

MÔ PHỎNG GIẢI THUẬT BANKER

1. Mục đích:

- Thể hiện được giải thuật Banker đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu deadlock, điều kiện cần của deadlock
- Tìm hiểu các phương pháp giải quyết deadlock.
- Tìm hiểu thuật toán Banker: Giải thuật xác định chuỗi cấp phát an toàn.
- Tìm hiểu thuật toán Banker: Giải thuật yêu cầu cấp phát tài nguyên.

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng giải thuật Banker (Nhà băng) của Dijkstra để tránh deadlock (tắc nghẽn):

- Input: Số lượng tiến trình, các nguồn tài nguyên, các yêu cầu của hệ thống, số tài nguyên đã cấp phát cho từng tiến trình, số tài nguyên hệ thống có sẵn.
- Output:
 - ✓ Thực hiện cách xét trạng thái hiện thời của hệ thống là an toàn hay không an toàn.
 - ✓ Nếu trạng thái là an toàn, chương trình phải chỉ ra cách để thu hồi tài nguyên hệ thống.
 - ✓ Với mỗi yêu cầu cung cấp tài nguyên, hãy hiển thị ra màn hình cách cấp phát, thu hồi tương ứng.

3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

ĐỀ TÀI THỰC TẬP CƠ SỞ

MÔ PHỎNG CÁC CHIẾN LƯỢC ĐIỀU PHỐI TIẾN TRÌNH

1. Mục đích:

- Thể hiện được các giải thuật điều phối tiến trình đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu tiến trình, cách tổ chức điều phối đa tiến trình.
- Tìm hiểu các chiến lược điều phối tiến trình:
 - ✓ Chiến lược công việc đến trước (First Come First Served - FCFS)
 - ✓ Chiến lược công việc ngắn nhất (Shortest Job First – SJF)
 - ✓ Chiến lược thời gian còn lại ngắn nhất (Sortest Remaining Time First – SRTF)
 - ✓ Chiến lược phân phối xoay vòng (Round Robin – RR)
 - ✓ Chiến lược điều phối theo độ ưu tiên độc quyền
 - ✓ Chiến lược điều phối theo độ ưu tiên không độc quyền

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng các chiến lược điều phối tiến trình.

- Input: Số lượng tiến trình và tên tương ứng; Thời điểm vào danh sách hàng đợi, Thời gian CPU, Độ ưu tiên của mỗi tiến trình.
- Output:
 - ✓ Thời gian chờ của mỗi tiến trình và thời gian chờ trung bình của từng chiến lược.
 - ✓ Thời gian lưu lại trong hệ thống của mỗi tiến trình và thời gian lưu lại trung bình.
 - ✓ Đánh giá độ hiệu quả của từng giải thuật.
- Với mỗi yêu cầu cung cấp tài nguyên, hãy hiển thị ra màn hình cách cấp phát, thu hồi tương ứng.

3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

MÔ PHỎNG CÁC CHIẾN LƯỢC CẤP PHÁT BỘ NHỚ

1. Mục đích:

- Thể hiện được các giải thuật cấp phát bộ nhớ đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu việc quản lý bộ nhớ trong hệ đa tiến trình.
- Tìm hiểu việc cấp phát bộ nhớ theo mô hình phân trang, phân đoạn.
- Tìm hiểu các chiến lược cấp phát bộ nhớ:
 - ✓ Chiến lược First Fit
 - ✓ Chiến lược Best Fit
 - ✓ Chiến lược Worst Fit
 - ✓ Chiến lược Next Fit

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng các chiến lược cấp phát bộ nhớ.

- Input: Số lượng phân mảnh bộ nhớ và kích thước tương ứng; Số lượng tiến trình và kích thước tương ứng của từng tiến trình.
- Output:
 - ✓ Giao diện đồ họa, minh họa quá trình cấp phát bộ nhớ của từng chiến lược: First Fit, Best Fit, Worst Fit và Next Fit.
 - ✓ Tính kích thước phân mảnh bộ nhớ của từng vùng.
 - ✓ Kết luận tiến trình nào được cấp phát, vị trí cấp phát. Tiến trình nào không được cấp phát.
 - ✓ Đưa ra nhận xét độ hiệu quả của từng giải thuật.

