**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ CẦN THƠ**

KHOA CÔNG NGHỆ THÔNG TIN

**CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH**

**★★★🕮★★★**

**A white circle with blue text and a book and a symbol

Description automatically generated**

ĐỀ TÀI ĐỒ ÁN:

**TÌM HIỂU VỀ GA VÀ ỨNG DỤNG**

***Cần Thơ, tháng 12, năm 2023***

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ CẦN THƠ**

KHOA CÔNG NGHỆ THÔNG TIN

**CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH**

**★★★🕮★★★**

**A white circle with blue text and a book and a symbol

Description automatically generated**

ĐỀ TÀI ĐỒ ÁN:

**TÌM HIỂU VỀ GA VÀ ỨNG DỤNG**

Giảng viên hướng dẫn: **LÊ ANH NHÃ UYÊN**

Sinh viên thực hiện: **LÊ PHƯỚC HỮU 2100450**

**NGUYỄN MỸ HẰNG**

***Cần Thơ, tháng 12, năm 2023***

**PHIẾU CHẤM ĐIỂM**

**LỜI CAM ĐOAN**

**LỜI CẢM ƠN**

Trong lần đầu tiên chúng em thực hiện đồ án 1 của học kỳ một năm thứ ba tại Trường Đại Học Kỹ Thuật Công Nghệ Cần Thơ, thật may mắn khi được giảng viên là cô Lê Anh Nhã Uyên đồng thời cũng là cố vấn học tập của chúng em đã đồng hành, hướng dẫn chúng em trong quá trình thực hiện đồ án 1 này. Với sự chân thành, nhiệt quyết của Cô chúng em muốn gửi lời cảm ơn chân thành đến cô Lê Anh Nhã Uyên vì đã cùng chúng em hoàn thành đồ án 1 này. Xin cảm ơn Cô!

**MỞ ĐẦU**

1. **Lý do chọn đề tài**

Lĩnh vực Công Nghệ Thông Tin (Information Technology) hiện nay đóng một vai trò không thể thay thế đối với sự phát triển của mọi thành phần ở một quốc gia cụ thể là về các lĩnh vực như: Tài chính, giáo dục, giao thông vận tải, cơ sở hạ tầng, giải trí, thể thao,... có thể thấy rằng lĩnh vực này dường như đã và đang không thể thiếu đối với sự phát của mọi lĩnh vực khác có mặt trong xã hội. Những vấn đề nan giải, những công việc tưởng chừng khó có thể hoàn thành một cách nhanh chóng, những bài toán thực tế chưa có lời giải tối ưu thì hiện nay đã có giải pháp, đã có phương pháp để giải quyết vấn đề một cách gọn gàng, những công việc phải sử dụng tay chân quá nhiều không cần thiết, nghiệp vụ rườm rà thiếu hiệu quả, mất nhiều thời gian thì nay đã được khắc phục một cách triệt để hơn rất nhiều so với trước kia, nhờ vào việc đã ứng ứng dụng thực tiễn các phần mềm, công cụ, công nghệ vào trong quá trình xử lý công việc, học tập, giảng dạy, giúp cho những công việc này được tiến hành hiệu quả hơn, tiết kiệm được một số chi phí dư thừa, từ đó cải thiện được các khía cạnh khác của một cá nhân của một tổ chức và khi cá nhân, tổ chức phát triển thì toàn xã hội đều được phát triển. Nhắc đến lĩnh vực Công Nghệ Thông Tin thì chúng ta không thể không nhắc đến một lĩnh vực chuyên sâu hơn đó là Trí Tuệ Nhân Tạo.

Lĩnh vực Trí Tuệ Nhân Tạo (Artificial Intelligence) được biết đến là một nhánh của chuyên ngành Khoa Học Máy Tính (Computer Science) và chuyên ngành này là một phần của lĩnh vực Công Nghệ Thông Tin. Ta có thể thấy Trí Tuệ Nhân Tạo như là bộ não của một phần mềm, công cụ, thiết bị,... Trí Tuệ Nhân Tạo sẽ quyết định tính hiệu quả của phần mềm, công cụ, thiết bị đó ở mức nào. Nhắc đến Trí Tuệ Nhân Tạo thì không thể không nhắc đến Máy Học (Machine Learning) và Học Sâu (Deep Learning), hai thành phần này đều nằm bên trong Trí Tuệ Nhân Tạo và là nền tảng giúp cho Trí Tuệ Nhân Tạo phát triển. Chung quy lại những lĩnh vực chuyên sâu được nêu trên đều cần đến Giải Thuật và chỉ có Giải Thuật mới giúp cải thiện được khả năng giao tiếp giữa Máy và Người và cải thiện được tính hiệu quả khi giải quyết các bài toán thực tế thông qua việc ứng dụng giải thuật vào trong Artificial Intelligence, Machine Learning, Deep Learning.

