# ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**PHAN HOÀNG NHÂN**

**ĐINH NGỌC THANH MY**

**NGHIÊN CỨU VÀ ỨNG DỤNG CÁC GIẢI THUẬT NGẪU NHIÊN**

**TÓM TẮT KHÓA LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC**

NGƯỜI HƯỚNG DẪN: TS. NGUYỄN HÒA

NGƯỜI PHẢN BIỆN: ThS. PHÙNG THÁI THIÊN TRANG

**TP. HỒ CHÍ MINH, THÁNG 10 NĂM 2013**

MỤC LỤC

[1. PHẠM VI VÀ MỤC TIÊU 2](#_Toc368494199)

[2. NHỮNG ĐÓNG GÓP CHÍNH CỦA KHÓA LUẬN 4](#_Toc368494200)

[3. CẤU TRÚC KHÓA LUẬN 4](#_Toc368494201)

[4. KẾT QUẢ ĐẠT ĐƯỢC 5](#_Toc368494202)

[5. ĐỀ NGHỊ 6](#_Toc368494203)

[TÀI LIỆU THAM KHẢO 7](#_Toc368494204)

## PHẠM VI VÀ MỤC TIÊU

Như chúng ta đã biết *giải thuật cổ điển* (classical algorithm) là cơ sở để xây dựng các hệ thống tính toán nói chung và phần mềm máy tính nói riêng [2, 3, 8]. Tính hiệu quả của các phần mềm máy tính phụ thuộc quyết định vào việc phát triển các giải thuật để tạo nên chúng. Các giải thuật càng tốt thì các phần mềm được xây dựng càng hiệu quả và khả năng đáp ứng nhu cầu thực tế càng cao. Chính vì vậy, lý thuyết thiết kế và phân tích giải thuật là một trong các lĩnh vực đã và đang được nghiên cứu mạnh mẽ, không chỉ để giải quyết các bài toán đang tồn tại mà còn đáp ứng nhu cầu giải các bài toán mới luôn luôn được đặt ra từ thực tiễn.

Theo tinh thần đó, đã có nhiều chiến lược thiết kế giải thuật kinh điển [3] được phát triển và ứng dụng như *chia để trị* (divide-and-conquer), *biến đổi để trị* (transform-and-conquer), *qui hoạch động* (dynamic programming), *tham ăn* (greedy), v.v. nhằm giảm độ phức tạp thời gian và nâng cao hiệu quả tính toán khi giải các bài toán. Tuy nhiên, có rất nhiều bài toán khó giải khi ứng dụng các kỹ thuật thiết kế giải thuật đã nêu ở trên [8]. Nói một cách khác, độ phức tạp giải thuật để giải các bài toán này là một hàm mũ theo kích thước của bài toán. Với độ phức tạp như vậy, những giải thuật này nói chung không sử dụng được trong thực tế, đặc biệt khi kích thước bài toán đủ lớn.

Một ví dụ điển hình cho lớp bài toán này là bài toán “tìm chu trình Hamilton ngắn nhất trong đồ thị có trọng số”. Cho đến nay khoa học máy tính chưa tìm thấy bất kỳ giải thuật nào tính toán đúng chu trình Hamilton ngắn nhất trong thời gian đa thức [2, 7]. Những bài toán không có giải thuật với thời gian chạy đa thức gọi là bài toán NP-Complete [8]. Có rất nhiều bài toán như vậy trong khoa học và thực tiễn cần phải giải quyết. Ví dụ, bài toán xác định tính *luôn đúng* (valid) của một *công thức logic* (well formed logic formula) là bài toán NP-Complete [8]. Một trong những phương pháp giải các bài toán NP-Complete là dùng các *giải thuật xấp xỉ* (approximation algorithm) để tìm lời giải gần đúng của bài toán với độ phức tạp đa thức [8]. Ví dụ có thể giải bài toán tìm chu trình Hamilton ngắn nhất bằng giải thuật xấp xỉ với độ phức tạp thời gian bậc hai theo số đỉnh của đồ thị.

