

Combinatorics – Tổ Hợp

Nguyễn Quân Bá Hồng*

Ngày 1 tháng 5 năm 2025

Tóm tắt nội dung

This text is a part of the series *Some Topics in Advanced STEM & Beyond*:

URL: https://nqbh.github.io/advanced_STEM/.

Latest version:

- *Combinatorics – Tổ Hợp*.

PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/NQBH_combinatorics.pdf.

TeX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/NQBH_combinatorics.tex.

Mục lục

1	Graph Theory	1
1.1	[Val02; Val21]. GABRIEL VALIENTE. Algorithms on Trees & Graphs With Python Code	1
2	Wikipedia's	4
2.1	Wikipedia/extremal combinatorics	4
2.2	Wikipedia/extremal graph theory	4
2.2.1	History	4
2.2.2	Topics & concepts	4
3	Miscellaneous	5
	Tài liệu	5

1 Graph Theory

1.1 [Val02; Val21]. GABRIEL VALIENTE. Algorithms on Trees & Graphs With Python Code

- Preface to 2e. 1e has been extensively used for graduate teaching & research all over world in last 2 decades. In this new edition, have substituted detail pseudocode for both literate programming description & implementation of algorithms using LEDA library of efficient data structures & algorithms. Although pseudocode is detailed enough to allow for a straightforward implementation of algorithms in any modern programming language, have added a proof-of-concept implementation in Python of all algorithms in Appendix A. This is, therefore, a thoroughly revised & extended edition.

Regarding new material, have added an adjacency map representation of trees & graphs, & both maximum cardinality & maximum weight bipartite matching as an additional application of graph traversal techniques. Further, have revised end-of-chap problems & exercises & have included solutions to all problems in Appendix B.

- Preface to 1e. Graph algorithms, a long-established subject in mathematics & CS curricula, are also of much interest to disciplines e.g. computational molecular biology & computational chemistry. This book goes beyond *classical* graph problems of shortest paths, spanning trees, flows in networks, & matchings in bipartite graphs, & addresses further algorithmic problems of practical application on trees & graphs. Much of material presented on book is only available in specialized research literature.

– Thuật toán đồ thị, một môn học lâu đời trong chương trình giảng dạy toán học & CS, cũng rất được các ngành học quan tâm, ví dụ như sinh học phân tử tính toán & hóa học tính toán. Cuốn sách này đi xa hơn các bài toán đồ thị *cổ điển* về đường đi ngắn nhất, cây bao trùm, luồng trong mạng, & phép ghép trong đồ thị hai phần, & giải quyết các bài toán thuật toán khác về ứng dụng thực tế trên cây & đồ thị. Phần lớn tài liệu trình bày trong sách chỉ có trong tài liệu nghiên cứu chuyên ngành.

Book is structured around fundamental problem of isomorphism. Tree isomorphism is covered in much detail, together with related problems of subtree isomorphism, maximum common subtree isomorphism, & tree comparison. Graph isomorphism is also covered in much detail, together with related problems of subgraph isomorphism, maximum common subgraph isomorphism, & graph edit distance. A building block for solving some of these isomorphism problems are algorithms for finding maximal & maximum cliques.

*A Scientist & Creative Artist Wannabe. E-mail: nguyenquanbahong@gmail.com. Bến Tre City, Việt Nam.

– Sách được cấu trúc xung quanh vấn đề cơ bản về phép đồng cấu. Phép đồng cấu cây được trình bày chi tiết, cùng với các vấn đề liên quan đến phép đồng cấu cây con, phép đồng cấu cây con chung cực đại, & so sánh cây. Phép đồng cấu đồ thị cũng được trình bày chi tiết, cùng với các vấn đề liên quan đến phép đồng cấu đồ thị con, phép đồng cấu đồ thị con chung cực đại, & khoảng cách chỉnh sửa đồ thị. Một khối xây dựng để giải quyết một số vấn đề về phép đồng cấu này là các thuật toán để tìm các nhóm & cực đại.