Trong quá trình học tập và nghiên cứu về chuyên ngành khoa học máy tính, nhận thấy rằng những phương pháp, những kỹ thuật được ứng dụng để tìm kiếm giải pháp cho các bài toán khó, làm sao có được một kết quả tối ưu nhất. Có rất nhiều phương pháp, kỹ thuật được ứng dụng thì một trong số những phương pháp, kỹ thuật được biết đến là Giải Thuật Di Truyền (Genetic Algorithm) kỹ thuật này được dựa trên nguyên lý sinh học tiến hóa để tìm ra một kết quả tối ưu nhất và khi tìm hiểu sơ bộ về Giải Thuật Di Truyền thì chúng em thấy được sự hứng thú nhất định về giải thuật này, thấy được những ứng dụng hữu ích mà kỹ thuật này mang lại trong việc đóng góp sự phát triển các giải thuật để tìm ra lời giải cho các bài toán nan giải.

Và trong lần đầu tiên thực hiện đề án 1 này, chúng em sẽ cùng nhau nghiên cứu, tìm hiểu, thực hành ứng dụng một giải thuật tối ưu kết quả cho bài toán đó là Giải Thuật Di Truyền (Genetic Algorithm).

1. **Đối tượng nghiên cứu**

Đối tượng nghiên cứu trong đề tài này sẽ tập trung vào các nội dung sau:

* Giải Thuật Di Truyền, các phương pháp biễu diễn bài toán, mã hóa nhiễm sắc thể và các toán tử trong Giải Thuật Di Truyền.
* Ứng dụng Giải Thuật Di Truyền vào bài toán.

1. **Mục tiêu**

Với đề tài là Giải Thuật Di Truyền này mục tiêu của chúng em đặt ra là:

* Giải thích được Giải Thuật Di Truyền.
* Nêu được các thành phần trong Giải Thuật Di Truyền.
* Trình bày được các phương pháp, cơ chế thực hiện và nguyên lý hoạt động của Giải Thuật Di Truyền.
* Và ứng dụng được Giải Thuật Di Truyền vào bài toán thực tế.

**GIỚI THIỆU**

Trong đề tài Giải Thuật Di Truyền của đồ án 1 này chúng em trình bày các nội dung như sau:

* **Chương 1. GIỚI THIỆU TỔNG QUAN**
* **Chương 2. GIẢI THUẬT DI TRUYỀN**
* **Chương 3. ỨNG DỤNG CỦA GIẢI THUẬT DI TRUYỀN VÀO CÁC BÀI TOÁN**
* **Chương 4. THỰC HÀNH ỨNG DỤNG GIẢI THUẬT DI TRUYỀN THỰC TẾ**

**Chương 1**: Nêu các khái niệm liên quan đến Giải Thuật Di Truyền như: Khoa Học Máy Tính là gì?, lịch sử Khoa Học Máy Tính, vai trò của Khoa Học Máy Tính đối với sự phát triển đa ngành nghề. Trình bày thêm về Giải Thuật, lý do phải sử dụng đến giải thuật và một số giải thuật tối ưu kết quả bài toán.

**Chương 2**: Ở chương này chúng em sẽ trình bày thuyết di truyền của Charles Darwin, các thành phần cấu thành quá trình di truyền và vẽ sơ đồ về quá trình di truyền xảy ra trong tự nhiên. Tiếp theo nghiên cứu sâu về Giải Thuật Di Truyền, lịch sử của Giải Thuật Di Truyền, vẽ sơ đồ tiến trình, tiếp theo trình bày về cơ chế thực hiện và nguyên lý hoạt động của Giải Thuật Di Truyền. Nêu ra các phương pháp biểu diễn bài toán của Giải Thuật Di Truyền và những toán tử cơ bản trong Giải Thuật Di Truyền.

**Chương 3**:

**Chương 4**:

**MỤC LỤC**

**DANH MỤC THUẬT NGỮ TIẾNG ANH**

**DANH MỤC HÌNH**

**Chương 1**

**GIỚI THIỆU TỔNG QUAN**

1. **KHOA HỌC MÁY TÍNH**
2. **Khoa Học Máy Tính là gì?**

*Khoa Học Máy Tính* là chuyên ngành nghiên cứu về các cơ sở lý thuyết về các dữ liệu thông tin và tính toán, cùng với sự thực hiện và ứng dụng của chuyên ngành này trong các hệ thống máy tính. *Khoa Học Máy Tính* là cách tiếp cận khoa học và thực tiễn nhất để có thể tính toán nghiên cứu tìm ra các giải pháp thích hợp, tối ưu nhất cho các vấn đề nan giải, việc thua thập dữ liệu, phân tích, xử lý, lưu trữ, truyền thông và truy cập dữ liệu tất cả đều là một phần trong *Khoa Học Máy Tính*.

Một định nghĩa khác về chuyên ngành này, *Khoa Học Máy Tính* nghiên cứu sâu về các quy trình, cách thức để thực hiện của các giải thuật từ đó trên nền tảng của các giải thuật đã nghiên cứu, kế tiếp là áp dụng vào các bài toán thực tế chưa có lời giải tối ưu nhất.