Mặc dù giải thuật xấp xỉ là một công cụ tốt để giải các bài toán khó, nhưng chúng không thể là phương tiện để cải thiện (giảm) độ phức tạp các giải thuật đa thức đã có nhằm nâng cao hiệu quả tổng thể của các ứng dụng thực tế. Để khắc phục hạn chế của các giải thuật cổ điển nói chung và giải thuật xấp xỉ nói riêng, các *giải thuật ngẫu nhiên* (randomized algorithm) đã được nghiên cứu và phát triển [1, 2, 5, 6, 12], nhằm giảm độ phức tạp về thời gian giải toán, như một giải pháp thay thế.

Giải thuật ngẫu nhiên là giải thuật kết hợp dữ liệu ngẫu nhiên trong tiến trình thực thi các dòng lệnh của nó [1]. Hệ quả là trong các giải thuật ngẫu nhiên, các lệnh có thể được chọn ngẫu nhiên để thực hiện, dẫn đến tính *không đơn định* (nondeterministic) của giải thuật và do đó làm giảm độ phức tạp của nó. Chẳng hạn, chúng ta có thể giải bài toán xác định cặp điểm gần nhau nhất trong không gian hai chiều bằng giải thuật ngẫu nhiên tuyến tính O(*n*) [2] trong khi độ phức tạp giải thuật cổ điển tốt nhất để giải bài toán này là O(*n*log2 *n*) [3]. Hay chúng ta có thể giải bài toán tìm cây bao trùm nhỏ nhất của đồ thị có trọng số bằng giải thuật ngẫu nhiên với độ phức tạp O(*n*+*m*), trong đó *n* và *m* tương ứng là số đỉnh và cạnh của đồ thị. Với độ phức tạp O(*n*+*m*), rõ ràng giải thuật ngẫu nhiên hiệu quả hơn nhiều so với giải thuật cổ điển giải nó trong lý thuyết đồ thị mà ta đã biết [2].

Mặc dù hiệu quả hơn so với giải thuật cổ điển tương ứng và được nghiên cứu, áp dụng rộng rãi trên thế giới, nhưng trong nước nói chung và trong các trường đại học Việt Nam nói riêng, các giải thuật ngẫu nhiên lại chưa được quan tâm nhiều. Đó cũng là động lực để chúng tôi thực hiện đề tài khóa luận tốt nghiệp “*nghiên cứu và ứng dụng các giải thuật ngẫu nhiên*” như một đóng góp nhằm làm sáng tỏ hơn tính chất, khả năng của giải thuật ngẫu nhiên cũng như cách thức thiết kế, tính toán độ phức tạp, hiện thực các giải thuật ngẫu nhiên nhằm áp dụng chúng vào các bài toán cụ thể.

Để thực hiện mục tiêu này, chúng tôi đã nghiên cứu khái niệm, các đặc trưng cơ bản và cách thức phân loại của các giải thuật ngẫu nhiên. Có hai loại giải thuật ngẫu nhiên [5, 6], loại thứ nhất luôn luôn cho lời giải đúng nhưng độ phức tạp thay đổi với một xác suất nhỏ, loại thứ hai có thể sai với một xác suất đủ nhỏ nhưng độ phức tạp luôn luôn xác định. Chúng tôi xem xét, nghiên cứu cách tiếp cận, phương pháp để thiết kế các giải thuật ngẫu nhiên. Cuối cùng, chúng tôi ứng dụng giải thuật ngẫu nhiên để giải một số bài toán cụ thể (bao gồm cả giải thuật và hiện thực chương trình máy tính) như sắp xếp, kiểm tra số nguyên tố, kiểm tra phép nhân ma trận, tìm cặp điểm gần nhau nhất, tìm cây bao trùm nhỏ nhất.

## NHỮNG ĐÓNG GÓP CHÍNH CỦA KHÓA LUẬN

Dưới đây là những đóng góp chính của khóa luận đối với lĩnh vực thiết kế, phân tích và hiện thực ứng dụng giải thuật.

1. Nghiên cứu và phân loại các giải thuật ngẫu nhiên.
2. Nghiên cứu và giới thiệu các giải thuật ngẫu nhiên (mã giả và tính độ phức tạp) để giải các bài toán sắp xếp (Quicksort), kiểm tra phép nhân ma trận, kiểm tra số nguyên tố, tìm cặp điểm gần nhau nhất và tìm cây bao trùm nhỏ nhất.
3. Hiện thực một số giải thuật ngẫu nhiên một cách hiệu quả như một hệ thống để giải các bài toán đã nêu trong điểm 2.

## CẤU TRÚC KHÓA LUẬN

Khóa luận bao gồm 6 chương. Chương 1 trình bày phạm vi, mục tiêu và ý nghĩa về lý thuyết cũng như ứng dụng của đề tài khóa luận, giới thiệu cấu trúc trong khóa luận. Mỗi chương tiếp theo, từ Chương 2 đến Chương 5 có một phần giới thiệu và một phần kết luận.

Chương 2 giới thiệu tổng quan về giải thuật. Đây là những vấn đề cơ bản, cốt lõi của lý thuyết giải thuật cổ điển như khái niệm giải thuật, đặc trưng cơ bản, các phương pháp thiết kế giải thuật và các kỹ thuật tính toán độ phức tạp giải thuật.

Chương 3 trình bày giải thuật ngẫu nhiên. Đó là các vấn đề liên quan đến khái niệm, tính chất, cách phân loại giải thuật ngẫu nhiên và các lĩnh vực ứng dụng giải thuật ngẫu nhiên.

Chương 4 trình bày một số ứng dụng giải thuật ngẫu nhiên để giải một số bài toán cụ thể như sắp xếp, kiểm tra số nguyên tố, kiểm tra phép nhân ma trận, tìm cặp điểm gần nhau nhất, tìm cây bao trùm nhỏ nhất.

Chương 5 giới thiệu quá trình hiện thực các giải thuật ngẫu nhiên để giải các bài toán đã nêu trong Chương 4 (bao gồm cả đồ thị biểu diễn độ phức tạp và thời gian thực sự trên máy tính).

Chương 6 là phần tổng kết và đề nghị các hướng nghiên cứu trong tương lai liên quan đến các vấn đề của khóa luận.

## KẾT QUẢ ĐẠT ĐƯỢC

Phát triển các hệ thống tính toán lớn và hiệu quả luôn luôn là đòi hỏi cấp thiết của khoa học và thực tiễn, vì vậy việc nghiên cứu, xây dựng các giải thuật có chi phí thời gian nhỏ nói chung và giải thuật ngẫu nhiên nói riêng vẫn được tiếp tục nghiên cứu. Các vấn đề đã giải quyết và trình bày trong khóa luận này là một đóng góp trong lĩnh vực nghiên cứu và ứng dụng giải thuật ngẫu nhiên. Các kết quả đạt được trong khóa luận càng có ý nghĩa hơn khi việc nghiên cứu, phổ biến lý thuyết và ứng dụng giải thuật ngẫu nhiên tại Việt Nam còn rất hạn chế. Kết quả đạt được của khóa luận có thể được tóm lược như sau:

1. Hệ thống hóa các vấn đề liên quan đến giải thuật cổ điển như khái niệm, tính chất, ngôn ngữ biểu diễn, các phương pháp thiết kế và các kỹ thuật tính toán độ phức tạp của giải thuật.
2. Cung cấp một cách có chọn lọc, hệ thống các công cụ về toán học nói chung và lý thuyết xác suất nói riêng hỗ trợ cho việc tính toán độ phức tạp giải thuật nói chung và giải thuật ngẫu nhiên nói riêng.
3. Phát biểu và trình bày có hệ thống về khái niệm, tính chất, sự phân loại và khả năng ứng dụng của giải thuật ngẫu nhiên.
4. Đề xuất các giải thuật ngẫu nhiên hiệu quả để giải một số bài toán ứng dụng như sắp xếp dãy số, xác định số nguyên tố, kiểm tra phép nhân ma trận, tìm cặp điểm gần nhau nhất và tìm cây bao trùm nhỏ nhất.
5. Cuối cùng, hiện thực thành công các giải thuật ngẫu nhiên bằng một chương trình máy tính để giải các bài toán ứng dụng trong thực tế.

## ĐỀ NGHỊ

Từ các nghiên cứu liên quan đã được đề cập và từ các kết quả của khóa luận này, chúng tôi đề nghị một số vấn đề và hướng nghiên cứu tiếp theo như sau:

1. Khóa luận là một nghiên cứu về giải thuật ngẫu nhiên, vì vậy việc trình bày các ứng dụng chưa đi sâu vào một lĩnh vực cụ thể. Do đó đề nghị đầu tiên là thực hiện một nghiên cứu để xây dựng các giải thuật ngẫu nhiên cho phép giải các bài toán trong một lĩnh vực chuyên sâu cụ thể như lý thuyết số, lý thuyết đồ thị, v.v.
2. Như đã trình bày ở trên, một giải thuật ngẫu nhiên Monte Carlo là giải thuật có thời chạy luôn luôn cố định nhưng chỉ thành công với một xác suất nào đó. Nghĩa là giải thuật ngẫu nhiên Monte Carlo có thể trả lời sai với một xác suất nhỏ nào đó. Vì vậy một đề nghị tiếp theo là nghiên cứu để chuyển một giải thuật ngẫu nhiên loại Monte Carlo thành giải thuật ngẫu nhiên Las Vegas (loại giải thuật ngẫu nhiên luôn cho kết quả đúng).
3. Hiện thực chương trình ứng dụng như một hệ thống tiện lợi, thân thiện và chuyên nghiệp hơn nhằm tích hợp thêm các ứng dụng để giải các bài toán khác.

# TÀI LIỆU THAM KHẢO

[1] Karp Richard M. *An introduction to randomized algorithms*. Discrete Applied Mathematics, 34, 1991, 165-201.

[2] Lee R.C.T, Tseng S.S, Chang R.C, Tsai Y.T. *Introduction to The design and Analysis of Algorithms-A Strategic Approach*. McGraw-Hill Education, 2005.

[3] Levitin A. *Introduction to The design and Analysis of Algorithms*. Addison-Wesley, 2012.

[4] Martin Dietzfelbinger. *A Reliable Randomized Algorithm for the Closest-Pair Problem*. Academic Press, 1997.

[5] Mitzenmacher M., Upfal E. *Probability and Computing Randomized Algorithms and Probabilistic Analysis*. Cambridge university press, 2005.

[6] Motwani R., Prabhakar Raghavan P. *Randomized algorithms*. Cambridge university press, 1995.

[7] Rosen K.H. *Discrete Mathematics* *and Its Applications*. Prentice Hall Inc., 2012.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald D. Rivest, clifford Stein. *Introduction to Algorithms*. McGraw-Hill Book Company, 2009.

[9] <http://en.wikipedia.org/wiki/Freivalds%27_algorithm>

[10] <http://en.wikipedia.org/wiki/Randomized_algorithm>

[11] http://www.codeproject.com/Articles/2728/C-BigInteger-Class

[12] [http://www.wisdom.weizmann.ac.il/~robi/teaching/2013a-RandomizedAlgorithms](http://www.wisdom.weizmann.ac.il/~robi/teaching/2013a-RandomizedAlgorithms/)