Most intractable graph problems of practical application are not even approximable to within a constant bound, & several of isomorphism problems addressed in this book are no exception. Book can thus be seen as a companion to recent texts on approximation algorithms [1, 16], but also as a complement to previous texts on combinatorial & graph algorithms [2–15, 17].

– Hầu hết các bài toán đồ thị khó giải của ứng dụng thực tế thậm chí không thể xấp xỉ trong một giới hạn hằng số, & một số bài toán đồng cấu được đề cập trong cuốn sách này cũng không ngoại lệ. Do đó, cuốn sách có thể được coi là một phần bổ sung cho các văn bản gần đây về thuật toán xấp xỉ [1, 16], nhưng cũng là phần bổ sung cho các văn bản trước đó về thuật toán đồ thị & tổ hợp [2–15, 17].

Book is conceived on ground of 1st, introducing simple algorithms for these problems in order to develop some intuition before moving on to more complicated algorithms from research literature & 2nd, stimulating graduate research on tree & graph algorithms by providing together with underlying theory, a solid basis for experimentation & further development.

– Cuốn sách được hình thành trên cơ sở thứ nhất, giới thiệu các thuật toán đơn giản cho các bài toán này để phát triển trực giác trước khi chuyển sang các thuật toán phức tạp hơn từ tài liệu nghiên cứu & Thứ hai, kích thích nghiên cứu sau đại học về thuật toán cây & đồ thị bằng cách cung cấp cùng với lý thuyết cơ bản, một cơ sở vững chắc cho thử nghiệm & phát triển hơn nữa.

Algorithms are presented on an intuitive basis, followed by a detailed exposition in a literate programming style. Correctness proofs are also given, together with a worst-case analysis of algorithms. Further, full C++ implementation of all algorithms using LEDA library of efficient data structures & algorithms is given along book. These implementations include result checking of implementation correctness using correctness certificates.

– Thuật toán được trình bày theo cách trực quan, tiếp theo là phần trình bày chi tiết theo phong cách lập trình dễ hiểu. Các bằng chứng về tính đúng đắn cũng được đưa ra, cùng với phân tích trường hợp xấu nhất của thuật toán. Ngoài ra, triển khai C++ đầy đủ của tất cả các thuật toán sử dụng thư viện LEDA về các cấu trúc dữ liệu hiệu quả & thuật toán được đưa ra cùng với sách. Các triển khai này bao gồm kiểm tra kết quả về tính đúng đắn của triển khai bằng cách sử dụng chứng chỉ tính đúng đắn.

Choice of LEDA, which is becoming a de-facto standard for graduate courses on graph algorithms throughout world is not casual, because it allows student, lecturer, researcher, & practitioner to complement algorithmic graph theory with actual implementation & experimentation, building upon a thorough library of efficient implementations of modern data structures & fundamental algorithms.

– Việc lựa chọn LEDA, đang trở thành tiêu chuẩn thực tế cho các khóa học sau đại học về thuật toán đồ thị trên toàn thế giới, không phải là việc tùy tiện, vì nó cho phép sinh viên, giảng viên, nhà nghiên cứu, & người thực hành bổ sung lý thuyết đồ thị thuật toán bằng cách triển khai & thử nghiệm thực tế, dựa trên một thư viện toàn diện về các triển khai hiệu quả của các cấu trúc dữ liệu hiện đại & thuật toán cơ bản.

An interactive demonstration including animations of all algorithms using LEDA is given in an appendix. Interactive demonstration also includes visual checkers of implementation correctness.

– Một bản trình diễn tương tác bao gồm hoạt ảnh của tất cả các thuật toán sử dụng LEDA được đưa ra trong phần phụ lục. Bản trình diễn tương tác cũng bao gồm các công cụ kiểm tra trực quan về tính chính xác của việc triển khai.