Có rất nhiều bài toán tối ưu chưa thật sự có một giải thuật tối ưu thõa đáng, như thế dẫn đến việc một chương trình, hệ thống giảm đi tính hiệu quả, tốn nhiều tài nguyên và thời gian. Công việc của một kỹ sư *Khoa Học Máy Tính* là nghiên cứu, thấu hiểu vấn đề, chọn lọc các giải pháp, tích hợp các dữ liệu đã được thu thập, từ đó đưa ra các giải thuật tối ưu thõa đáng cho một bài toán.

1. **Lịch sử về Khoa Học Máy Tính**

Lịch sử của Khoa Học Máy Tính đã bắt đầu từ nhiều năm trước khi các kỹ sư phát minh ra máy tính hiện đại. Những máy tính dành cho việc tính toán các bài toán số học đã tồn tại từ thời cổ đại, chẳng hạn như bàn tính. Hơn nữa, các giải thuật để thực hiện tính toán đã có kể từ thời cổ đại, ngay cả trước khi phát triển các thiết bị tính toán phức tạp.

*Wilhelm Schickard* đã thiết kế và hoàn thành chiếc máy tính cơ học đầu tiên năm 1623. Đến năm 1673, *Gottfried Leibniz* trình diễn một máy tính cơ học số, được gọi là *Stepped Reckoner*. Ông có thể được coi là nhà *Khoa Học Máy Tính* đầu tiên và nhà lý thuyết thông tin mà đã ghi lại hệ thống số nhị phân.

*Blaise Pascal* thiết kế và xây dựng máy tính cơ học hoạt động được mang tên *Pascaline.* Năm 1642 *Charles Babbage* đã thiết kế một máy tính theo hiệu (*Difference Engine*) vào thời *Victoria* và *Ada Lovelace* đã viết bản hướng dẫn sử dụng máy. Nhờ công trình này, ngày nay bà được biết đến là lập trình viên đầu tiên trên thế giới. Vào khoảng năm 1900, tập đoàn *IBM* đã bán những chiếc máy tính dùng thẻ đục lỗ. Tuy nhiên, tất cả những chiếc máy này đều chỉ thực hiện một nhiệm vụ đơn hoặc cùng lắm là một tập nhỏ các nhiệm vụ.

Trước năm 1920, công việc tính toán được thực hiện chủ yếu bởi những nhân viên chuyên nghiệp. Những nhà nghiên cứu đầu tiên về ngành mà sau này được gọi là *Khoa Học Máy Tính*, chẳng hạn như *Kurt Gödel*, *Alonzo Church* và *Alan* *Turing*, đã quan tâm đến câu hỏi về khả năng tính toán: “*những gì có thể được tính toán bởi một người thủ quỹ người chỉ đơn giản dùng giấy và bút chì để làm một danh sách các bước tính toán, cho đến khi nào xong việc mà không cần đến trí thông minh hay hiểu biết?*”. Một phần của động cơ này là ước muốn phát triển các máy tính có khả năng tự động hóa các công việc tính toán thường là buồn tẻ và dễ sai của một người tính toán. Vấn đề then chốt là xây dựng các hệ thống tính toán phổ dụng có khả năng thực hiện mọi nhiệm vụ tính toán có thể cần đến và nhờ đó tổng quát hóa tất cả các máy tính chuyên biệt trước kia thành một khái niệm đơn nhất về chiếc máy tính phổ dụng.

Trong những năm 1940, khi các máy tính mới hơn và mạnh hơn được phát triển, người ta thấy rõ ràng hơn rằng máy tính có thể được sử dụng cho nhiều mục đích khác ngoài các tính toán toán học, lĩnh vực *Khoa Học Máy Tính* được mở rộng thành ngành nghiên cứu tính toán nói chung. Từ thập kỷ 60, *Khoa Học Máy Tính* bắt đầu được thiết lập như là một ngành học riêng biệt, với sự ra đời của các khoa *Khoa Học Máy Tính* đầu tiên và các chương trình đào tạo đại học chuyên ngành *Khoa Học Máy Tính*. Từ khi các máy tính được sử dụng trong thực tiễn, nhiều ứng dụng của tính toán đã trở thành các lĩnh vực nghiên cứu riêng biệt.

1. **Tầm quan trọng của Khoa Học Máy Tính**

Lĩnh vực Công Nghệ Thông Tin nói chung và Khoa Học Máy Tính nói riêng đối với sự phát triển đa ngành nghề hiện nay thì không thể thiếu tính ứng dụng của hai lĩnh vực này. Lấy một ví dụ “*Một doanh nghiệp cần đến một hệ thống phân tích hành vi tiêu dùng của các khách hàng của họ thông qua việc những dữ liệu đã được thu thập được, dựa trên sự phân tích của hệ thống mà doanh nghiệp sẽ đưa ra các chương trình khuyên mãi, các chiến lược dài hạn trong tương lai.*”, tùy theo tính hiệu quả của các giải thuật đã áp dụng trong hệ thống phân tích đó, mà có thể ảnh hưởng đến hiệu năng phân tích. Một giải thuật đủ tối ưu thì sẽ cho ra kết quả gần đúng nhất với thực tế từ đó tạo một gốc nhìn khách quan hơn cho các tổ chức doanh nghiệp, để đưa ra các lựa chọn đúng đắn.

