**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**ĐỀ TÀI: TÁN GÁI ĐẠI CƯƠNG**

**SINH VIÊN THỰC HIỆN:**

**Nguyễn Vũ Khánh Uy LỚP: 21TCLC\_DT3 NHÓM: 21.14A**

**Nguyễn Nhật Minh LỚP: 21TCLC\_DT4 NHÓM: 21.14A**

**GIẢNG VIÊN HƯỚNG DẪN: Nguyễn Thị Minh Hỷ**

**Đà Nẵng 12-2021**

LỜI MỞ ĐẦU

Hiện nay, tình trạng mất căn bằng giới tính đang diễn ra đáng báo động và dẫn đến việc thiếu hụt phụ nữ và ‘dư thừa’ đàn ông[1]. Điều đó sẽ dẫn đến sự cạnh tranh lớn trong việc tìm kiếm bạn đời của người đàn ông, dĩ nhiên nếu người đàn ông thành công, tốt tính hơn tìm được người phụ nữ thì không có gì để nói. Nhưng các cụ có câu: ‘Phụ nữ yêu bằng tai’, thật vậy phụ nữ thường lại thích những anh chàng dẽo miệng, có nhiều kinh nghiệm cưa cẩm hơn là những anh chàng hiền lành, ổn định. Nếu lướt trên những diễn đàn mạng xã hội như sinh viên đà nẵng confession.. ta thấy không thiếu những bài viết nói trai Bách Khoa nói chuyện nhạt nhẽo, trai IT chỉ biết hỏi ‘em ăn cơm chưa’, trai Bách Khoa chỉ biết chạy deadline, không sành điệu.. Nhưng chắc hẳn thầy cô dạy Bách Khoa cũng biết sinh viên Bách Khoa những người nhạt nhẽo, không quá sành điệu vậy thường là những người rất tốt tình và ra xã hội thường thành công. Những người như vậy nếu chỉ vì tập trung sự nghiệp, học tập mà không tìm được bạn đời thì sẽ tạo nên trào lưu xấu ngược lại không quan tâm tới sự nghiệp mà lo tìm vợ trước, như vậy thì làm sao Việt Nam ta có thể phát triển, sánh ngang với các cường quốc năm châu như lời Bác Hồ đã dạy. Vì vậy, sự ra đời của ứng dụng hướng dẫn tán gái là vô cùng cần thiết.

Thực ra cũng đã có những sản phẩm đi trước, trên thị trường cũng có một số ứng dụng hướng dẫn tán gái, ngoài vấn đề chung là việc tán gái không đơn giản chỉ đọc một lý thuyết là có thể làm được thì những app trên đa số đều dạy tán gái bằng hình thức đưa ra bài học và hiệu quả của việc đọc đơn thuần là không cao, dễ khiến người dùng bị cuốn theo việc đọc quá nhiều lý thuyết mà không thực hành. Vì vậy nhóm em đã cho ra đời ứng dụng **‘Tán gái đại cương’**, không chỉ có những bài viết để hướng dẫn những lý thuyết tán gái cơ bản mà khi những bạn trai thực hành tán gái thử ngoài đời nếu con gái nói những câu nói khó hiểu(và điều này thường diễn ra rất thường xuyên), các bạn nam có thể sử dụng một tính năng đột phá của ứng dụng so với sản phẩm khác là từ điển tiếng em, chứa một kho từ điển lớn về các câu nói khó hiểu của phụ nữ và có thể kèm theo cách xử lí tình huống. Ngoài ra kĩ thuật tán gái luôn tiến triển theo thời gian, ứng dụng còn cho phép người dùng ghi chú lại những kĩ thuật mới vừa phát hiện hoặc những sai lầm mắc phải trên con đường tán gái.

