

# Course Project: Combinatorics & Graph Theory

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

Nguyễn Quân Bá Hồng\*

Ngày 10 tháng 6 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 5: Graph Traversing Problems – Đồ Án 4: Các Bài Toán Duyệt Đồ Thị</b>	<b>2</b>
4.1 Breadth-first search algorithm – Thuật toán tìm kiếm theo chiều rộng	2
4.2 Depth-first search algorithm – Thuật toán tìm kiếm theo chiều rộng	2
<b>5 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>
5.1 Dijkstra’s algorithm – Thuật toán Dijkstra	3
5.2 Bellman–Ford algorithm – Thuật toán Bellman–Ford	3

\*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>.

5.3	A* algorithm – Thuật toán A*	3
5.4	Floyd–Warshall algorithm – Thuật toán Floyd–Warshall	3
5.5	Johnson’s algorithm – Thuật toán Johnson	3
5.6	Viterbi’s algorithm – Thuật toán Viterbi	3
6	Miscellaneous	4
	Tài liệu	4

## Rules

1. Chấp/Cho phép sử dụng AIs.
2. Các đồ án nên được trình bày chi tiết nhất có thể, về cả 3 phương diện Toán học, thuật toán, & lập trình:
  - 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 chạy được với đồ án.
4. Ưu tiên nộp đồ án theo 1 GitHub repository để tiện `git clone`.
5. Đồ 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 đồ án của mình.
6. Mỗi sinh viên làm tất cả các đồ 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 – Đồ Án 1: Quy Nạp Toán Học & Quan Hệ Truy Hồi**
- 2 **Project 2: Counting, Probability, Balls, & Boxes – Đồ Án 2: Đếm, Xác Suất, Bánh & Hộp**
- 3 **Project 3: Generating Functions – Đồ Án 3: Hàm Sinh**
- 4 **Project 5: Graph Traversing Problems – Đồ Án 4: Các Bài Toán Duyệt Đồ Thị**

#### 4.1 Breadth-first search algorithm – Thuật toán tìm kiếm theo chiều rộng

**Bài toán 1.** Let  $G = (V, E)$  be a finite simple graph. Implement the breadth-first search on  $G$ .

**Bài toán 2.** Let  $G = (V, E)$  be a finite multigraph. Implement the breadth-first search on  $G$ .

**Bài toán 3.** Let  $G = (V, E)$  be a general graph. Implement the breadth-first search on  $G$ .

#### 4.2 Depth-first search algorithm – Thuật toán tìm kiếm theo chiều rộng

**Bài toán 4.** Let  $G = (V, E)$  be a finite simple graph. Implement the depth-first search on  $G$ .

**Bài toán 5.** Let  $G = (V, E)$  be a finite multigraph. Implement the depth-first search on  $G$ .

**Bài toán 6.** Let  $G = (V, E)$  be a general graph. Implement the depth-first search on  $G$ .

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

Resources – Tài nguyên.

1. [Wikipedia/shortest path problem](#).

### 5.1 Dijkstra's algorithm – Thuật toán Dijkstra

**Bài toán 7.** Let  $G = (V, E)$  be a finite simple graph. Implement the Dijkstra's algorithm to find the shortest path problem on  $G$ .

**Bài toán 8.** Let  $G = (V, E)$  be a finite multigraph. Implement the Dijkstra's algorithm to find the shortest path problem on  $G$ .

**Bài toán 9.** Let  $G = (V, E)$  be a general graph. Implement the Dijkstra's algorithm to find the shortest path problem on  $G$ .

### 5.2 Bellman–Ford algorithm – Thuật toán Bellman–Ford

**Bài toán 10.** Let  $G = (V, E)$  be a finite simple graph. Implement the Bellman–Ford algorithm to find the shortest path problem on  $G$ .

**Bài toán 11.** Let  $G = (V, E)$  be a finite multigraph. Implement the Bellman–Ford algorithm to find the shortest path problem on  $G$ .

**Bài toán 12.** Let  $G = (V, E)$  be a general graph. Implement the Bellman–Ford algorithm to find the shortest path problem on  $G$ .

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

**Bài toán 13.** Let  $G = (V, E)$  be a finite simple graph. Implement the A\* algorithm to find the shortest path problem on  $G$ .

**Bài toán 14.** Let  $G = (V, E)$  be a finite multigraph. Implement the A\* algorithm to find the shortest path problem on  $G$ .

**Bài toán 15.** Let  $G = (V, E)$  be a general graph. Implement the A\* algorithm to find the shortest path problem on  $G$ .

### 5.4 Floyd–Warshall algorithm – Thuật toán Floyd–Warshall

**Bài toán 16.** Let  $G = (V, E)$  be a finite simple graph. Implement the Floyd–Warshall algorithm to find the shortest path problem on  $G$ .

**Bài toán 17.** Let  $G = (V, E)$  be a finite multigraph. Implement the Floyd–Warshall algorithm to find the shortest path problem on  $G$ .

**Bài toán 18.** Let  $G = (V, E)$  be a general graph. Implement the Floyd–Warshall algorithm to find the shortest path problem on  $G$ .

### 5.5 Johnson's algorithm – Thuật toán Johnson

**Bài toán 19.** Let  $G = (V, E)$  be a finite simple graph. Implement the Johnson's algorithm to find the shortest path problem on  $G$ .

**Bài toán 20.** Let  $G = (V, E)$  be a finite multigraph. Implement the Johnson's algorithm to find the shortest path problem on  $G$ .

**Bài toán 21.** Let  $G = (V, E)$  be a general graph. Implement the Johnson's algorithm to find the shortest path problem on  $G$ .

### 5.6 Viterbi's algorithm – Thuật toán Viterbi

**Bài toán 22.** Let  $G = (V, E)$  be a finite simple graph. Implement the Viterbi's algorithm to find the shortest path problem on  $G$ .

**Bài toán 23.** Let  $G = (V, E)$  be a finite multigraph. Implement the Viterbi's algorithm to find the shortest path problem on  $G$ .

**Bài toán 24.** Let  $G = (V, E)$  be a general graph. Implement the Viterbi's algorithm to find the shortest path problem on  $G$ .

## 6 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>.