Thông qua ví dụ trên chúng ta có thể thấy một hệ thống đã mang đến những thuận lợi như thế nào cho một tổ chức, doanh nghiệp. Từ đó sự nhân rộng của lĩnh vực *Khoa Học Máy Tính* được phát triển, thúc đẩy sự phát triển của nhiều lĩnh vực khác như giáo dục, y tế, thể thao, giải trí, xây dựng, tài chính,...kể cả lĩnh vực *Công Nghệ Thông Tin*.

1. **GIẢI THUẬT**
2. **Giải Thuật là gì?**

[Giải thuật](https://quantrimang.com/cong-nghe/cau-truc-du-lieu-va-giai-thuat) (*Algorithms*)  là một [tập hợp hữu hạn](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p#L%E1%BB%B1c_l%C6%B0%E1%BB%A3ng_c%E1%BB%A7a_t%E1%BA%ADp_h%E1%BB%A3p_-_H%E1%BB%AFu_h%E1%BA%A1n_v%C3%A0_v%C3%B4_h%E1%BA%A1n) các hướng dẫn [được xác định rõ ràng](https://vi.wikipedia.org/w/index.php?title=%C4%90%C6%B0%E1%BB%A3c_x%C3%A1c_%C4%91%E1%BB%8Bnh_r%C3%B5&action=edit&redlink=1), có thể thực hiện được bằng máy tính, thường để giải quyết một lớp vấn đề hoặc để thực hiện một phép tính. Các giải thuật luôn [rõ ràng](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%91i_ngh%C4%A9a&action=edit&redlink=1) và được sử dụng chỉ rõ việc thực hiện các [phép tính](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A9p_t%C3%ADnh&action=edit&redlink=1), [xử lý dữ liệu](https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%AD_d%E1%BB%AF_li%E1%BB%87u), [suy luận tự động](https://vi.wikipedia.org/w/index.php?title=Suy_lu%E1%BA%ADn_t%E1%BB%B1_%C4%91%E1%BB%99ng&action=edit&redlink=1) và các tác vụ khác. Nói cách khác, giải thuật cung cấp một bộ các quy tắc hay quy trình giải quyết một vấn đề nào đó sau một số bước hữu hạn xuất phát từ một tập hợp các dữ kiện cho trước. Trạng thái xuất phát (dữ liệu đầu vào) của giải thuật thường được kí hiệu là INPUT, trạng thái đạt được (kết quả đầu ra) là OUTPUT.

Từ tiếng Anh của giải thuật là *Algorithm* (là tên gọi Latin hóa của *Al-Khwarizmi* nhà toán học vùng *Vịnh Ba Tư* nửa đầu thế kỉ thứ IX) ban đầu được sử dụng để chỉ các bộ quy tắc và kỹ thuật được *Al-Khwarizmi* sử dụng để giải các phương trình đại số, trước khi được khái quát hóa để chỉ bất kỳ bộ quy tắc hoặc kỹ thuật nào.

Sự phát triển của lý thuyết giải thuật bắt đầu bằng chứng minh của *Kurt Gödel* các định lý về tính không đầy đủ của các hệ thống tiên đề (định lý đầu tiên công bố vào năm 1931) đòi hỏi cần phải chuẩn hóa khái niệm giải thuật (Định lý bất toàn *Gödel*). Các phiên bản chuẩn hóa đầu tiên của khái niệm này được phát triển vào những năm 1930 bởi *Alan Turing*, *Emil Post* và *Alonzo Church*. Sau đó, lý thuyết giải thuật đã nhận được sự phát triển hơn nữa trong các công trình của *Stephen Cole Kleene*, *Andrey Andreyevich Markov* và những người khác. Vào năm 1954 *Andrey Andreyevich Markov* đề xuất tinh chỉnh khái niệm giải thuật bằng phương tiện mà ông gọi là giải thuật bình thường, theo đánh giá được coi là “*một trong những phiên bản tiêu chuẩn hóa thành công nhất của giải thuật*”. Cách tiếp cận chung nhất khái niệm giải thuật được đề xuất bởi *Andrey Nikolaevich Kolmogorov* vào năm 1965. Trong những năm tiếp theo, *Donald Knuth*, *Alfred Aho* và *Jeffrey Ullman* đã đóng góp đáng kể cho lý thuyết giải thuật.

Giải thuật có thể được mô tả như một thủ tục hoặc công thức để giải quyết vấn đề. Chính vì vậy giải thuật được sử dụng rộng rãi trong các lĩnh vực khác nhau, đặc biệt trong toán học và khoa học máy tính, cũng như trong cuộc sống hàng ngày.

Riêng trong lĩnh vực toán học, ngoài các ứng dụng cụ thể trong đại số, giải tích, lý thuyết số, khái niệm giải thuật là cơ sở cho một trong những khái niệm trung tâm của logic toán học, cụ thể là khái niệm tính toán. Đó là lý do tại sao, định lý *Gödel* về tính không đầy đủ của các hệ thống tiên đề có thể được coi là hệ quả của các định lý của lý thuyết giải thuật. Lý thuyết giải thuật cũng có mối liên kết chặt chẽ với chính nền tảng của toán học, trong đó mối quan hệ giữa tính kiến thiết (Constructive) và không kiến thiết (Non-Constructive) chiếm một vị trí trung tâm. Có thể nói, lý thuyết giải thuật cung cấp các công cụ cần thiết để phát triển xu hướng mang tính kiến thiết trong toán học.