# MỤC LỤC

[1](#_Toc122781857)

[LỜI MỞ ĐẦU 2](#_Toc122781858)

[MỤC LỤC 3](#_Toc122781859)

[DANH MỤC HÌNH VẼ 4](#_Toc122781860)

[1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc122781861)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 5](#_Toc122781862)

[2.1. Đọc lý thuyết tán gái 5](#_Toc122781863)

[2.2. Từ điển tiếng em 5](#_Toc122781864)

[2.3. Thêm ghi chú 5](#_Toc122781865)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU 6](#_Toc122781866)

[3.1. Phát biểu bài toán 6](#_Toc122781867)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống 7](#_Toc122781868)

[3.2.1. Cách giải bài toán đơn giản 7](#_Toc122781869)

[3.2.2. Cải tiến thuật toán bằng cấu trúc dữ liệu Trie 7](#_Toc122781870)

[3.2.3. Cải tiến bộ nhớ cho Trie bằng cây đỏ đen 8](#_Toc122781871)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 9](#_Toc122781872)

[4.1. Cấu trúc hệ thống hướng đối tượng 9](#_Toc122781873)

[4.2. Kết quả 9](#_Toc122781874)

[4.2.1. Giao diện chính của chương trình 9](#_Toc122781875)

[4.2.2. Kết quả thực thi của chương trình 9](#_Toc122781876)

[4.2.3. Nhận xét 9](#_Toc122781877)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 9](#_Toc122781878)

[a. Kết luận 9](#_Toc122781879)

[b. Hướng phát triển 9](#_Toc122781880)

[TÀI LIỆU THAM KHẢO 10](#_Toc122781881)

DANH MỤC HÌNH VẼ

# GIỚI THIỆU ĐỀ TÀI

Đề tài là một ứng dựng hướng dẫn tán gái thông qua các bài học (có thể đi kèm video) và từ điển với giao diện tương tác trực quan dễ dùng. Ứng dụng chạy được trên hệ điều hành window và chạy tốt trên máy có ram >= 2GB và ổ cứng còn trống >= 1GB. Ứng dụng được viết bằng C# và Windows Forms.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Đọc lý thuyết tán gái

Khi người dùng cần đọc lý thuyết tán gái, họ có thể vào tính năng này. Ứng dụng sẽ hiển thị các lý thuyết tán gái theo tiêu đề, người dùng có thể chọn một chủ đề muốn học để học, mỗi bài học là một bài viết và có thể có kèm video hướng dẫn ở cuối bài. Các bài học tán gái được tổng hợp từ nhiều nguồn sách uy tín từ Âu sang Á trong bộ môn tán gái như: The PlayBook (Barney Stinson)[2], Alpha Male VietNam(Cộng đồng PUA VN)[3]..

A person's face on a book cover

Description automatically generated with low confidenceText

Description automatically generated

## Từ điển tiếng em

Khi người dùng cần tra một cụm từ, câu nói khó hiểu của phụ nữ, họ có thể vào tính năng này. Ứng dụng sẽ hiện ra một thanh tìm kiếm và người dùng gõ vào nó sẽ gợi ý các từ(câu) bắt đầu bằng từ người dùng nhập, khi người dùng nhìn thấy từ(câu) tương tự điều họ muốn tìm kiếm, người dùng có thể chọn xem ý nghĩa và cách xử lí khi gặp trường hợp đó dưới dạng bài viết. Các từ(câu) được tổng hợp từ bộ từ điển tiếng em[4] và nhiều từ do tác giả tự thu thập..

## Thêm ghi chú

Khi người dùng cần thêm một ghi chú, họ có thể vào tính năng này. Ứng dụng sẽ hiển thị một màn hình cho người dùng ghi chú gồm 3 phần cơ bản là tiêu đề, nội dung, loại ghi chú. Nếu ghi chú là lý thuyết tán gái thì người dùng có thể chọn một file video trong máy để lưu kèm ghi chú, nếu là từ điển tiếng em thì người dùng có thể thêm cách xử lí. Ở đây ứng dụng có tính năng thêm mà không có tính năng xóa hay sửa, đối với mặc lập trình thì chỉ cần thêm vài chục dòng lệnh nhưng tác giả muốn dạy cho người dùng bài học rằng mọi thứ đã làm thì hậu quả để lại mãi mãi, đó là một trong những bài học quan trọng nhất trong bộ môn tán gái đại cương.