Book is divided into 4 parts. Part I has an introductory nature & consists of 2 chaps. Chap. 1 includes a review of basic graph-theoretical notions & results used along book, a brief primer of literate programming, & an exposition of implementation correctness approach by result checking using correctness certificates. Chap. 2 is devoted exclusively to fundamental algorithmic techniques used in book: backtracking, branch-&-bound, divide-&-conquer, & DP. These techniques are illustrated by means of a running example: algorithms for tree edit distance problem.

– Sách được chia thành 4 phần. Phần I có tính chất giới thiệu & gồm 2 chương. Chương 1 bao gồm phần tổng quan về các khái niệm cơ bản về lý thuyết đồ thị & kết quả được sử dụng trong sách, một bài tóm tắt ngắn gọn về lập trình có hiểu biết, & trình bày về cách tiếp cận tính đúng đắn của việc triển khai bằng cách kiểm tra kết quả bằng cách sử dụng các chứng chỉ tính đúng đắn. Chương 2 dành riêng cho các kỹ thuật thuật toán cơ bản được sử dụng trong sách: quay lui, nhánh-&-bound, chia-&-chinh phục, & DP. Các kỹ thuật này được minh họa bằng một ví dụ đang chạy: các thuật toán cho bài toán khoảng cách chỉnh sửa cây.

Part II also consists of 2 chaps. Chap. 3 addresses most common methods for traversing general, rooted trees: depth-1st prefix leftmost (preorder), depth-1st prefix rightmost, depth-1st postfix leftmost (postorder), depth-1st postfix rightmost, breadth-1st leftmost (top-down), breadth-1st rightmost, & bottom-up traversal. Tree drawing is also discussed as an application of tree traversal methods. Chap. 4 addresses several isomorphism problems on ordered & unordered trees: tree isomorphism, subtree isomorphism, & maximum common subtree isomorphism. Computational molecular biology is also discussed as an application of different isomorphism problems on trees.

– Phần II cũng bao gồm 2 chương. Chương 3 đề cập đến các phương pháp phổ biến nhất để duyệt các cây có gốc chung: tiền tố độ sâu 1 bên trái nhất (thứ tự trước), tiền tố độ sâu 1 bên phải nhất, hậu tố độ sâu 1 bên trái nhất (thứ tự sau), hậu tố

độ sâu 1 bên phải nhất, chiều rộng 1 bên trái nhất (từ trên xuống), chiều rộng 1 bên phải nhất, duyệt & từ dưới lên. Vẽ cây cũng được thảo luận như một ứng dụng của các phương pháp duyệt cây. Chương 4 đề cập đến một số vấn đề đồng cấu trên cây có thứ tự & không có thứ tự: đồng cấu cây, đồng cấu cây con, đồng cấu cây con chung & lớn nhất. Sinh học phân tử tính toán cũng được thảo luận như một ứng dụng của các vấn đề đồng cấu khác nhau trên cây.

Part III consists of 3 chaps. Chap. 5 addresses most common methods for traversing graphs: depth-1st & breadth-1st traversal, which resp. generalize depth-1st prefix leftmost (preorder) & breadth-1st left-most (top-down) tree traversal. Leftmost depth-1st traversal of an undirected graph, a particular case of depth-1st traversal, is also discussed. Isomorphism of ordered graphs is also discussed as an application of graph traversal methods. Chap. 6 addresses related problems of finding cliques, independent sets, & vertex covers in trees & graphs. Multiple alignment of protein sequences in computational molecular biology is also discussed as an application of clique algorithms. Chap. 7 addresses several isomorphism problems on graphs: graph isomorphism, graph automorphism, subgraph isomorphism, & maximum common subgraph isomorphism. Chemical structure search is also discussed as an application of different graph isomorphism problems.