Trong Khoa Học Máy Tính, năm 1965, *Andrey Nikolaevich Kolmogorov* đã đề xuất sử dụng lý thuyết giải thuật làm cơ sở cho lý thuyết thông tin. Lý thuyết giải thuật cũng là nền tảng lý thuyết cho một số vấn đề của toán học tính toán và có liên quan chặt chẽ với điều khiển học, trong đó việc nghiên cứu các giải thuật điều khiển đóng vai trò quan trọng. Cuối cùng, cần phải nói, giải thuật là bước không thể thiếu trong quá trình giải quyết bất kì bài toán nào trên máy tính, do vậy khái niệm giải thuật là một trong các khái niệm cơ bản cần biết đối với bất kì ai muốn lập trình cho máy tính.

Khi một giải thuật đã hình thành, ta không xét đến việc chứng minh tính đầy đủ và tính đúng đắn của giải thuật đó mà chỉ chú trọng đến việc áp dụng các bước theo đúng trình tự quy định để thu được kết quả cần thiết. Việc chứng minh ấy phải được tiến hành trong quá trình thiết kế giải thuật (Thiết kế giải thuật). Nói cách khác, giải thuật chỉ là việc áp dụng các công thức hay quy tắc, quy trình đã được công nhận là đúng hay đã được chứng minh về mặt toán học.

1. **Đặc trưng của Giải Thuật**

Một đối tượng trong tự nhiên đều có các đặc trưng riêng của đối tượng đó, dự trên các đặc trưng đó chúng ta có thể xác định được đối tượng đó là gì. Giải thuật cũng không ngoại lệ, một giải thuật sẽ có các đặc trưng và dự trên các đặc trưng đó chúng ta có thể nhận biết được đối tượng đó có thật sự là một giải thuật, không phải thủ tục nào cũng được xem là giải thuật, các đặc trưng đó là:

* **Tính xác định**:Mộtgiải thuật nên có một cấu trúc rõ ràng và không mơ hồ. Mỗi một giai đoạn (hay mỗi bước) nên rõ ràng và chỉ mang một mục đích nhất định.
* **Dữ liệu đầu vào xác định**:Một giải thuật nên có không hoặc nhiều hơn dữ liệu đầu vào đã xác định.
* **Kết quả đầu ra**:Một giải thuật nên có một hoặc nhiều dữ liệu đầu ra đã xác định và nên kết nối với kiểu kết quả bạn mong muốn.
* **Tính dừng**:Các giải thuật phải kết thúc sau một số hữu hạn các bước.
* **Tính hiệu quả**: Một giải thuật nên là có thể thi hành được với các nguồn có sẵn, có nghĩa là có khả năng giải quyết hiệu quả vấn đề trong điều kiện thời gian và tài nguyên cho phép.
* **Tính phổ biến**:Một giải thuật có tính phổ biến nếu giải thuật này có thể giải quyết được một lớp các vấn đề tương tự.
* **Độc lập**:Một giải thuật nên có các chỉ thị độc lập với bất kỳ phần code lập trình nào.

1. **Giải Thuật quan trọng**
2. **Giải thuật tìm kiếm**

Giải thuật tìm kiếm là quy trình từng bước để tìm một phần tử cụ thể trong cấu trúc dữ liệu. Các giải thuật tìm kiếm được sử dụng trong nhiều ứng dụng, bao gồm tìm kiếm trên web, hệ thống cơ sở dữ liệu và trí tuệ nhân tạo. Có nhiều loại giải thuật tìm kiếm khác nhau, mỗi loại đều có điểm mạnh và điểm yếu riêng. Các giải thuật tìm kiếm phổ biến nhất bao gồm:

* **Linear search**: Giải thuật này chỉ đơn giản kiểm tra từng phần tử trong cấu trúc dữ liệu theo thứ tự cho đến khi tìm thấy phần tử đích. Tìm kiếm tuyến tính dễ thực hiện nhưng cũng là giải thuật tìm kiếm kém hiệu quả nhất, đặc biệt đối với các cấu trúc dữ liệu lớn.
* **Binary search**: Giải thuật này chỉ hoạt động trên các cấu trúc dữ liệu được sắp xếp. Giải thuật hoạt động bằng cách chia đôi cấu trúc dữ liệu và so sánh phần tử đích với phần tử ở giữa. Nếu phần tử đích nhỏ hơn phần tử ở giữa thì việc tìm kiếm sẽ tiếp tục ở nửa dưới của cấu trúc dữ liệu. Nếu không, việc tìm kiếm sẽ tiếp tục ở nửa trên của cấu trúc dữ liệu. Tìm kiếm nhị phân hiệu quả hơn nhiều so với tìm kiếm tuyến tính, đặc biệt đối với các cấu trúc dữ liệu lớn.
* **Hashing**: Giải thuật này hoạt động bằng cách tạo bảng băm, đây là cấu trúc dữ liệu ánh xạ các khóa tới các giá trị. Để tìm kiếm một phần tử, giải thuật băm chuyển phần tử đó thành một khóa và sau đó tra cứu khóa đó trong bảng băm. Nếu tìm thấy khóa thì giá trị tương ứng sẽ được trả về. Ngược lại, phần tử không được tìm thấy. Băm là giải thuật tìm kiếm hiệu quả nhất cho cấu trúc dữ liệu lớn, nhưng đòi hỏi nhiều bộ nhớ hơn các giải thuật tìm kiếm khác.