3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

MÔ PHỎNG CÁC CHIẾN LƯỢC THAY THẾ TRANG BỘ NHỚ

1. Mục đích:

- Thể hiện được các giải thuật thay thế trang bộ nhớ đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu về bộ nhớ ảo, cơ chế thực hiện bộ nhớ ảo: phân trang theo yêu cầu.
- Tìm hiểu các thuật toán thay thế trang:
 - ✓ Thuật toán FIFO
 - ✓ Thuật toán tối ưu OPT (Optimum)
 - ✓ Thuật toán “Lâu nhất chưa sử dụng” (Least Recently Used - LRU)
 - ✓ Thuật toán cơ hội thứ 2 (Second chance)

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng các thuật toán thay thế trang.

- Input: Số khung trang, chuỗi truy xuất bộ nhớ.
- Output:
 - ✓ Giao diện đồ họa, minh họa quá trình thay thế trang của từng giải thuật: FIFO, OPT, LRU và Second chance.
 - ✓ Đánh dấu * và khoanh tròn vào vị trí lỗi trang.
 - ✓ Tính tổng số lượng lỗi trang (Page fault) của từng giải thuật.
 - ✓ Đưa ra nhận xét độ hiệu quả của từng giải thuật.

3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

MÔ PHỎNG CÁC THUẬT TOÁN ĐIỀU PHỐI ĐĨA

1. Mục đích:

- Thể hiện được các giải thuật điều phối đĩa đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu về ổ đĩa cứng, cơ chế quản lý đĩa.
- Tìm hiểu các thuật toán điều phối đĩa:
 - ✓ Thuật toán điều phối theo FCFS (First Come First Served)
 - ✓ Thuật toán điều phối theo SSTF (Shortest Seek Time First)
 - ✓ Thuật toán điều phối theo SCAN
 - ✓ Thuật toán điều phối theo C-SCAN
 - ✓ Thuật toán điều phối theo LOOK
 - ✓ Thuật toán điều phối theo C-LOOK

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng các thuật toán điều phối đĩa.

- Input: Hàng đợi đĩa gồm các yêu cầu đọc dữ liệu tại các cylinder theo thứ tự cho trước; Vị trí của đầu đĩa.
- Output:
 - ✓ Giao diện đồ họa, minh họa quá trình điều phối đĩa của từng giải thuật: FCFS, SSTF, SCAN, C-SCAN, LOOK và C-LOOK.
 - ✓ Vẽ mũi tên biểu diễn đường đi và thứ tự di chuyển của đầu đĩa tương ứng theo từng giải thuật.
 - ✓ Tính tổng quãng đường di chuyển (số cylinder) của đầu đĩa theo từng giải thuật.
 - ✓ Đưa ra nhận xét độ hiệu quả của từng giải thuật.

3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

BÀI TOÁN 05 TRIẾT GIA ĂN TỐI

1. Mục đích:

- Thể hiện được các giải thuật đồng bộ tiến trình đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu về các phương pháp đồng bộ hóa tiến trình.
 - ✓ Busy - Waiting
 - ✓ Sleep and Wake up
- Tìm hiểu bài toán “Bữa ăn tối của các triết gia” được đưa ra bởi Dijkstra là một bài toán kinh điển về đồng bộ hoá trong môi trường đa luồng và các kỹ thuật để giải quyết bài toán. Phải đặt ra thuật toán sao cho khi một triết gia bị đói thì ông ta sẽ được ăn và đảm bảo không có triết gia nào bị chết đói.

Về cài đặt, thực nghiệm:

Viết chương trình minh họa bữa ăn tối của các triết gia, sử dụng Semaphore.

- Input: 5 tiến trình con miêu tả hoạt động của 5 triết gia; 5 chiếc đũa/nĩa tương ứng cho mỗi triết gia.

- Output:

Mức 1:

- ✓ Tìm một phương pháp đảm bảo để các triết gia đều có thể được ăn, hệ thống không bị tắc nghẽn.
- ✓ Thể hiện thứ tự của các triết gia (ăn và suy nghĩ).

Mức 2

- ✓ Minh họa thuật toán dưới dạng đồ họa.

3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

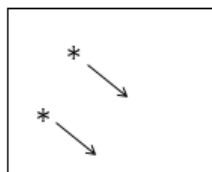
ÁP DỤNG SEMAPHORE MÔ PHỎNG CHUYỂN ĐỘNG CỦA QUẢ BANH TRÊN MÀN HÌNH

1. Mục đích:

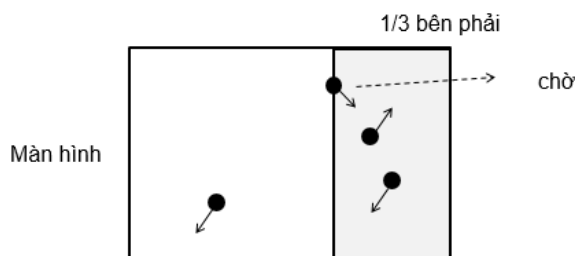
- Thể hiện được các giải thuật đồng bộ tiến trình đã học trong môn học “Hệ điều hành” dưới dạng chương trình hoàn chỉnh.

2. Yêu cầu:

Viết chương trình cho phép 2 (và tổng quát là n) trái banh bay cùng lúc trong màn hình.



Áp dụng Semaphore, sửa chương trình trái banh bay trên, sao cho tại một thời điểm chỉ có tối đa 2 quả banh đi vào 1/3 phải màn hình.



3. Tài liệu tham khảo chính:

- Phạm Thị Thu Thúy, Bài giảng Hệ điều hành, Trường Đại học Nha Trang, 2021.

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

MÔ PHỎNG THUẬT TOÁN CHUẨN HÓA QUAN HỆ (NORMALIZATION)

1. Mục đích:

Sinh viên có thể xây dựng một công cụ hoặc mô phỏng quy trình chuẩn hóa cơ sở dữ liệu từ dạng không chuẩn đến chuẩn hóa 1NF, 2NF, 3NF và BCNF. Đề tài này sẽ giúp sinh viên hiểu và thực hành quy trình tối ưu hóa thiết kế cơ sở dữ liệu bằng cách loại bỏ dư thừa và đảm bảo tính toàn vẹn dữ liệu.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu các dạng chuẩn của cơ sở dữ liệu quan hệ.
- Tìm hiểu các phương pháp chuẩn hóa CSDL.

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng quy trình chuẩn hóa cơ sở dữ liệu từ dạng không chuẩn đến chuẩn hóa 1NF, 2NF, 3NF và BCNF:

- Input: Cho phép nhập dữ liệu từ bàn phím hoặc từ file. Cho lược đồ quan hệ, tập thuộc tính, tập phụ thuộc hàm hoặc cho quan hệ.
- Output:
 - ✓ Kiểm tra lược đồ quan hệ ở dạng chuẩn mấy. Giải thích.
 - ✓ Đưa lược đồ quan hệ lên các dạng chuẩn cao hơn. Giải thích.
 - ✓ Mô phỏng thuật toán bằng đồ họa nếu có thể.

3. Tài liệu tham khảo chính:

- Nguyễn Đức Thuận, Giáo trình Cơ sở dữ liệu quan hệ, Trường Đại học Nha Trang, 2007.
- Nguyễn Đức Thuận, Trương Ngọc Châu, Phương pháp giải bài tập CSDL QH, NXB KH&KT, 2012.
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, Database System: The Complete Book, Pearson India; 2nd edition (January 1, 2013).

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

MÔ PHỎNG THUẬT TOÁN TÌM MỘT KHÓA VÀ TẤT CẢ CÁC KHÓA

1. Mục đích:

Mô phỏng quá trình phân rã lược đồ quan hệ thành các bảng con mà không làm mất dữ liệu. Công cụ này giúp sinh viên hiểu cách thiết kế lược đồ mà vẫn đảm bảo đầy đủ thông tin và phụ thuộc hàm khi phân rã.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu lý thuyết phân rã bảo toàn thông tin và phân rã bảo toàn phụ thuộc hàm.
- Tìm hiểu các thuật toán kiểm tra một phân rã có bảo toàn thông tin hoặc bảo toàn phụ thuộc hàm hay không.

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng thuật toán kiểm tra một phân rã có bảo toàn thông tin và bảo toàn phụ thuộc hàm hay không:

- Input: Cho phép nhập dữ liệu từ bàn phím hoặc từ file. Cho lược đồ quan hệ, tập thuộc tính, tập phụ thuộc hàm, tập các phân rã.
- Output:
 - ✓ Kiểm tra phân rã có bảo toàn thông tin hay không. Giải thích.
 - ✓ Kiểm tra phân rã có bảo toàn phụ thuộc hàm hay không. Giải thích.
 - ✓ Mô phỏng thuật toán bằng đồ họa nếu có thể.

3. Tài liệu tham khảo chính:

- Nguyễn Đức Thuận, Giáo trình Cơ sở dữ liệu quan hệ, Trường Đại học Nha Trang, 2007.
- Nguyễn Đức Thuận, Trương Ngọc Châu, Phương pháp giải bài tập CSDL QH, NXB KH&KT, 2012.
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, Database System: The Complete Book, Pearson India; 2nd edition (January 1, 2013).

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

MÔ PHỎNG THUẬT TOÁN KIỂM TRA PHÂN RÃ CÓ BẢO TOÀN THÔNG TIN VÀ PHỤ THUỘC HÀM**1. Mục đích:**

Xây dựng một ứng dụng xác định khóa chính trong các lược đồ quan hệ, dựa vào phụ thuộc hàm. Điều này giúp sinh viên nắm vững khái niệm về khóa chính và vai trò của nó trong thiết kế cơ sở dữ liệu.

2. Yêu cầu:**Về lý thuyết:**

- Tìm hiểu các loại khóa trong CSDL QH.
- Tìm hiểu các thuật toán tìm một khóa và tìm tất cả các khóa.

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng thuật toán tìm một khóa và tìm tất cả các khóa:

- Input: Cho phép nhập dữ liệu từ bàn phím hoặc từ file. Cho lược đồ quan hệ, tập thuộc tính, tập phụ thuộc hàm.
- Output:
 - ✓ Một khóa của lược đồ quan hệ. Giải thích.
 - ✓ Tất cả các khóa của lược đồ quan hệ. Giải thích.
 - ✓ Mô phỏng thuật toán bằng đồ họa nếu có thể.

3. Tài liệu tham khảo chính:

- Nguyễn Đức Thuận, Giáo trình Cơ sở dữ liệu quan hệ, Trường Đại học Nha Trang, 2007.
- Nguyễn Đức Thuận, Trương Ngọc Châu, Phương pháp giải bài tập CSDL QH, NXB KH&KT, 2012.
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, Database System: The Complete Book, Pearson India; 2nd edition (January 1, 2013).

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn **Phone :** 0901905679

MÔ PHỎNG THUẬT TOÁN TÌM PHỦ TỐI THIỂU CỦA LƯỢC ĐỒ QUAN HỆ (MINIMAL COVER)

1. Mục đích:

Việc tìm phủ tối thiểu giúp đơn giản hóa lược đồ và tối ưu hóa các truy vấn.

2. Yêu cầu:

Về lý thuyết:

- Tìm hiểu khái niệm phụ thuộc hàm, bao đóng của tập thuộc tính.
- Tìm hiểu định nghĩa phủ tối thiểu và thuật toán tìm phủ tối thiểu.

Về cài đặt, thực nghiệm:

Viết chương trình mô phỏng thuật toán tìm phủ tối thiểu của lược đồ quan hệ:

- Input: Cho phép nhập dữ liệu từ bàn phím hoặc từ file. Cho lược đồ quan hệ, tập thuộc tính, tập phụ thuộc hàm.
- Output:
 - ✓ **Bước 1:** Chia các phụ thuộc hàm có nhiều thuộc tính bên phải thành các phụ thuộc có một thuộc tính bên phải.
 - ✓ **Bước 2:** Loại bỏ các thuộc tính không cần thiết ở phía bên trái của các phụ thuộc hàm.
 - ✓ **Bước 3:** Loại bỏ các phụ thuộc hàm dư thừa để tạo ra một tập phụ thuộc hàm tối giản.
 - ✓ Phụ thuộc hàm tối thiểu

3. Tài liệu tham khảo chính:

- Nguyễn Đức Thuần, Giáo trình Cơ sở dữ liệu quan hệ, Trường Đại học Nha Trang, 2007.
- Nguyễn Đức Thuần, Trương Ngọc Châu, Phương pháp giải bài tập CSDL QH, NXB KH&KT, 2012.
- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, Database System: The Complete Book, Pearson India; 2nd edition (January 1, 2013).

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

Mô phỏng thuật toán Dijkstra & BFS/DFS

1. Mục đích:

Mô phỏng các thuật toán tìm đường trong đồ thị.

