

# Course Project: Combinatorics & Graph Theory

## Đồ Án Môn Học: Tổ Hợp & Lý Thuyết Đồ Thị

Nguyễn Quân Bá Hồng\*

Ngày 30 tháng 7 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/](https://nqbh.github.io/advanced_STEM/).

Latest version:

- *Course Project: Combinatorics & Graph Theory – Đồ Án Môn Học: Tổ Hợp & Lý Thuyết Đồ Thị.*  
PDF: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/project/NQBH\\_combinatorics\\_graph\\_theory\\_project.pdf](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/project/NQBH_combinatorics_graph_theory_project.pdf).  
TEX: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/project/NQBH\\_combinatorics\\_graph\\_theory\\_project.tex](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/project/NQBH_combinatorics_graph_theory_project.tex).
- *Lecture Note: Combinatorics & Graph Theory – Bài Giảng: Tổ Hợp & Lý Thuyết Đồ Thị.*  
PDF: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/lecture/NQBH\\_combinatorics\\_graph\\_theory\\_lecture.pdf](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/lecture/NQBH_combinatorics_graph_theory_lecture.pdf).  
TEX: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/lecture/NQBH\\_combinatorics\\_graph\\_theory\\_lecture.tex](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/lecture/NQBH_combinatorics_graph_theory_lecture.tex).
- *Slide: Combinatorics & Graph Theory – Slide Bài Giảng: Tổ Hợp & Lý Thuyết Đồ Thị.*  
PDF: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/slide/NQBH\\_combinatorics\\_graph\\_theory\\_slide.pdf](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/slide/NQBH_combinatorics_graph_theory_slide.pdf).  
TEX: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/slide/NQBH\\_combinatorics\\_graph\\_theory\\_slide.tex](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/slide/NQBH_combinatorics_graph_theory_slide.tex).
- *Survey: Combinatorics & Graph Theory – Khảo Sát: Tổ Hợp & Lý Thuyết Đồ Thị.*  
PDF: URL: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/NQBH\\_combinatorics.pdf](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](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/NQBH_combinatorics.tex).
- Codes:
  - C/C++: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/C++.](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/C++.)
  - Pascal: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/Pascal.](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/Pascal.)
  - Python: [https://github.com/NQBH/advanced\\_STEM\\_beyond/blob/main/combinatorics/Python.](https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/Python.)

## Mục lục

<b>1 Project 1: Mathematical Induction &amp; Recurrence Relations – Đồ Án 1: Quy Nạp Toán Học &amp; Quan Hệ Truy Hồi</b>	<b>2</b>
<b>2 Project 2: Counting, Probability, Balls, &amp; Boxes – Đồ Án 2: Đếm, Xác Suất, Bánh &amp; Hộp</b>	<b>2</b>
<b>3 Project 3: Generating Functions – Đồ Án 3: Hàm Sinh</b>	<b>2</b>
<b>4 Project: Integer Partition – Đồ Án: Phân Hoạch Số Nguyên</b>	<b>2</b>
<b>5 Project 4: Graph &amp; Tree Traversing Problems – Đồ Án 4: Các Bài Toán Duyệt Đồ Thị &amp; Cây</b>	<b>3</b>
5.1 Breadth-first search (BFS) algorithm – Thuật toán tìm kiếm theo chiều rộng	3
5.2 Depth-first search (DFS) algorithm – Thuật toán tìm kiếm theo chiều sâu	3
<b>6 Project 5: Shortest Path Problems on Graphs – Đồ Án 5: Các Bài Toán Tìm Đường Đi Ngắn Nhất Trên Đồ Thị</b>	<b>3</b>

\*A scientist- & creative artist wannabe, a mathematics & computer science lecturer of Department of Artificial Intelligence & Data Science (AIDS), School of Technology (SOT), UMT Trường Đại học Quản lý & Công nghệ TP.HCM, Hồ Chí Minh City, Việt Nam.  
E-mail: [nguyenquanbahong@gmail.com](mailto:nguyenquanbahong@gmail.com) & [hong.nguyenquanba@umt.edu.vn](mailto:hong.nguyenquanba@umt.edu.vn). Website: <https://nqbh.github.io/>. GitHub: <https://github.com/NQBH>.