Giải thuật tìm kiếm tốt nhất để sử dụng tùy thuộc vào ứng dụng cụ thể. Ví dụ: nếu cấu trúc dữ liệu nhỏ và chưa được sắp xếp thì tìm kiếm tuyến tính có thể là giải pháp đơn giản và hiệu quả nhất. Nếu cấu trúc dữ liệu lớn và được sắp xếp thì tìm kiếm nhị phân là giải pháp hiệu quả nhất. Nếu cấu trúc dữ liệu lớn và chưa được sắp xếp thì băm là giải pháp hiệu quả nhất.

Ngoài 3 giải thuật tìm kiếm nêu trên còn có rất nhiều loại giải thuật tìm kiếm khác như:

* Jump search
* Interpolation search
* Exponential search
* Fibonacci search
* Depth – first search
* Breadth – first search

1. **Giải thuật sắp xếp**

Giải thuật sắp xếp là giải thuật sắp xếp một tập hợp các mục theo một thứ tự cụ thể. Thứ tự phổ biến nhất là thứ tự số tăng dần hoặc giảm dần, nhưng giải thuật sắp xếp cũng có thể được sử dụng để sắp xếp các loại dữ liệu khác, chẳng hạn như chuỗi hoặc đối tượng. Các giải thuật sắp xếp được sử dụng trong rất nhiều ứng dụng, bao gồm:

* **Database systems:** Các giải thuật sắp xếp được sử dụng để sắp xếp các bản ghi cơ sở dữ liệu nhằm giúp hệ thống dễ dàng tìm kiếm và truy xuất hơn.
* **Website search:** Giải thuật sắp xếp được sử dụng để sắp xếp kết quả tìm kiếm nhằm cung cấp kết quả phù hợp nhất cho người dùng.
* **Machine learning:** Các giải thuật sắp xếp được sử dụng để sắp xếp dữ liệu huấn luyện nhằm huấn luyện các mô hình học máy.
* **Data visualization:** Các giải thuật sắp xếp được sử dụng để sắp xếp dữ liệu trước khi được hiển thị trực quan để dễ hiểu hơn.

Có nhiều loại giải thuật sắp xếp khác nhau, mỗi loại đều có điểm mạnh và điểm yếu riêng. Các giải thuật sắp xếp phổ biến nhất bao gồm:

* **Selection sort:** Giải thuật này hoạt động bằng cách tìm phần tử nhỏ nhất trong danh sách và hoán đổi với phần tử đầu tiên. Sau đó, tìm phần tử nhỏ thứ hai trong danh sách và hoán đổi với phần tử thứ hai, v.v. Sắp xếp lựa chọn rất đơn giản để thực hiện, nhưng cũng là giải thuật sắp xếp kém hiệu quả nhất.
* **Insertion sort:** Hoạt động bằng cách chèn từng phần tử vào danh sách đã sắp xếp vào đúng vị trí. Sắp xếp chèn hiệu quả hơn sắp xếp lựa chọn, nhưng vẫn không hiệu quả lắm đối với các danh sách lớn.
* **Merge sort:** Giải thuật này hoạt động bằng cách chia danh sách làm đôi, sắp xếp đệ quy từng nửa, sau đó hợp nhất hai nửa đã sắp xếp lại với nhau. Sắp xếp hợp nhất là một giải thuật sắp xếp rất hiệu quả nhưng yêu cầu bộ nhớ bổ sung để lưu trữ hai nửa được sắp xếp của danh sách.
* **Quick sort:** Hoạt động bằng cách phân vùng danh sách xung quanh một phần tử trụ. Các phần tử nhỏ hơn phần tử trục được đặt ở một bên của trục và các phần tử lớn hơn phần tử trục được đặt ở phía bên kia của trục. Giải thuật quicksort sau đó được áp dụng đệ quy cho mỗi bên của trục xoay. Quicksort là giải thuật sắp xếp hiệu quả nhất cho hầu hết các đầu vào, nhưng có thể có độ phức tạp về thời gian trong trường hợp xấu nhất là O(n^2).