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phát biểu bài toán

Bài toán chính ở đây là tìm kiếm từ(câu) phù hợp trong từ điển tiếng em, đây là một bài toán phức tạp vì mỗi khi người dùng nhập 1 kí tự ta phải trả lời lại các từ(câu) phù hợp trong tập dữ liệu có sẳn rất lớn đủ nhanh (mỗi giây trung bình người dùng có thể gõ được 2 3 kí tự). Việc tìm kiếm phải chính xác, không thể sử dụng các thuật toán mang tính xác xuất, nó cũng phải tương đối tiết kiệm bộ nhớ để chạy được trên hầu hết các máy tính cá nhân hiện nay.

Input: Cho một chuỗi độ dài L (đây là từ người dùng cần tìm kiếm trong từ điển tiếng em) và một tập s gồm N chuỗi (đây là tập các từ đã được giải thích nghĩa đã được lưu trong chương trình).

Output: Hãy tìm ra nhiều nhất 10 chuỗi con bắt đầu bằng L kí tự đã cho trong tập s (các chuỗi in ra là chuỗi gợi ý từ người dùng đang tìm kiếm, nếu người dùng muốn hiển thị hết thì có một nút giúp làm điều đó nhưng bài toán đó không đáng để đề cập vì nó đơn giản là duyệt hết các chuỗi phù hợp trong tập s).

Giới hạn:

* 0 < L < 66 (người dùng thường sẽ không nhập nhiều và thực ra giới hạn của các từ câu được hỗ trợ là 66 kí tự)
* 0 < N < 1000000 (1 triệu tuy là một con số khổng lồ trong dữ liệu nhưng nếu so với độ phức tạp của các câu nói phụ nữ thì nó chưa chắc đã chứa đủ nhiều)
* Độ dài của một chuỗi con trong tập dữ liệu không quá 66 kí tự.

Yêu cầu:

* Thời gian chạy: < 0.3s (vì như đã đề cập ở trên mỗi giây người dùng có thể nhập tầm 3 kí tự mới nên ta có thể phải giải bài toán này 3 lần trong một giây)
* Bộ nhớ sử dụng tối đa: 2GB (vì muốn hướng tới việc chạy được trên mọi máy cá nhân hiện nay nên ứng dụng không được sử dụng quá nhiều tài nguyên)

Ví dụ:

N = 5, s = {Em chỉ xem anh là bạn, Em yêu anh, Em cảm thấy mình không còn hợp nhau, em đang bận, mình chia tay đi}, chuỗi người dùng nhập: Em => { Em chỉ xem anh là bạn , Em yêu anh, , Em cảm thấy mình không còn hợp nhau }

## Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống

Nhắc lại một chút về kiến thức cấu trúc máy tính vi xử lí và nguyên lí hệ điều hành: Trong 1s máy tính hiện đại có thể tính toán khoảng 10^8 phép tính, vì vậy để đảm bảo yêu cầu của bài toán đặt ra số lương phép tính của **thuật toán của chúng ta chỉ được dùng khoảng 10^7 phép tính** (vì việc hiển thị dữ liệu lên winform cũng tốn thời gian xử lí đồ họa, giao tiếp giữa các tiến trình..). Bộ nhớ cho 2GB mà để lưu trữ ta đã phải sử dụng 16 \* 10^6 \* 66 bit ~ 1GB để lưu tập dữ liệu, vậy **thuật toán của ta phải có độ phức tạp bộ nhớ thêm tuyến tính** để không sinh thêm quá nhiều bộ nhớ. Một vấn đề nhỏ là nếu mỗi lần người dùng sử dụng chức năng từ điển ta phải nạp lại dữ liệu thì sẽ gây ra hiện tượng giật lag đáng kể vì phải load lại tới 1GB dữ liệu, vì vậy ta lợi dụng một tính chất của các ứng dụng winform là các form sau form đầu tiên được gọi ra chỉ là tiến trình con của form đó và ta có thể nạp dữ liệu lên form chính trước sau đó mỗi lần gọi form từ điển ta chỉ cần trao đổi dữ liệu với form chính.