2. Yêu cầu:

- Tìm hiểu BFS, DFS và Dijkstra
- Cho phép người dùng nhập đồ thị (ma trận hoặc danh sách cạnh)
- Hiển thị bước duyệt và đường đi ngắn nhất

Output

- Màu sắc trực quan các nút và cạnh được duyệt
- Bảng so sánh số bước, độ phức tạp

3. Tài liệu tham khảo chính:

- Nguyễn Thanh Thủy, Đỗ Văn Nhơn. *Giáo trình Cấu trúc dữ liệu và Giải thuật*, NXB Giáo dục.
- Trương Tấn Quang. *Giáo trình Toán rời rạc*, NXB Giáo dục.
- GeeksforGeeks – Graph Algorithms. <https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/>.
- VisuAlgo – Dijkstra & BFS/DFS Visualization. <https://visualgo.net/en/graph>
- Khan Academy – Graphs & Shortest Path <https://www.khanacademy.org/computing/computer-science/algorithms>

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

Mô phỏng thuật toán Kruskal & Prim – Cây khung nhỏ nhất (MST)

1. Mục đích:

Giúp sinh viên hiểu rõ cách xây dựng cây khung nhỏ nhất trong đồ thị.

2. Yêu cầu:

- Tìm hiểu khái niệm đồ thị, trọng số, cây khung nhỏ nhất
- Cài đặt thuật toán Kruskal & Prim
- Cho phép người dùng nhập hoặc sinh đồ thị ngẫu nhiên
- Hiển thị từng bước chọn cạnh + phát hiện chu trình (Union-Find với Kruskal)

Output

- Vẽ đồ thị, tô màu các cạnh được chọn
- So sánh số cạnh duyệt & tổng trọng số giữa 2 thuật toán
- Bảng log các bước thực hiện

3. Tài liệu tham khảo chính:

- Nguyễn Thanh Thủy, Đỗ Văn Nhơn. *Giáo trình Cấu trúc dữ liệu và Giải thuật*, NXB Giáo dục.
- Trương Tấn Quang. *Giáo trình Toán rời rạc*, NXB Giáo dục.
- GeeksforGeeks – Graph Algorithms. <https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/>.
- Visualgo — Kruskal, Prim Animation. <https://visualgo.net/en/mst>
- CP-Algorithms — MST algorithms
https://cp-algorithms.com/graph/mst_kruskal.html

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn Phone : 0901905679

Mô phỏng thuật toán Bellman–Ford & phát hiện chu trình âm

1. Mục đích:

- Hiểu rõ nguyên lý hoạt động của thuật toán Bellman–Ford, đặc biệt là cơ chế “thư giãn cạnh” (edge relaxation).
- Minh họa quy trình tính toán đường đi ngắn nhất từ một đỉnh nguồn đến các đỉnh còn lại trong đồ thị có trọng số âm.
- Phát hiện chu trình âm trong đồ thị (negative cycle detection).
- Tăng cường kỹ năng lập trình mô phỏng thuật toán, tư duy thuật toán và phân tích độ phức tạp.
- Ứng dụng lý thuyết vào các bài toán thực tế như định tuyến mạng (routing), tối ưu hóa chi phí, hệ thống logistic.

2. Yêu cầu:

Yêu cầu lý thuyết

Sinh viên cần tìm hiểu và trình bày:

- Khái niệm đồ thị có trọng số, trọng số âm
- Nguyên lý “relaxation” trong thuật toán ngắn nhất đường đi
- So sánh Bellman–Ford với Dijkstra (ưu, nhược điểm)
- Cách phát hiện chu trình âm bằng Bellman–Ford
- Độ phức tạp thời gian và không gian

Yêu cầu cài đặt & thực nghiệm

- Cho phép nhập đồ thị (số đỉnh, cạnh, trọng số)
- Cho phép chọn đỉnh nguồn
- Hiển thị ma trận cạnh hoặc danh sách cạnh
- Hiển thị bảng cập nhật khoảng cách theo từng vòng lặp:

Vòng lặp	u	v	w(u,v)	dist[v] trước	dist[v] sau:

- Tô màu/đánh dấu cạnh được thư giãn thành công
- Xác định & thông báo nếu tồn tại chu trình âm
- Hiển thị đường đi ngắn nhất từ nguồn đến mỗi đỉnh (nếu tồn tại)