Việc lựa chọn giải thuật sắp xếp phụ thuộc vào một số yếu tố, bao gồm kích thước của danh sách, loại dữ liệu được sắp xếp và các đặc tính hiệu suất mong muốn. Giải thuật sắp xếp tốt nhất phụ thuộc vào ứng dụng cụ thể. Ví dụ: nếu danh sách nhỏ và chưa được sắp xếp thì sắp xếp lựa chọn hoặc sắp xếp chèn có thể là giải pháp đơn giản và hiệu quả nhất. Nếu danh sách lớn và được sắp xếp thì sắp xếp hợp nhất hoặc sắp xếp nhanh là giải pháp hiệu quả nhất. Nếu danh sách lớn và chưa được sắp xếp thì sắp xếp nhanh là giải pháp hiệu quả nhất, nhưng điều quan trọng là phải nhận thức được độ phức tạp về thời gian trong trường hợp xấu nhất. Ngoài ra còn có một số giải thuật sắp xếp khác chuyên biệt hơn cho một số loại dữ liệu hoặc ứng dụng nhất định. Ví dụ: sắp xếp cơ số là giải thuật sắp xếp rất hiệu quả cho số nguyên và sắp xếp đếm là giải thuật sắp xếp rất hiệu quả cho các danh sách nhỏ có phạm vi giá trị giới hạn. Nói chung, tốt nhất nên chọn giải thuật sắp xếp đã được nghiên cứu và triển khai kỹ lưỡng trong một thư viện tốt. Điều này sẽ giúp đảm bảo rằng giải thuật hiệu quả và chính xác.

1. **Giải thuật cập nhật**

Giải thuật cập nhật là một thủ tục sửa đổi trạng thái của hệ thống. Các giải thuật cập nhật được sử dụng trong rất nhiều ứng dụng, bao gồm:

* **Software systems:** Giải thuật cập nhật được sử dụng để cập nhật trạng thái của hệ thống phần mềm, chẳng hạn như cơ sở dữ liệu, hệ điều hành và ứng dụng web.
* **Machine learning systems:** Giải thuật cập nhật được sử dụng để cập nhật các tham số của mô hình học máy khi được huấn luyện trên dữ liệu mới.
* **Control systems:** Giải thuật cập nhật được sử dụng để cập nhật tín hiệu điều khiển của hệ thống điều khiển, chẳng hạn như robot và ô tô tự lái.

Có nhiều loại giải thuật cập nhật khác nhau, mỗi loại đều có điểm mạnh và điểm yếu riêng. Các giải thuật cập nhật phổ biến nhất bao gồm:

* **Gradient descent:** Giải thuật này được sử dụng để cập nhật các tham số của mô hình học máy. Hoạt động bằng cách tính toán liên tục độ dốc của hàm mất đối với các tham số và sau đó cập nhật các tham số theo hướng ngược lại của độ dốc.
* **Backpropagation:** Giải thuật này là một kiểu giảm độ dốc được sử dụng để huấn luyện mạng lưới thần kinh. Hoạt động bằng cách tính toán độ dốc của hàm mất mát tương ứng với các trọng số của mạng nơ-ron và sau đó cập nhật các trọng số theo hướng ngược lại của độ dốc.
* **Kalman filter:** Giải thuật này được sử dụng để cập nhật trạng thái của hệ thống khi có nhiễu. Hoạt động bằng cách kết hợp kiến thức tiên nghiệm về trạng thái của hệ thống với các phép đo của hệ thống để tạo ra ước tính cập nhật về trạng thái.
* **Particle filter:** Giải thuật này là một loại bộ lọc Kalman có thể được sử dụng để theo dõi nhiều mục tiêu cùng một lúc. Hoạt động bằng cách duy trì một tập hợp các hạt có trọng số, mỗi hạt đại diện cho một trạng thái có thể có của hệ thống. Giải thuật sau đó cập nhật trọng lượng của các hạt dựa trên các phép đo của hệ thống.

Việc lựa chọn giải thuật cập nhật phụ thuộc vào một số yếu tố, bao gồm loại hệ thống đang được cập nhật, sự hiện diện của nhiễu và các đặc tính hiệu suất mong muốn. Dưới đây là một số ví dụ về giải thuật cập nhật đang hoạt động:

* Hệ thống cơ sở dữ liệu sử dụng giải thuật cập nhật để cập nhật số dư tài khoản ngân hàng khi khách hàng gửi tiền hoặc rút tiền.
* Mô hình học máy sử dụng giải thuật cập nhật để cập nhật các tham số khi được đào tạo dựa trên dữ liệu mới về sở thích của khách hàng.
* Xe tự lái sử dụng giải thuật cập nhật để cập nhật các tín hiệu điều khiển khi lái xe trên đường và nhận thông tin mới từ các cảm biến.

Các giải thuật cập nhật là một phần thiết yếu của nhiều hệ thống hiện đại. Cho phép các hệ thống được sửa đổi và thích ứng với thông tin mới và các điều kiện thay đổi.

1. **Giải thuật xóa**

Giải thuật xóa là quy trình từng bước để xóa một phần tử khỏi cấu trúc dữ liệu. Các giải thuật xóa được sử dụng trong rất nhiều ứng dụng, bao gồm cơ sở dữ liệu, hệ điều hành và trình biên dịch. Giải thuật xóa cụ thể được sử dụng tùy thuộc vào loại cấu trúc dữ liệu đang được sửa đổi. Ví dụ: để xóa một phần tử khỏi một mảng, chúng ta chỉ cần dịch chuyển tất cả các phần tử tiếp theo sang trái để điền vào chỗ trống. Để xóa một phần tử khỏi danh sách liên kết, chúng ta cần tìm nút chứa phần tử cần xóa sau đó cập nhật con trỏ của các nút xung quanh để bỏ qua nút đã xóa.