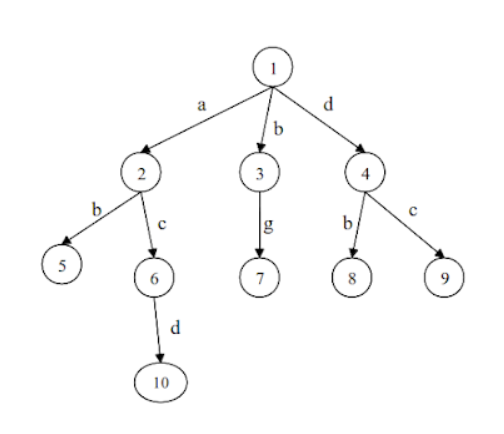
### Cách giải bài toán đơn giản

Để giải bài toán trên cách đơn giản nhất là ta duyệt qua hết tập dữ liệu xem từ nào có phần đầu giống với phần người dùng đang gõ thì ta hiển thị, đến khi đủ 10 từ hoặc quét hết tập dữ liệu thì dừng, dễ dàng thấy được độ phức tạp trường hợp xấu nhất của thuật toán trên là O(N\*L) vì phải duyệt qua N chuỗi trong tập dữ liệu và với mỗi chuỗi phải kiểm tra L kí tự đầu và có độ phức tạp bộ nhớ thêm là O(1) vì nó tính trực tiếp trên dữ liệu có sẵn. Với giới hạn của bài toán ta có thể thấy thuật toán chạy quá giới hạn thời gian vì nó có thể phải tính 10^6 \* 66 phép tính > 10^7, đương nhiên thời gian chạy thực tế của thuật toán này sẽ nhanh hơn như vậy vì ta có thể tìm được 10 từ trong vị trí sớm hơn và khi một từ sai thì có thể nó sai ở vài kí tự đầu, dù vậy đó chỉ là độ phức tạp của việc xử lí còn hiển thị ra giao diện trên ứng dụng winform cũng sẽ tốn thêm một phần thời gian nữa. Nói chung đây là thuật toán không đạt yêu cầu vì thời gian chạy không ổn định và có thể cực kì chậm đối với tập dữ liệu và truy vấn của người dùng xấu.

### Cải tiến thuật toán bằng cấu trúc dữ liệu Trie

Để cải tiến thuật toán ta có thể sử dụng cấu trúc dữ liệu Trie(cây tiền tố) để lưu và quản lí các chuỗi. Trie cho phép thêm, xóa, tìm một chuỗi trong nó với độ phức tạp là O(L). Đây là cấu trúc dữ liệu không mới trong các bài toán tìm kiếm chuỗi từ trong tập dữ liệu lớn.

Trie gồm một gốc không chứa thông tin, trên mỗi cạnh lưu một ký tự, mỗi nút và đường đi từ gốc đến nút đó thể hiện 1 xâu, gồm các ký tự là các ký tự thuộc cạnh trên đường đi đó.



Trong hình vẽ trên, nút 1 là nút gốc, nút 7 thể hiện có 1 xâu là ‘bg’, nút 8 thể hiện có 1 xâu là ‘db’, nút 9 thể hiện có 1 xâu là ‘dc’, nút 10 thể hiện có 1 xâu là ‘acd’, nút 5 thể hiện là có 1 xâu là ‘ab’.

