

Lecture Note: Combinatorics & Graph Theory

Bài Giảng: Tổ Hợp & Lý Thuyết Đồ Thị

Nguyễn Quân Bá Hồng*

Ngày 8 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:

- *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.
TeX: URL: 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.
TeX: URL: 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.
TeX: URL: 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++.
 - Pascal: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/Pascal.
 - Python: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/Python.

Mục lục

1 Basic Combinatorics – Tổ Hợp Cơ Bản	2
1.1 Set theory	3
1.2 Inclusion–exclusion principle – Nguyên lý bù trừ	3
1.3 Problems on inclusion–exclusion principle	5
1.4 Problems on counting	6
1.5 Method of mathematical induction & recurrence – Phương pháp quy nạp toán học & truy hồi/đệ quy	6
1.6 Principle of strong induction – Nguyên lý quy nạp mạnh	7
1.7 Fibonacci & Lucas numbers	7
1.8 Recurrence Relations – Quan hệ truy hồi/hồi quy/đệ quy	7
1.9 Pigeonhole principle & Ramsey theory – Nguyên lý chuồng bồ câu & lý thuyết Ramsey	7
1.10 Counting rules & Stirling number of type 1 & type 2	7
1.11 Hoán vị & tổ hợp	8
1.12 Hệ số nhị thức & đa thức	9
1.13 Phân vùng số nguyên & nguyên tắc loại suy	9
1.14 Generating function – Hàm sinh	9
2 Graph Theory – Lý Thuyết Đồ Thị	10
2.1 Trees & graphs: Some basic concepts – Cây & đồ thị: Vài khái niệm cơ bản	10
2.1.1 Walks, trails, paths, cycles, & complete graphs	12
2.1.2 Graphic sequences	13
2.1.3 Miscellaneous: Graph theory	13

*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 & hong.nguyenquanba@umt.edu.vn. Website: <https://nqbh.github.io/>. GitHub: <https://github.com/NQBH>.

3 Algorithms on Graphs – Các Thuật Toán Trên Đồ Thị	14
3.1 Shortest path problem – Bài toán tìm đường đi ngắn nhất	14
3.1.1 Algorithms	15
3.2 Dijkstra’s algorithm – Thuật toán Dijkstra	15
4 CSES Problem Set/Graph Algorithms	15
5 CSES Problem Set/Tree Algorithms	19
6 Posets, Kết Nối, Lưới Boolean	19
7 Miscellaneous	19
Tài liệu	19

Preliminaries

Notation – Ký hiệu

- $\overline{m, n} := \{m, m+1, \dots, n-1, n\}$, $\forall m, n \in \mathbb{Z}$, $m \leq n$. Hence the notation “for $i \in \overline{m, n}$ ” means “for $i = m, m+1, \dots, n$ ”, i.e., chỉ số/biến chạy i chạy từ $m \in \mathbb{Z}$ đến $n \in \mathbb{Z}$. Trong trường hợp $a, b \in \mathbb{R}$, ký hiệu $\overline{a, b} := [\overline{a}], [\overline{b}]$ có nghĩa như định nghĩa trước đó với $m := \lceil a \rceil$, $n := \lfloor b \rfloor \in \mathbb{Z}$; khi đó ký hiệu “for $i \in \overline{a, b}$ ” với $a, b \in \mathbb{R}$, $a \leq b$ có nghĩa là “for $i = \lceil a \rceil, \lceil a \rceil + 1, \dots, \lfloor b \rfloor - 1, \lfloor b \rfloor$ ”, i.e., chỉ số/biến chạy i chạy từ $\lceil a \rceil$ đến $\lfloor b \rfloor \in \mathbb{Z}$.
- $\{x\}$, $\{x\}$ lần lượt được gọi là *phần nguyên & phần lẻ* (integer- & fractional parts) của $x \in \mathbb{R}$, see, e.g., [Wikipedia/floor & ceiling functions](#), [Wikipedia/fractional part](#).
- $x_+ := \max\{x, 0\}$, $x_- := \max\{-x, 0\} = -\min\{x, 0\}$ lần lượt được gọi là *phần dương & phần âm* (positive- & negative parts) của $x \in \mathbb{R}$.
- s.t.: abbreviation of ‘such that’.
- w.l.o.g.: abbreviation of ‘without loss of generality’.
- $|A|$ or $\#A$: the number of elements of a set A – số phần tử của 1 tập hợp A hữu hạn.
- $\text{card}(A)$: cardinality of a set A (finite or infinite) – lực lượng của 1 tập hợp A (hữu hạn hoặc vô hạn).
- $[n] := \{1, 2, \dots, n\}$: the set of 1st $n \in \mathbb{N}^*$ positive integers, which serves as 1 of prototypical example of a finite set with n elements – tập hợp $n \in \mathbb{N}^*$ số nguyên dương đầu tiên, đóng vai trò là 1 trong ví dụ nguyên mẫu của một tập hợp hữu hạn với n phần tử. Quy ước $[0] = \emptyset$ ký hiệu tập rỗng, i.e., tập hợp không chứa bất cứ phần tử nào.
- A_n^k : Chính hợp chập k phần tử từ 1 tập hợp có n phần tử.
- $C_n^k = \binom{n}{k}$: Tổ hợp chập k phần tử từ 1 tập hợp có n phần tử.
- $P_n = n!$, $\forall n \in \mathbb{N}$: the number of permutations.
- O: Olympiad problem – Bài tập luyện Olympic Toán Hoặc Olympic Tin.
- R: Research-oriented problems – Bài tập định hướng nghiên cứu.

