

BÀI TẬP LỚN MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

ĐỀ TÀI 1: CÀI ĐẶT CÔNG CỤ KIỂM TRA CHÍNH TẢ SPELL CHECKER

1. Sơ lược bài toán

Kiểm tra lỗi chính tả là một qui trình kiểm tra một từ có đúng chính tả hay không dựa trên một từ điển có sẵn. Đây là một qui trình được sử dụng trong nhiều ứng dụng như các phần mềm soạn thảo văn bản hay là các từ điển điện tử.

Việc cài đặt một chương trình kiểm tra lỗi chính tả cơ bản thì chỉ dựa thuần túy trên việc kiểm tra trong một từ điển được xây dựng sẵn trên một cấu trúc dữ liệu nào đó. Tuy nhiên có những chương trình còn thực hiện một thao tác là nếu như từ đưa vào không đúng chính tả thì sẽ đưa ra một số các từ đúng chính tả, gợi ý để người dùng có thể nhanh chóng sửa lại lỗi của mình. Việc làm này đòi hỏi thêm một số các thao tác kiểm tra.

2. Yêu cầu

- Yêu cầu cơ bản của bài tập này là lựa chọn một cấu trúc dữ liệu thích hợp để cài đặt một từ điển lưu trữ một số lượng lớn các từ. Khai báo và cài đặt các thuật toán cho phép thêm một từ mới vào từ điển, xóa một từ có sẵn trong từ điển.
- Cài đặt một chương trình cho phép sử dụng một từ điển có sẵn được lưu trữ trong cấu trúc vừa xây dựng và thực hiện công việc kiểm tra xem một từ bất kỳ đưa vào có đúng chính tả hay không.
- Tìm hiểu các giải thuật để thực hiện việc đưa ra một số từ gợi ý sửa lỗi chính tả và cài đặt giải thuật đó.

3. Chú ý

- Ngôn ngữ lập trình : Tùy chọn, ưu tiên sử dụng Java hoặc C++
- Sinh viên cần phải viết các chương trình để kiểm thử kết quả hoạt động của các thao tác đã được cài đặt.
- Sinh viên nên cung cấp một giao diện để người dùng có thể tự nhập dữ liệu đầu vào và kiểm thử các thao tác đã được cài đặt

BÀI TẬP LỚN MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

ĐỀ TÀI 2: CÁC GIẢI THUẬT SẮP XẾP

1. Yêu cầu:

Trong chương trình môn học Cấu trúc dữ liệu và giải thuật, chúng ta có xem xét một số các giải thuật khác nhau để thực hiện việc sắp xếp một dãy các số theo chiều tăng dần hay giảm dần.

Hãy tìm hiểu phương pháp sắp xếp kiểu phân đoạn , sắp xếp kiểu vun đống , sắp xếp kiểu hòa nhập. So sánh các phương pháp sắp xếp trên.

Cài đặt ít nhất 2 trong số các phương pháp sắp xếp đó, thiết lập giao diện đồ họa để minh họa các phương pháp sắp xếp.

Đưa ra một qui trình kiểm thử để có thể đánh giá được các giải thuật đã được cài đặt bằng phương pháp thực nghiệm

Khi cài đặt các phương pháp, chú ý đến việc cho phép hiển thị rõ từng bước thực hiện các giải thuật để thấy được đặc trưng của từng giải thuật.

2. Chú ý

- Ngôn ngữ lập trình : Tùy chọn
- Sinh viên cần phải viết các chương trình để kiểm thử kết quả hoạt động của các thao tác đã được cài đặt.
- Sinh viên nên cung cấp một giao diện để người dùng có thể tự nhập dữ liệu đầu vào và kiểm thử các thao tác đã được cài đặt

BÀI TẬP LỚN MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

ĐỀ TÀI 3: CẤU TRÚC ĐỒ THỊ VÀ BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT

1. Sơ lược bài toán

Hãy xem xét một bài toán thực tế về việc chuyển các gói dữ liệu trên một mạng máy tính. Một mạng máy tính bao gồm nhiều các router được nối với nhau thông qua các dây cáp. Một router có thể là nguồn, là đích hoặc là một trạm trung chuyển các gói dữ liệu trên mạng. Một mạng như vậy có thể được hình dung như là một đồ thị với các router là các đỉnh trong đồ thị và các đường nối giữa các router đóng vai trò như là những cung nối các đỉnh trong đồ thị. Một mạng máy tính tuân thủ theo giao thức OPSF tức là sử dụng đường đi ngắn nhất được xác định sử dụng giải thuật Dijkstra.

Trong phạm vi của bài tập lớn này, hãy hình dung mạng là một đồ thị trọng số, vô hướng. Các cung của đồ thị sẽ được gán một trọng số coi như là thời gian để chuyển một gói dữ liệu bất kỳ giữa hai router được biểu diễn bằng hai đỉnh đầu của cung.

2. Yêu cầu : Các yêu cầu của bài tập lớn này như sau

- Khai báo và cài đặt cấu trúc dữ liệu ĐỒ THỊ cho phép tạo lập và cập nhật một đồ thị trọng số vô hướng. Mỗi đỉnh của đồ thị được đặc trưng bởi một cái tên là một xâu ký tự chứa các chữ cái hoặc chữ số. Mỗi cung của đồ thị được xác lập bằng một cặp đỉnh và một trọng số là số thực.

- Cài đặt các thao tác cơ bản trên đồ thị như sau
 - Khởi tạo một đồ thị ban đầu rỗng
 - InsertEdge(Vertex s, Vertex d, float weight) : bổ sung một cung có trọng số w giữa hai đỉnh s, d
 - InsertVertex(name) : thêm một đỉnh mới vào đồ thị
 - GetWeight(Vertex s, Vertex d): lấy ra trọng số của một cạnh
 - ShortestPath (Vertex s, Vertex d) : Tìm đường đi ngắn nhất từ s đến d trong ngữ cảnh hiện tại của đồ thị (cài đặt giải thuật Dijkstra)
 - Print: hiển thị hiện trạng của đồ thị

Để sử dụng cho ứng dụng tìm đường truyền trong mạng, cài đặt các thao tác sau:

- EdgeDown(Vertex s, Vertex d) : đánh dấu một cạnh nào đó đang được sử dụng, không dùng được
- EdgeUp(Vertex s, Vertex d) Giải phóng việc sử dụng một cạnh
- VertexDown(name) : Đánh dấu một đỉnh đang được sử dụng
- VertexUp(name) : Giải phóng việc sử dụng một đỉnh, đỉnh này trở nên có thể dùng được

3. Chú ý

- Ngôn ngữ lập trình : Tùy chọn, ưu tiên sử dụng Java hoặc C++
- Sinh viên cần phải viết các chương trình để kiểm thử kết quả hoạt động của các thao tác đã được cài đặt.
- Sinh viên nên cung cấp một giao diện để người dùng có thể tự nhập dữ liệu đầu vào và kiểm thử các thao tác đã được cài đặt

BÀI TẬP LỚN MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

ĐỀ TÀI 4: CẤU TRÚC ĐỒ THỊ VÀ BÀI TOÁN TÌM CÂY KHUNG CỰC TIỂU

1. Sơ lược bài toán

Bài toán tìm cây khung cực tiểu trên cấu trúc đồ thị là một bài toán được áp dụng nhiều trong thực tế.

Trong ngữ cảnh của cấu trúc đồ thị, một cây T là một đồ thị trong đó bất kỳ hai đỉnh trong đồ thị này đều được nối với nhau bằng một đường đi duy nhất.

Cho một đồ thị G , một cây khung T_1 trong G là một phần con của G mà nó là một cây và nó chứa tất cả các đỉnh trong đồ thị G . Một đồ thị có thể có nhiều cây khung. Nếu đồ thị đã cho ban đầu là một đồ thị trọng số thì chúng ta có thể tính trọng số của một cây khung bằng tổng tất cả các trọng số của các cạnh trong cây.

Một cây khung có trọng số nhỏ nhất được gọi là cây khung cực tiểu của đồ thị

2. Yêu cầu : Các yêu cầu của bài tập lớn này như sau

- Khai báo và cài đặt cấu trúc dữ liệu ĐỒ THỊ cho phép tạo lập và cập nhật một đồ thị trọng số vô hướng. Mỗi đỉnh của đồ thị được đặc trưng số tự nhiên có giá trị duy nhất trong đồ thị. Mỗi cung của đồ thị được xác lập bằng một cặp đỉnh và một trọng số là số thực.
- Cài đặt các thao tác cơ bản trên đồ thị như sau
 - Khởi tạo một đồ thị ban đầu rỗng
 - `InsertEdge(Vertex s, Vertex d, float weight)` : bổ sung một cung có trọng số w giữa hai đỉnh s, d
 - `InsertVertex(name)` : thêm một đỉnh mới vào đồ thị
 - `GetWeight(Vertex s, Vertex d)` : lấy ra trọng số của một cạnh
 - `Print` : hiển thị hiện trạng của đồ thị
- Tìm hiểu về hai thuật toán Kruskal và Prim để tìm một cây khung cực tiểu trong một đồ thị trọng số cho trước, so sánh hai giải thuật này về mặt hiệu năng
- Cài đặt hai giải thuật đã tìm hiểu
- Xây dựng một chương trình nhận đầu vào là một file text chứa thông tin về một đồ thị cho trước, tạo lập đồ thị và chỉ rõ từng bước hoạt động của giải thuật tìm cây khung cực tiểu đã cài đặt trên đồ thị tạo lập được

3. Chú ý

- Ngôn ngữ lập trình : Tùy chọn, ưu tiên sử dụng Java hoặc C++
- Sinh viên cần phải viết các chương trình để kiểm thử kết quả hoạt động của các thao tác đã được cài đặt.
- Sinh viên nên cung cấp một giao diện để người dùng có thể tự nhập dữ liệu đầu vào và kiểm thử các thao tác đã được cài đặt

BÀI TẬP LỚN MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

ĐỀ TÀI 5: CẤU TRÚC CÂY NHỊ PHÂN TÌM KIẾM CÂN BẰNG

1. Sơ lược bài toán

- Cây nhị phân tìm kiếm là một cấu trúc đặc biệt, được sử dụng trong bài toán tìm kiếm. Mô tả về cây nhị phân tìm kiếm có thể tham khảo trong sách giáo trình của môn học. Cây AVL là một dạng đặc biệt của cây nhị phân tìm kiếm, mô tả về cây AVL cũng có trong sách giáo trình

2. Yêu cầu

Hãy khai báo một lớp đối tượng dạng cây nhị phân tìm kiếm có tên BST trong đó chấp nhận các nút của cây có thể nhận giá trị là các số nguyên sử dụng kiểu lưu trữ móc nối của cây. Cài đặt các thao tác sau trên cây nhị phân tìm kiếm:

- Create(): Khởi tạo một cây rỗng, thủ tục này sẽ trả ra một con trỏ tới gốc của cây, con trỏ ban đầu sẽ có giá trị không xác định
- Contain(T,x) : Thực hiện việc tìm kiếm, xác định xem trong cây đã có nút nào có giá trị x cho trước hay không? Nếu có, hãy chỉ ra chỉ số của nút chứa giá trị đó
- Insert(T, x) : Bổ sung một nút mới có giá trị là x cho trước vào một cây T. Trong quá trình bổ sung có thực hiện tìm kiếm, nếu giá trị x đã có trong cây, không cần tiến hành bổ sung
- Delete(T,x): Loại bỏ một nút có giá trị x cho trước ra khỏi cây T
- Print(T) : Thủ tục này cho phép hiển thị một cây T lên màn hình
- IsAVL(T) : Kiểm tra xem một cây T có phải là một cây AVL không?

Phân mở rộng: Mở rộng thủ tục Insert để có thể đáp ứng được yêu cầu như đối với cây AVL (thực hiện các thao tác quay để đảm bảo tính cân bằng của cây)

3. Chú ý

- Ngôn ngữ lập trình : Tùy chọn, ưu tiên sử dụng C, Java hoặc C++
- Sinh viên cần phải viết các chương trình để kiểm thử kết quả hoạt động của các thao tác đã được cài đặt.
- Sinh viên nên cung cấp một giao diện để người dùng có thể tự nhập dữ liệu đầu vào và kiểm thử các thao tác đã được cài đặt

BÀI TẬP LỚN MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

ĐỀ TÀI 6: Các giải thuật cho bài toán đối sánh mẫu

1. Sơ lược bài toán

Đối sánh mẫu (Pattern matching) là một bài toán tìm sự xuất hiện của một xâu ký tự mẫu (pattern) trong một văn bản. Bài toán này có thể được áp dụng trong một số lĩnh vực như tìm kiếm thông tin, xử lý văn bản , các chương trình soạn thảo văn bản.

2. Yêu cầu

Tìm hiểu 3 giải thuật cho bài toán đối sánh mẫu

- Giải thuật Naïve
- Giải thuật Knutt-Morris-Pratt (KMP)
- Giải thuật Boyer-Moore (BM)

So sánh 3 giải thuật

Cài đặt chương trình demo ít nhất 2 trong số 3 giải thuật trên, Chương trình demo cần cho phép người sử dụng nhập dữ liệu đầu vào một cách linh hoạt và cho người sử dụng nhìn thấy từng bước xử lý của giải thuật trong quá trình xác định vị trí xuất hiện của mẫu trong văn bản đầu vào.