– Phần III gồm 3 chương. Chương 5 đề cập đến các phương pháp phổ biến nhất để duyệt đồ thị: duyệt theo chiều sâu 1 & theo chiều rộng 1, tương ứng là tổng quát hóa tiền tố chiều sâu 1 bên trái nhất (thứ tự trước) & duyệt cây theo chiều rộng 1 bên trái nhất (từ trên xuống). Duyệt theo chiều sâu 1 bên trái nhất của đồ thị vô hướng, một trường hợp cụ thể của duyệt theo chiều sâu 1, cũng được thảo luận. Đồng cấu của đồ thị có thứ tự cũng được thảo luận như một ứng dụng của các phương pháp duyệt đồ thị. Chương 6 đề cập đến các vấn đề liên quan đến việc tìm clique, các tập độc lập, & các lớp phủ đỉnh trong cây & đồ thị. Căn chỉnh nhiều chuỗi protein trong sinh học phân tử tính toán cũng được thảo luận như một ứng dụng của các thuật toán clique. Chương 7 đề cập đến một số vấn đề đồng cấu trên đồ thị: đồng cấu đồ thị, tự động cấu đồ thị, đồng cấu đồ thị con, đồng cấu & đồ thị con chung lớn nhất. Tìm kiếm cấu trúc hóa học cũng được thảo luận như một ứng dụng của các vấn đề đồng cấu đồ thị khác nhau.

Part IV consists of 2 appendices, followed by bibliographies references & an index. Appendix A gives an overview of LEDA, including a simple C++ representation of trees as LEDA graphs, & a C++ implementation of radix sort using LEDA. Interactive demonstration of graph algorithms presented along book is put together in Appendix B. Finally, Appendix C contains a complete index to all program modules described in book.

– Phần IV gồm 2 phụ lục, tiếp theo là các tài liệu tham khảo & một chỉ mục. Phụ lục A cung cấp tổng quan về LEDA, bao gồm một biểu diễn C++ đơn giản của cây dưới dạng đồ thị LEDA, & một triển khai C++ của thuật toán sắp xếp radix sử dụng LEDA. Bản trình bày tương tác về các thuật toán đồ thị được trình bày dọc theo sách được tập hợp trong Phụ lục B. Cuối cùng, Phụ lục C chứa một chỉ mục đầy đủ cho tất cả các mô-đun chương trình được mô tả trong sách.

This book is suitable for use in upper undergraduate & graduate level courses on algorithmic graph theory. This book can also be used as a supplementary text in basic undergraduate & graduate level courses on algorithms & data structures, & in computational molecular biology & computational chemistry courses as well. Some basic knowledge of discrete mathematics, data structures, algorithms, & programming at undergraduate level is assumed.

PART I: INTRODUCTION.

- 1. Introduction.
 - 1.1. Trees & Graphs. Notion of graph which is most useful in CS is that of a directed or just a graph. A graph is a combinatorial structure consisting of a finite nonempty set of objects, called *vertices*, together with a finite (possibly empty) set of ordered pairs of vertices, called *directed edges* or *arcs*.
 - 1.2. Basic Data Structures.
 - 1.3. Representation of Trees & Graphs.
 - Summary.
- 2. Algorithmic Techniques.

PART II: ALGORITHMS ON TREES.

- 3. Tree Traversal.
- 4. Tree Isomorphism.

PART III: ALGORITHMS ON GRAPHS.

- 5. Graph Traversal.
- 6. Clique, Independent Set, & Vertex Cover.
- 7. Graph Isomorphism.
- A: Implementation of Algorithms in Python.
- B: Solutions to All Problems.

2 Wikipedia's

2.1 Wikipedia/extremal combinatorics

“*Extremal combinatorics* is a field of mathematics, which is itself a part of mathematics. Extremal combinatorics studies how large or how small a collection of finite objects (numbers, graphs, vectors, sets, etc.) can be, if it has to satisfy certain restrictions.

Much of extremal combinatorics concerns **classes** of sets; this is called *extremal set theory*. E.g., in an n -element set, what is the largest number of k -element subsets that can pairwise intersect one another? What is the largest number of subsets of which more contains any other? The latter question is answered by **Sperner's theorem**, which gave rise to much of extremal set theory.

Another kind of example: How many people can be invited to a party where among each 3 people there are 2 who know each other & 2 who don't know each other? **Ramsey theory** shows: at most 5 persons can attend such a party (see **Theorem on Friends & Strangers**). Or, suppose given a finite set of nonzero integers, & are asked to mark as large a subset as possible of this set under the restriction that the sum of any 2 marked integers cannot be marked. It appears that (independent of what the given integers actually are) we can always mark at least $\frac{1}{3}$ of them.” – **Wikipedia/extremal combinatorics**

2.2 Wikipedia/extremal graph theory

“**Turán graph** $T(n, r)$ is an example of an extremal graph. It has the maximum possible number of edges for a graph on n vertices without $(r+1)$ -**cliques**. This is $T(13, 4)$. *Extremal graph theory* is a branch of combinatorics, itself an area of mathematics, that lies at the intersection of **extremal combinatorics** & **graph theory**. In essence, extremal graph theory studies how global properties of a graph influence local substructure. Results in extremal graph theory deal with quantitative connections between various **graph properties**, both global (e.g. number of vertices & edges) & local (e.g. existence of specific subgraphs), & problems in extremal graph theory can often be formulated as optimization problems: how big or small a parameter of a graph can be, given some constraints that the graph has to satisfy? A graph that is an optimal solution to such an optimization problem is called an *extremal graph*, & extremal graphs are important objects of study in extremal graph theory.

Extremal graph theory is closely related to fields e.g. **Ramsey theory**, **spectral graph theory**, **computational complexity theory**, & **additive combinatorics**, & frequently employs **probabilistic method**.

2.2.1 History

“Extremal graph theory, in its strictest sense, is a branch of graph theory developed & loved by Hungarians.” – BOLLOBÁS (2004)

Mantel's Theorem (1907) & **Turán's Theorem** (1941) were some of 1st milestones in stud of extremal graph theory. In particular, Turán's theorem would later on become a motivation for the finding of results e.g. **Erdős–Stone theorem** (1946). This result is surprising because it connects chromatic number with maximal number of edges in an H -free graph. An alternative proof of Erdős–Stone was given in 1975, & utilized **Szemerédi regularity lemma**, an essential technique in resolution of extremal graph theory problems.

2.2.2 Topics & concepts

- **Graph coloring**. Main article: **Wikipedia/graph coloring**. A *proper (vertex) coloring* of a graph G is a coloring of vertices of G s.t. no 2 adjacent vertices have the same color. Minimum number of colors needed to properly color G is called *chromatic number* of G , denoted $\chi(G)$. Determining chromatic number of specific graphs is a fundamental question in extremal graph theory, because many problems in area & related areas can be formulated in terms of graph coloring.

2 simple lower bounds to chromatic number of a graph G is given by **clique number** $\omega(G)$ – all vertices of a clique must have distinct colors – & by $\frac{|V(G)|}{\alpha(G)}$, where $\alpha(G)$ is independence number, because set of vertices with a given color must form an **independent set**.

A **greedy coloring** gives upper bound $\chi(G) \leq \Delta(G) + 1$, where $\Delta(G)$ is maximum degree of G . When G is not an odd cycle or a clique, **Brooks' theorem** states: upper bound can be reduced to $\Delta(G)$. When G is a **planar graph**, **4-color theorem** states: G has chromatic number ≤ 4 .

In general, determining whether a given graph has a coloring with a prescribed number of colors is known to be **NP-hard**.

In addition to vertex coloring, other types of coloring are also studied, e.g. **edge colorings**. *Chromatic index* $\chi'(G)$ of a graph G is minimum number of colors in a proper edge-coloring of a graph, & **Vizing's theorem** states: chromatic index of a graph G is either $\Delta(G)$ or $\Delta(G) + 1$.