Yêu cầu nâng cao (không bắt buộc nhưng khuyến khích)

- Mô phỏng trực quan đồ thị (node–edge graph)
- So sánh kết quả Bellman–Ford và Dijkstra trên cùng đồ thị
- Hiện thị animation từng bước chạy
- Cho phép tải dữ liệu từ file .txt/.csv
- Chạy thử với các bộ dữ liệu mẫu

Output đầu ra mong muốn

- Màn hình mô phỏng quá trình chạy thuật toán
- Bảng cập nhật theo từng vòng lặp (iteration table)
- Thông báo: => Không tồn tại chu trình âm trong đồ thị Hoặc => Phát hiện chu trình âm: ...
- Vẽ đồ thị và chu trình.

3. Tài liệu tham khảo chính:

- Trương Tấn Quang. *Giáo trình Toán rời rạc*, NXB Giáo dục.
- GeeksforGeeks - Bellman–Ford Algorithm. <https://www.geeksforgeeks.org/bellman-ford-algorithm>
- CP-Algorithms - Bellman–Ford. https://cp-algorithms.com/graph/bellman_ford.html
- Visualgo - Graph Algorithms Simulation. <https://visualgo.net/en/sssp>

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn **Phone :** 0901905679

Mô phỏng thuật toán Floyd–Warshall (Ma trận khoảng cách)

1. Mục đích:

- Hiểu cách sử dụng lập trình động (Dynamic Programming) trong đồ thị.
- Tính đường đi ngắn nhất giữa mọi cặp đỉnh.
- Nắm ma trận chuyển đổi & xử lý qua từng vòng lặp k.
- Rèn luyện kỹ năng phân tích ma trận đồ thị.

2. Yêu cầu:

Yêu cầu lý thuyết

- Trình bày công thức quy hoạch động:

$$\text{dist}[i][j] = \min(\text{dist}[i][j], \text{dist}[i][k] + \text{dist}[k][j])$$

- Vai trò của đỉnh trung gian k.
- Xử lý trọng số âm (nếu không có chu trình âm).

Yêu cầu lập trình

- Input:
 - Ma trận kề & trọng số (∞ nếu không có cạnh)
 - Số đỉnh
- Output hiển thị:
 - Ma trận dist sau mỗi vòng lặp k
 - Bảng log cập nhật giá trị
- Hiển thị:
 - Trực quan đồ thị (tùy chọn)
 - Highlight cập nhật khi có giảm chi phí

Nâng cao

- Lưu lại ma trận truy vết next[][] để xuất đường đi cụ thể
- Kiểm tra và báo lỗi nếu tồn tại chu trình âm

Output mong đợi

- Ma trận khoảng cách cuối cùng
- Log thay đổi từng bước (ma trận sau mỗi k)
- (Tùy chọn) Danh sách đường đi cho từng cặp đỉnh
- File báo cáo + video demo

3. Tài liệu tham khảo chính:

- Cormen et al., *Introduction to Algorithms*, Floyd–Warshall chapter
- Rosen, *Discrete Mathematics and Its Applications*
- TS. Phạm Thị Thu Thúy, *Slide Lý thuyết đồ thị - Floyd-Warshall*
- Visualgo - Floyd–Warshall demo
- CP-Algorithms - Floyd–Warshall

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn **Phone :** 0901905679

Mô phỏng thuật toán tìm cầu (Bridge Detection)

1. Mục đích:

- Hiểu vai trò của cạnh cầu (bridge) trong mạng.
- Áp dụng DFS để tìm cạnh mà nếu xóa sẽ làm đồ thị bị tách.
- Minh họa thuật toán tìm bridge của Tarjan.
- Ứng dụng trong:
 - Mạng xã hội (cầu nối giữa cộng đồng)
 - Hệ thống hạ tầng mạng
 - Graph connectivity analysis

2. Yêu cầu:

Yêu cầu lý thuyết

- Trình bày các khái niệm:
 - Thành phần liên thông
 - Cầu (bridge)
 - DFS tree, back-edge
 - $disc[u]$ & $low[u]$
- Mã giả thuật toán Tarjan

Yêu cầu lập trình

- Input: Số đỉnh & danh sách cạnh
- Output:
 - Danh sách các cạnh là cầu
 - Log giá trị $disc[]$ và $low[]$ theo từng bước DFS
- Hiển thị:
 - Đồ thị
 - Highlight cầu (màu đỏ)
 - Hiển thị thứ tự DFS

Nâng cao

- Visualization từng bước DFS
- So sánh kết quả bằng BFS/DFS kiểm tra liên thông sau xóa cạnh

Output mong đợi

- Danh sách cạnh cầu
- Bảng chỉ số $disc[]$, $low[]$ từng đỉnh

- Đồ thị tô màu cạnh cầu
- Báo cáo & video thuyết trình

3. Tài liệu tham khảo chính:

- Sedgewick & Wayne, *Algorithms*, Chương Graph Algorithms
- Cormen et al., *Introduction to Algorithms* - DFS & Bridges
- TS. Phạm Thị Thu Thúy, *Slide Lý thuyết đồ thị - Thành phần liên thông & Bridge*
- GeeksforGeeks - Bridge in Graph
- CP-Algorithms - Bridge-finding algorithm
- Visualgo - Graph DFS Visualization

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn **Phone :** 0901905679

Mô phỏng biểu diễn đồ thị bằng Ma trận kề & Danh sách kề

1. Mục đích:

- Hiểu rõ các khái niệm cơ bản về đồ thị: đỉnh, cạnh, kề, hướng, trọng số.
- Nắm được hai phương pháp biểu diễn đồ thị trong máy tính:
 - Ma trận kề (Adjacency Matrix)
 - Danh sách kề (Adjacency List)
- So sánh ưu – nhược điểm của từng cách biểu diễn.
- Rèn luyện kỹ năng lập trình cấu trúc dữ liệu đồ thị và thao tác với dữ liệu mạng.
- Tăng cường khả năng trực quan hóa cấu trúc đồ thị và hiểu độ phức tạp lưu trữ.

2. Yêu cầu:

Yêu cầu lý thuyết

Sinh viên phải trình bày:

- Định nghĩa đồ thị (vô hướng, có hướng, có trọng số, đơn đồ thị).
- Khái niệm đỉnh kề, cạnh kề, bậc đỉnh.
- Định nghĩa & mô tả:
 - Ma trận kề
 - Danh sách kề

Yêu cầu lập trình

Xây dựng chương trình cho phép:

- Nhập đồ thị từ bàn phím/ file:
 - Số đỉnh
 - Danh sách cạnh
- Chọn loại đồ thị:
 - Vô hướng / có hướng
 - Có trọng số / không trọng số
- Tự động sinh cấu trúc:
 - Ma trận kề
 - Danh sách kề
- Chức năng hiển thị:
 - In ma trận kề dạng bảng
 - In danh sách kề dạng danh sách
 - Vẽ đồ thị trực quan (node–edge graph)

Yêu cầu nâng cao (khuyến khích)

- Cho phép thêm/ xóa đỉnh/ cạnh
- Xuất file cấu trúc

- Đọc & hiển thị đồ thị Karate Club (34 nodes)
- Đánh giá đồ thị thưa hay dày

Output mong đợi

Hiển thị dữ liệu

- Ma trận kề ví dụ:

```

      A B C D
A 0 1 1 0
B 1 0 1 0
C 1 1 0 1
D 0 0 1 0

```

- Danh sách kề:

```

A → B, C
B → A, C
C → A, B, D
D → C

```

Biểu diễn trực quan

- Đồ thị vẽ bằng thư viện (Matplotlib, NetworkX, Graphviz...)
- Highlight cạnh và đỉnh

3. Tài liệu tham khảo chính:

- Nguyễn Thanh Thủy, Đỗ Văn Nhơn, *Cấu trúc dữ liệu & Giải thuật*.
- TS. Phạm Thị Thu Thúy, **Slide Lý thuyết đồ thị & Chương 3** — Biểu diễn đồ thị
- NetworkX Documentation — <https://networkx.org>
- GeeksforGeeks — *Adjacency Matrix vs List*
<https://www.geeksforgeeks.org/graph-and-its-representations/>
- Visualgo Graph Simulator — <https://visualgo.net/graph>

4. Người ra đề: Phạm Thị Thu Thúy - Bộ môn Hệ thống Thông tin

Email : thuthuy@ntu.edu.vn **Phone :** 0901905679