Giải thuật này hoạt động bằng cách trước tiên tìm nút trước đó trong danh sách liên kết. Nếu nút cần xóa là nút đầu của danh sách liên kết thì con trỏ đầu chỉ được cập nhật tới nút tiếp theo trong danh sách. Ngược lại, con trỏ của nút trước đó sẽ được cập nhật để bỏ qua nút đã xóa. Giải thuật xóa có thể phức tạp hơn đối với các cấu trúc dữ liệu phức tạp hơn, chẳng hạn như cây tìm kiếm nhị phân và bảng băm. Tuy nhiên, nguyên tắc chung là như nhau: tìm phần tử cần xóa và sau đó cập nhật cấu trúc dữ liệu để phản ánh việc xóa.

**Chương 2**

**GIẢI THUẬT DI TRUYỀN**

1. **LỊCH SỬ VÀ THUYẾT DI TRUYỀN CỦA CHARLES DARWIN**
2. **Lịch sử về di truyền**

Từ thời xa xưa, người ta đã nhận thức được rằng mọi sinh vật trên thế giới từ thực vật đến động vật đều thể hiện ít nhiều những đặc tính từ cha và mẹ; và khác biệt giữa con và cha mẹ cũng thường được truyền lại cho thế hệ cháu. Trên cơ sở đó, con người dò đoán và tìm được phương pháp tuyển chọn tự nhiên và phát triển giống tốt cho gia súc cũng như rau cải, lúa gạo, v.v.... Hiện tượng di truyền được con người chấp nhận như chuyện hiển nhiên cho đến cuối thế kỷ XIX mới có giải thích khoa học hơn. Nhóm nghiên cứu di truyền theo chủ thuyết của *Jean-Baptiste Lamarck* cho rằng ngoài di truyền của những biểu hiện bẩm sinh còn có di truyền của những biểu hiện mắc phải bởi môi trường.

*Charles Darwin* đưa lên giả thuyết về tiến hoá vào năm 1859 nhưng gặp một số khúc mắc, khó khăn nhất là giải thích phương thức của di truyền. *Charles Darwin* cho rằng có pha trộn giữa di truyền bẩm sinh và di truyền của những biểu hiện gây nên bởi môi trường. Nhưng nếu thật sự có sự pha trộn này, chỉ sau một vài thế hệ sẽ nảy sinh ra hiện tượng đồng dạng của toàn chủng và sẽ không có đủ biến dị để sự tuyển chọn tự nhiên có thể xảy ra. Do đó, Darwin phải tiếp thu phần nào giả thuyết của *Jean-Baptiste Lamarck* vào công trình nghiên cứu của mình. Cách trình bày của *Charles Darwin* về di truyền là cho thấy nó xảy ra như thế nào và người ta có thể dự đoán hướng di truyền trong thế hệ tới (*ví dụ những đặc tính được di truyền nhưng không biểu hiện ở cha hay mẹ vào lúc thụ thai, nhiều đặc tính được di truyền phân biệt theo giống đực hay cái*). Với phương thức của di truyền này thì *Charles Darwin* không giải thích được.

Khái niệm di truyền của *Charles Darwin* được người anh em bà con của ông *Francis Galton* cải tiến sau đó, từ đó tạo cơ sở cho nền tảng khoa học di truyền. Tuy nhiên *Francis Galton* không chấp nhận thuyết tiến hóa toàn diện của *Charles Darwin* đó là về thuyết Pangenesis: là một cơ chế giả thuyết về di truyền được đề xuất bởi *Charles Darwin* trong cuốn sách The Variation of Animals and Plants under Domestication (Sự biến đổi của động vật và thực vật được thuần hóa) năm 1868. *Charles Darwin* cho rằng mỗi bộ phận của cơ thể phát ra các hạt nhỏ gọi là *Gemmules*, chúng lưu thông khắp cơ thể và cuối cùng tập trung lại ở các cơ quan sinh sản. Những hạt *Gemmules* này sau đó được truyền lại cho con cái và định hướng sự phát triển của các bộ phận tương tự trên cơ thể ở thế hệ tiếp theo.

Năm 1880, *August Friedrich Leopold Weismann* cắt đuôi của nhiều thế hệ chuột trong phòng thử nghiệm và cho thấy các con chuột trong thế hệ sau vẫn có đuôi. Từ đó, ông chứng minh rằng không có sự di truyền của các tính trạng mắc phải trong cuộc sống.

1. **Di truyền là gì?**
2. **Thành phần trong quy trình di truyền**
3. **Sơ đồ quy trình của di truyền**
4. **GIẢI THUẬT DI TRUYỀN LÀ GÌ?**
5. **CÁC PHƯƠNG PHÁP BIỂU DIỄN BÀI TOÁN TRONG GIẢI THUẬT DI TRUYỀN**
6. **CÁC TOÁN TỬ TRONG GIẢI THUẬT DI TRUYỀN**