- **Forbidden subgraphs**. Main article: **Wikipedia/forbidden subgraph problem**. *Forbidden subgraph problem* is 1 of central problems in extremal graph theory. Given a graph G , forbidden subgraph problem asks for maximal number of edges $\text{ex}(n, G)$ in an n -vertex graph that does not contain a subgraph isomorphic to G .

When $G = K_r$ is a complete graph, **Turán's theorem** gives an exact value for $\text{ex}(n, K_r)$ & characterizes all graphs attaining this maximum; such graphs are known as Turán graphs. For non-bipartite graphs G , **Erdős–Stone theorem** gives an asymptotic value of $\text{ex}(n, G)$ in terms of chromatic number of G . Problem of determining asymptotics of $\text{ex}(n, G)$ when G is a **bipartite graph** is open; when G is a complete bipartite graph, this is known as **Zarankiewicz problem**.

- **Homomorphism density.** Main article: [Wikipedia/Homomorphism density](#). *Homomorphism density* $t(H, G)$ of a graph H in a graph G describes probability that a randomly chosen map from vertex set of H to vertex set of G is also a **graph homomorphism**. It is closely related to *subgraph density*, which describes how often a graph H is found as a subgraph of G .

Forbidden subgraph problem can be restated as maximizing edge density of a graph with G -density 0, & this naturally leads to generalization in form of *graph homomorphism inequalities*, which are inequalities relating $t(H, G)$ for various graphs H . By extending homomorphism density to **graphons**, which are objects that arise as a limit of **dense graphs**, graph homomorphism density can be written in form of integrals, & inequalities e.g. **Cauchy–Schwarz inequality** & **Hölder’s inequality** can be used to derive homomorphism inequalities.

A major open problem relating homomorphism densities is **Sidorenko’s conjecture**, which states a tight lower bound on homomorphism density of a bipartite graph in a graph G in terms of edge density of G .

- **Graph regularity.** Main article: [Wikipedia/Szemerédi regularity lemma](#). Edges between parts in a regular partition behave in a “random-like” fashion. *Szemerédi’s regularity lemma* states: all graphs are ‘regular’ in sense: vertex set of any given graph can be partitioned into a bounded number of parts s.t. bipartite graph between most pairs of parts behave like **random bipartite graphs**. This partition gives a structural approximation to original graph, which reveals information about properties of original graph.

Regularity lemma is a central result in extremal graph theory, & also has numerous applications in adjacent fields of **additive combinatorics** & **computational complexity theory**. In addition to (Szemerédi) regularity, closely related notions of graph regularity e.g. strong regularity & Frieze-Kannan weak regularity have also been studied, as well as extensions of regularity to **hypergraphs**.

Applications of graph regularity often utilize forms of counting lemmas & removal lemmas. In simplest forms, **graph counting lemma** uses regularity between pairs of parts in a regular partition to approximate number of subgraphs, & **graph removal lemma** states: given a graph with few copies of a given subgraph, can remove a small number of edges to eliminate all copies of subgraph.” – [Wikipedia/extremal graph theory](#)

3 Miscellaneous

Tài liệu

- [Val02] Gabriel Valiente. *Algorithms on trees and graphs*. Springer-Verlag, Berlin, 2002, pp. xiv+490. ISBN: 3-540-43550-6. DOI: [10.1007/978-3-662-04921-1](#). URL: [https://doi.org/10.1007/978-3-662-04921-1](#).
- [Val21] Gabriel Valiente. *Algorithms on trees and graphs—with Python code*. Texts in Computer Science. Second edition [of 1926815]. Springer, Cham, [2021] ©2021, pp. xv+386. ISBN: 978-3-303-81884-5; 978-3-303-81885-2. DOI: [10.1007/978-3-030-81885-2](#). URL: [https://doi.org/10.1007/978-3-030-81885-2](#).