

Nhật ký thực hiện dự án: LZW Compression & Decompression Tool

◆ Bắt đầu dự án (Ngày 1)

- Sau khi đọc yêu cầu từ giảng viên, nhóm quyết định chọn đề tài xây dựng công cụ nén/giải nén file sử dụng thuật toán **LZW** vì:
 - LZW là một thuật toán điển hình, phù hợp kiểm thử và biểu diễn rõ quy trình nén.
 - Đề tài vừa có tính thực tiễn vừa giúp ôn lại kiến thức về cấu trúc dữ liệu, GUI.
-

◆ Thiết kế và tìm hiểu (Ngày 2–3)

- Tìm hiểu lại nguyên lý thuật toán LZW:
 - Cách tạo dictionary từ các chuỗi byte.
 - Cách ánh xạ chuỗi → mã → file nén.
 - Thiết kế sơ đồ module: tách riêng `lzw.py`, `file_handler.py`, `gui_app.py`.
 - Phân công:
 - 1 bạn phụ trách cốt lõi thuật toán (`lzw.py`).
 - 1 bạn làm giao diện Tkinter.
 - 2 bạn làm phần xử lý file, kiểm thử và tài liệu.
-

◆ Cài đặt thuật toán và xử lý file (Ngày 4–5)

- Viết xong thuật toán `lzw_compress()` và `lzw_decompress()`, kiểm thử bằng file `.txt`, `.png`.
 - Tạo module `file_handler.py` để:
 - Ghi file `.lzw` theo định dạng gồm: định dạng ký mã, đuôi file, SHA256 nếu có mật khẩu.
 - Đảm bảo file nén có thể dễ dàng tái cấu trúc khi giải nén.
 - Thêm logic bảo vệ bằng mật khẩu (SHA256).
-

◆ Giao diện và xử lý người dùng (Ngày 6–7)

- Dùng Tkinter để làm giao diện nén/giải nén, thêm các nút chức năng:
 - Compress, Decompress
 - Batch compress (nhiều file)
 - Show Chart, Export CSV

- Giao diện sử dụng matplotlib để vẽ biểu đồ tỷ lệ nén.
-

♦ Kiểm thử và cải tiến (Ngày 8–9)

- Kiểm thử chức năng nén/giải nén nhiều loại file: .txt, .jpg, .pdf, .exe.
 - Phát hiện lỗi:
 - Trường hợp file nén ra lớn hơn file gốc → tự động xóa file .lzw và thông báo người dùng.
 - Viết thêm ShowDict.py và ReadLZW.py để hỗ trợ debug và thuyết trình.
-

♦ Đóng gói và hoàn thiện (Ngày 10–12)

- Build thành .exe bằng pyinstaller.
- Viết file README.txt, hoàn tất file TM.
- Nén toàn bộ mã nguồn, tài liệu theo đúng định dạng FITSP25PRJmsv.zip.