6.1	Dijkstra's algorithm – Thuật toán Dijkstra	3
7	Projects for 2026–?	3
7.1	Bellman–Ford algorithm – Thuật toán Bellman–Ford	3
7.2	A* algorithm – Thuật toán A*	3
7.3	Floyd–Warshall algorithm – Thuật toán Floyd–Warshall	3
7.4	Johnson's algorithm – Thuật toán Johnson	3
7.5	Viterbi's algorithm – Thuật toán Viterbi	3
8	Miscellaneous	4
	Tài liệu	4

## Rules

1. Chấp/Cho phép sử dụng AIs. Phải edit cho bớt raw AIs.
2. Các đề án nên được trình bày chi tiết nhất có thể, về cả 3 phương diện Toán học, thuật toán, & lập trình:
  - Cách thiết lập (derivation) của các công thức đệ quy, quy hoạch động.
  - Giải thích code, ý nghĩa của các biến quan trọng, i.e., các biến số chính đại diện cho đại lượng nào.
3. Gửi kèm codes C/C++, Java, Python chạy được với đề án.
4. Ưu tiên nộp đề án theo 1 GitHub repository để tiện `git clone`.
5. Đề án môn học được thực hiện theo cá nhân mỗi sinh viên, không thực hiện theo nhóm. Khi nộp sẽ có vấn đáp để kiểm tra khả năng & mức độ thấu hiểu vấn đề cũng như các phương pháp & thuật toán mà sinh viên sử dụng trong đề án của mình.
6. Mỗi sinh viên làm tất cả các đề án.

### Resources – Tài nguyên.

1. [Bal97]. V. K. BALAKRISHNAN. *Schaum's Outline of Graph Theory*.
2. [Gol18]. BORIS GOLDENGORIN. *Optimization Problems in Graph Theory*.
3. [Sha22]. SHAHRIAR SHAHRIARI. *An Invitation To Combinatorics*.
4. [Val02; Val21]. GABRIEL VALIENTE. *Algorithms on Trees & Graphs With Python Code*.

- 1 Project 1: Mathematical Induction & Recurrence Relations – Đề Án 1: Quy Nạp Toán Học & Quan Hệ Truy Hồi
- 2 Project 2: Counting, Probability, Balls, & Boxes – Đề Án 2: Đếm, Xác Suất, Bánh & Hộp
- 3 Project 3: Generating Functions – Đề Án 3: Hàm Sinh
- 4 Project: Integer Partition – Đề Án: Phân Hoạch Số Nguyên

**Bài toán 1** (Ferrers & Ferrers transpose diagrams – Biểu đồ Ferrers & biểu đồ Ferrers chuyển vị). Nhập  $n, k \in \mathbb{N}$ . Viết chương trình C/C++, Python để in ra  $p_k(n)$  biểu đồ Ferrers  $F$  & biểu đồ Ferrers chuyển vị  $F^\top$  cho mỗi phân hoạch  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k) \in (\mathbb{N}^*)^k$  có định dạng các dấu chấm được biểu diễn bởi dấu `*`.

**Bài toán 2.** Nhập  $n, k \in \mathbb{N}$ . Đếm số phân hoạch  $p_k(n)$  của  $n$  có đúng  $k$  phần tử. Viết chương trình C/C++, Python để đếm số phân hoạch  $p_{\max}(n, k)$  của  $n$  sao cho phần tử lớn nhất là  $k$ . So sánh  $p_k(n)$  &  $p_{\max}(n, k)$ .

**Bài toán 3** (Số phân hoạch tự liên hợp). Nhập  $n, k \in \mathbb{N}$ . (a) Đếm số phân hoạch tự liên hợp của  $n$  có  $k$  phần, ký hiệu  $p_k^{\text{selfcjug}}(n)$ , rồi in ra các phân hoạch đó. (b) Đếm số phân hoạch của  $n$  có lẻ phần (having odd parts), có lẻ phần phân biệt (having distinct odd parts), rồi so sánh với  $p_k^{\text{selfcjug}}(n)$ . (c) Thiết lập công thức truy hồi cho  $p_k^{\text{selfcjug}}(n)$ , rồi implementation bằng: (i) đệ quy. (ii) quy hoạch động.