Tuy nhiên, như các bạn có thể thấy, đối với một số nút, chẳng hạn nút 4, ta không biết nó là thể hiện kết thúc 1 xâu hay chỉ là 1 phần của đường đi từ nút 1 đến nút 9. Vì vậy, khi cài đặt, thông thường, tại nút U ta cần lưu thêm thông tin nút U có là kết thúc của 1 xâu hay không, hoặc nút U là kết thúc của bao nhiêu xâu, tuỳ theo yêu cầu bài toán.

Vậy nếu ứng dụng nó vào bài toán của ta độ phức tạp thời gian chạy sẽ là O(L), thừa đủ thời gian để vượt qua yêu cầu của đề. Nhưng có một vấn đề nghiêm trọng, đó là độ phức tạp bộ nhớ. Vì chương trình cho phép mỗi kí tự thuộc bảng mã utf-16, vậy với mỗi nút con trong trie sẽ phải có tới 2^16 cạnh nối tới nút con khác => Bộ nhớ thực tế để sử dụng lưu trie sẽ cực kì khủng khiếp và không thể lưu nổi trên máy tính cá nhân.

### Cải tiến bộ nhớ cho Trie bằng cây đỏ đen

Nhận xét việc sử dụng Trie với mỗi nút ta lưu 2^16 cạnh nối tới các nút con là không cần thiết bởi chỉ có một số lượng nhỏ các nút con thực sự có ý nghĩa, vậy ta có thể cải tiến bằng việc chỉ lưu các cạnh nối có ý nghĩa. Vậy bài toán giờ trở thành lưu các cạnh nối có ý nghĩa sao cho tiết kiệm bộ nhớ và thời gian tìm kiếm lại một cạnh bất kì phải đủ nhanh. VD nếu giờ mỗi nút ta lưu một danh sách liên kết các cạnh nối có ý nghĩa thì sẽ rất tiết kiệm bộ nhớ nhưng đổi lại việc tìm kiếm mộ cạnh bất kì sẽ rất lâu và ảnh hưởng lớn đến độ phức tạp tìm kiếm của Trie. Giải pháp ở đây là lưu một cây tìm kiếm nhị phân chứa các cạnh có ý nghĩa, việc thêm hoặc xóa hoặc tìm một cạnh trong cây tìm kiếm nhị phân giờ chỉ tốn O(log(n)) với n là số lượng cạnh có ý nghĩa trong nút đang xét. Và đương nhiên để cây không suy biến thì ta sẽ lưu các cạnh vào một cây tìm kiếm nhị phân tự cân bằng (AVL Tree), cụ thể ở đây là cây đỏ đen(RBT). Vậy mỗi nút của Trie bây giờ sẽ chứa một cây đỏ đen hay cấu trúc dữ liệu của chúng ta sử dụng bây giờ sẽ là một cây tiền tố chứa các cây đỏ đen. Vậy độ phức tạp thời gian chạy của cấu trúc dữ liệu mới của chúng ta là O(L\*log(n)) với n (n < 2^16) là số cạnh có ý nghĩa của một nút trong Trie và độ phức tạp bộ nhớ là O(N\*L) vì phải lưu lại cả bộ dữ liệu của bài toán dưới dạng Trie và RBT. Dễ dàng thấy độ phức tạp của thuật toán trên của ta hoàn toàn đạt yêu cầu của bài toán và chạy ổn định trong mọi trường hợp.

Diagram

Description automatically generated

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## Cấu trúc hệ thống hướng đối tượng

## Kết quả

### Giao diện chính của chương trình

### Kết quả thực thi của chương trình

Mô tả kết quả thực hiện chương trình.

### Nhận xét

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

## Hướng phát triển

TÀI LIỆU THAM KHẢO

[1] Tên tác giả, Tên tài liệu, Tên nhà xuất bản, năm xuất bản

[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập