsibilities of ASCII in Software Engineering:

* + Communication within systems: ASCII is the standard encoding format that allows systems to understand and exchange information in string format.
  + String handling: In programming, each character in a string is usually represented by an ASCII code. For example, 'A' is equivalent to 65.
  + Character comparison: When comparing two characters, the computer compares their ASCII values.
  + Character case transformation: ASCII enables transformation between uppercase and lowercase characters.

1. Unicode:

a, Unicode is a character encoding standard designed to:

* + Represent all characters from different writing systems (Latin, Cyrillic, Chinese, Arabic, Devanagari, etc.),
  + Support special symbols (mathematical symbols, emojis, control characters, etc.),
  + Solve the problem of limited character sets in ASCII or local encodings (like ISO, TCVN, etc.).

+ ASCII uses only 7 bits (128 characters)

+ Unicode can use up to 32 bits, representing over 1 million characters (currently ~143,000 characters are defined).

**Unicode does not define how characters are stored in binary, but rather assigns a unique number to each character, called a code point.**

b, What is UTF (Unicode Transformation Format)?

Unicode requires an encoding method to store data in memory or transmit over networks. The most common methods are:

* + UTF-8: The most widely used today. Uses 1 to 4 bytes per character.
  + UTF-16: Uses 2 or 4 bytes per character.
  + UTF-32: Uses exactly 4 bytes per character.

Advantage of UTF-8: Backward compatible with ASCII and efficient for English text.

1. Color (RGB)

Applications in Software Engineering:

* + Image processing: Digital color images are usually stored as RGB matrices.
  + Web design (CSS): Used to define element colors.
  + Computer vision: RGB values help detect objects or features.
  + Deep learning for images: RGB images are commonly used as input for CNN models.

In Python:

|  |
| --- |
| import cv2  image = cv2.imread('flower.jpg') # Reads the image → by default, it's BGR  b, g, r = image[100, 100] # Get pixel color at position (100, 100)  print(f"Red: {r}, Green: {g}, Blue: {b}") |

Note: OpenCV reads images in **BGR** order, not RGB.

1. Pseudocode

Pseudocode is a way of writing algorithms using a language that resembles natural English (or your native language), without strictly following the syntax of any specific programming language.

What is Pseudocode?

* + It serves as an intermediate step between thinking through an algorithm and writing actual code in Python, C++, Java, etc.
  + It is easy to read and understand, focuses on logic, and does not emphasize syntax.
  + Pseudocode is commonly used in algorithm design, teaching, or technical interviews.

**Lecture 2: VS Code and C:**

**Terminal Command:**

* + **Lưu ý: xem lại cách chạy make**
    1. Header file
    2. Terminal Command:
    3. Conditionals
    4. Variables – Constant
    5. Compare
    6. Loop
    7. Function
    8. Calculator

9. What is Integer Overflow?

Integer Overflow is a phenomenon that occurs when an integer variable exceeds the maximum (or minimum) value that its data type can store.

Integer Overflow là hiện tượng xảy ra khi một biến số nguyên (integer) vượt quá giới hạn giá trị mà kiểu dữ liệu của nó có thể lưu trữ được.

* + 1. Boeing: Example for interger Overflow
    2. Pacman: Example for interger Overflow
    3. Truncation
    4. Truncation là quá trình loại bỏ phần thập phân hoặc cắt bỏ phần vượt quá giới hạn

Truncation is the process of removing decimals or cutting off excess parts

* + 1. Casting: Ép kiểu
    2. Floating point imprecision
  + Floating Point Imprecision là gì?

What is Floating Point Imprecision?

Floating point imprecision refers to the small inaccuracies that occur when computers represent and calculate real (decimal) numbers, due to limitations in binary storage format.

* + Why does it happen?

Computers store numbers in binary, not decimal like humans do.

However:

Some decimal numbers (like 0.1, 0.2) cannot be represented exactly in binary.

As a result, when these numbers are stored or used in calculations, tiny rounding errors occur.

* + Ví dụ điển hình:

print(0.1 + 0.2) # Kết quả: 0.30000000000000004 ❗️

Bạn mong đợi ra 0.3, nhưng Python (và hầu hết các ngôn ngữ) sẽ ra 0.30000000000000004 do biểu diễn nhị phân của 0.1 và 0.2 là xấp xỉ, không chính xác tuyệt đối.

**Lecture3: Array**

* + 1. Compling:

Compiling là quá trình chuyển mã nguồn (source code) bạn viết bằng ngôn ngữ lập trình như C, C++, Java... → thành mã máy (machine code) mà máy tính có thể hiểu và chạy được.

Trong Python: Python là ngôn ngữ thông dịch (interpreted), tức là vừa dịch vừa chạy từng dòng code

C: C là ngôn ngữ biên dịch: dịch hết sang ngôn ngữ máy rồi mới chạy, nên quá trình chạy diễn ra rất nhanh

* + 1. Debugging: qua trinh nguoc lai cua compling
    2. Data type
    3. Array
    4. String
    5. Ép kiểu từ lớn sang nhỏ mất dữ liệu và interger overflow

A screenshot of a computer

AI-generated content may be incorrect.

* + 1. So sánh mảng trong C và list trong python: A screenshot of a computer program

       AI-generated content may be incorrect.
    2. Upper case:

Trong C cần trừ đi 32 khi ép kiểu sang int, sau đó ép kiểu ngược lại string thì ok

Trong Python có hàm sẵn

* + 1. Exit status: return
    2. Encryption

**key + plaintext → [ cipher ] → ciphertext**

**Lecture3: Algorithms**

1. **Binary search**

**Tìm kiếm nhị phân:** Binary Search là một thuật toán tìm kiếm trong một dãy đã sắp xếp, bằng cách chia đôi khoảng tìm kiếm sau mỗi bước.

Nó nhanh hơn rất nhiều so với tìm kiếm tuần tự (linear search).

Độ phức tạp: O(1) => O(logn)

1. Cần luyện tập viết hàm đệ quy
2. Running time or Time Complexity (Độ phức tạp của thuật toán)

| **Trường hợp** | **Kí hiệu** | **Ý nghĩa** |
| --- | --- | --- |
| ✅ **Tốt nhất** | **Ω (Omega)** | Thời gian ít nhất cần thiết để hoàn thành (ví dụ: tìm thấy ngay lần đầu) |
| ⚖️ **Trung bình** | **Θ (Theta)** | Thời gian trung bình cho tất cả các trường hợp đầu vào |
| ❌ **Xấu nhất** | **O (Big-O)** | Thời gian nhiều nhất có thể xảy ra (đảm bảo không tệ hơn mức này) |

1. Struct in C

A black background with white text

AI-generated content may be incorrect.

1. Selection sort and Bubble sort

Có độ phức tạp thuật toán rất lớn O(n^2)

1. Recursion
2. Merge sort: Running time: O(nlogn)

Khi muốn rút ngắn thời gian thì cần tốn thêm bộ nhớ

Merge sort:

Tách ra tnay xong sắp xếp từ dưới lên => nhanh hơn

**Merge Sort là gì? (muốn ôn lại thì xem lại vid, hay lắm)**

Merge Sort là thuật toán sắp xếp theo phương pháp chia để trị (Divide and Conquer), được chia làm 3 bước chính:

* + Chia (Divide): Chia dãy thành 2 nửa nhỏ hơn (đệ quy)
  + Trị (Conquer): Sắp xếp từng nửa bằng chính merge sort
  + Trộn (Combine): Gộp hai dãy đã sắp xếp lại thành dãy hoàn chỉnh

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Code:

|  |
| --- |
| def merge\_sort(arr):  if len(arr) <= 1:  return arr # Mảng 1 phần tử thì đã sắp xếp  mid = len(arr) // 2  left = merge\_sort(arr[:mid]) # Sắp xếp nửa trái  right = merge\_sort(arr[mid:]) # Sắp xếp nửa phải  return merge(left, right)  def merge(left, right):  result = []  i = j = 0  # So sánh và trộn  while i < len(left) and j < len(right):  if left[i] <= right[j]:  result.append(left[i])  i += 1  else:  result.append(right[j])  j += 1  # Thêm phần còn lại  result.extend(left[i:])  result.extend(right[j:])  return result |

**Lecture4: Hexadecimal**

* + 1. Hexadecimal

Cơ số 16 nghĩa là có 16 ký tự để biểu diễn giá trị.

Các chữ số trong hệ hex gồm:

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

* + 1. Các cách chuyển từ thập phân sang nhị phân:
  + Chuyển từng chữ cái (hoặc chữ số) sang nhị phân rồi ghép vào(chính xác và trực tiếp)
  + Chuyển sang hệ thập phân sau đó đổi sang nhị phân (đúng về mặt giá trị)