## 5 Project 4: Graph & Tree Traversing Problems – Đề Án 4: Các Bài Toán Duyệt Đồ Thị & Cây

Đọc [Val21, 1.3: Representation of Trees & Graphs, p. 23].

**Bài toán 4.** *Viết chương trình C/C++, Python chuyển đổi giữa 4 dạng biểu diễn: adjacency matrix, adjacency list, extended adjacency list, adjacency map cho 3 đồ thị: đơn đồ thị, đa đồ thị, đồ thị tổng quát; & 3 dạng biểu diễn: array of parents, first-child next-sibling, graph-based representation of trees của cây.*

Sẽ có  $3A_4^3 + A_3^2 = 36 + 6 = 42$  converter programs.

**Bài toán 5.** *Làm Problems 1.1–1.6 & Exercises 1.1–1.10 [Val21, pp. 39–40].*

**Bài toán 6** (Tree edit distance). *Viết chương trình C/C++, Python để giải bài toán tree edit distance problem bằng cách sử dụng: (a) Backtracking. (b) Branch-&-bound. (c) Divide-&-conquer – chia để trị. (d) Dynamic programming – Quy hoạch động.*

**Bài toán 7** (Tree traversal – Duyệt cây). *Viết chương trình C/C++, Python để duyệt cây: (a) preorder traversal. (b) postorder traversal. (c) top-down traversal. (d) bottom-up traversal.*

### 5.1 Breadth-first search (BFS) algorithm – Thuật toán tìm kiếm theo chiều rộng

**Bài toán 8.** *Implement the breadth-first search on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

### 5.2 Depth-first search (DFS) algorithm – Thuật toán tìm kiếm theo chiều sâu

**Bài toán 9.** *Implement the depth-first search on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

## 6 Project 5: Shortest Path Problems on Graphs – Đề Án 5: Các Bài Toán Tìm Đường Đi Ngắn Nhất Trên Đồ Thị

### 6.1 Dijkstra’s algorithm – Thuật toán Dijkstra

**Bài toán 10.** *Implement the Dijkstra’s algorithm to find the shortest path problem on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

## 7 Projects for 2026–?

### 7.1 Bellman–Ford algorithm – Thuật toán Bellman–Ford

**Bài toán 11.** *Implement the Bellman–Ford algorithm to find the shortest path problem on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

### 7.2 A\* algorithm – Thuật toán A\*

**Bài toán 12.** *Implement the A\* algorithm to find the shortest path problem on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

### 7.3 Floyd–Warshall algorithm – Thuật toán Floyd–Warshall

**Bài toán 13.** *Implement the Floyd–Warshall algorithm to find the shortest path problem on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

### 7.4 Johnson’s algorithm – Thuật toán Johnson

**Bài toán 14.** *Implement the Johnson’s algorithm to find the shortest path problem on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

### 7.5 Viterbi’s algorithm – Thuật toán Viterbi

**Bài toán 15.** *Implement the Viterbi’s algorithm to find the shortest path problem on: (a) Finite simple graphs. (b) Finite multigraphs. (c) Finite general graphs.*

## 8 Miscellaneous

### Tài liệu

- [Bal97] V. K. Balakrishnan. *Schaum's Outline of Graph Theory*. Schaum's Outline Series. McGraw Hill, 1997, p. 293.
- [Gol18] Boris Goldengorin, ed. *Optimization problems in graph theory*. Vol. 139. Springer Optimization and Its Applications. In honor of Gregory Z. Gutin's 60th birthday. Springer, Cham, 2018, pp. xviii+331. ISBN: 978-3-319-94829-4; 978-3-319-94830-0. DOI: [10.1007/978-3-319-94830-0](https://doi.org/10.1007/978-3-319-94830-0). URL: <https://doi.org/10.1007/978-3-319-94830-0>.
- [Sha22] Shahriar Shahriari. *An invitation to combinatorics*. Cambridge Mathematical Textbooks. Cambridge University Press, Cambridge, 2022, pp. xv+613. ISBN: 978-1-108-47654-6.
- [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](https://doi.org/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: [10.1007/978-3-303-81885-2](https://doi.org/10.1007/978-3-303-81885-2). URL: <https://doi.org/10.1007/978-3-303-81885-2>.