1 Basic Combinatorics – Tổ Hợp Cơ Bản

Combinatorics is a collection of techniques & a language for the study of (finite or countably infinite) discrete structures. Given a set of elements (& possibly some structure on that set), typical questions in combinatorics are:

- Does a specific arrangement of the elements exist?
 - How many such arrangements are there?
 - What properties do these arrangements have?
 - Which 1 of the arrangements is maximal, minimal, or optimal according to some criterion?
- Tổ hợp là một tập hợp các kỹ thuật & một ngôn ngữ để nghiên cứu các cấu trúc rời rạc (hữu hạn hoặc vô hạn đếm được). Cho một tập hợp các phần tử (& có thể có một số cấu trúc trên tập hợp đó), các câu hỏi điển hình trong tổ hợp là:
- Có tồn tại sự sắp xếp cụ thể của các yếu tố không?
 - Có bao nhiêu sự sắp xếp như vậy?
 - Những sự sắp xếp này có những tính chất gì?
 - Sự sắp xếp nào là tối đa, tối thiểu hoặc tối ưu theo một số tiêu chí?

1.1 Set theory

Resources – Tài nguyên.

1. [Hal60; Hal74]. PAUL RICHARD HALMOS, *Naive Set Theory*.

Comments. 1 trong những quyển sách có nội dung đẹp nhất về Lý Thuyết Tập Hợp Ngây Thơ (distinguish Naive Set Theory vs. Axiomatic Set Theory).

2. [Kap72; Kap77]. IRVING KAPLANSKY. *Set Theory & Metric Spaces*.

A *multiset* is like a set except that its members need not be distinct.

Definition 1 (Multisets, [Sha22], Def. 2.13, p. 53). A multiset is a set together with a function that assigns a positive integer or ∞ – called a repetition number (or a multiplicity) – to each member of the set.

Định nghĩa 1 (Đa tập). Một đa tập là một tập hợp cùng với một hàm gán một số nguyên dương hoặc ∞ – được gọi là số lặp lại (hoặc bội số) – cho mỗi phần tử của tập hợp.

Example 1 ([Sha22], Ex. 2.14, p. 53). $A = \{a, a, a, b, c, c, d, d\} = \{3 \cdot a, 2 \cdot c, 2 \cdot d\}$ is a multiset with members a, b, c, d & with repetitions numbers 3, 1, 2, 2, resp.

Remark 1 ([Sha22], Rmk. 2.15, p. 53). A set can be thought of as a multiset with all repetition numbers equal to 1. In an infinite multiset, some members could have infinite repetition numbers. I.e., we allow for the possibility that a multiset contains an infinite number of copies of a certain element.

– Một tập hợp có thể được coi là một đa tập hợp với tất cả các số lặp lại bằng 1. Trong một đa tập hợp vô hạn, một số phần tử có thể có số lặp lại vô hạn. Tức là, chúng ta cho phép khả năng một đa tập hợp chứa một số lượng vô hạn các bản sao của một phần tử nhất định.

Definition 2 ($|\cdot|$ notation). If X is a finite set, then $|X|$ will denote the number of elements in X . If X is a finite multiset, then $|X|$ will denote the sum of its repetition numbers.

Định nghĩa 2 (Ký hiệu $|\cdot|$). Nếu X là một tập hợp hữu hạn, thì $|X|$ sẽ biểu thị số phần tử trong X . Nếu X là một đa tập hữu hạn, thì $|X|$ sẽ biểu thị tổng số lần lặp lại của nó.

1.2 Inclusion–exclusion principle – Nguyên lý bù trừ

In combinatorics, the *inclusion–exclusion principle* is a counting technique which generalizes the familiar method of obtaining the number of elements in the **union** of 2 **finite sets**; symbolically expressed as $|A \cup B| = |A| + |B| - |A \cap B|$ where A, B : 2 finite sets & $|S|$ indicates the **cardinality** of a set S (which may be considered as the number of elements of the set, if the set is finite). The formula expresses the fact that the sum of the sizes of the 2 sets may be too large since some elements may be counted twice. The double-counted elements are those in the **intersection** of the 2 sets & the count is corrected by subtracting the size of the intersection.

The inclusion–exclusion principle, being a generalization of the 2-set case, is perhaps more clearly seen in the case of 3 sets, which for the sets A, B, C is given by $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$. This formula can be verified by counting how many times each region in the **Venn diagram** figure is included in RHS of the formula. In this case, when removing the contributions of over-counted elements, the number of elements in the mutual intersection of 3 sets has been subtracted too often, so must be added back in to get the corrected total.

Generalizing the results of these examples gives the principle of inclusion–exclusion. To find the cardinality of the union of n sets:

1. Include the cardinalities of the sets.
2. Exclude the cardinalities of pairwise intersections.
3. Include the cardinalities of the triple-wise intersections.
4. Exclude the cardinalities of the quadruple-wise intersections.
5. Include the cardinalities of quintuple-wise intersections.
6. Continue, until the cardinality of the n -tuple-wise intersection is included (if n is odd) or excluded (if n is even).

The name comes from the idea that the principle is based on over-generous *inclusion*, followed by compensating *exclusion*. The principle can be viewed as an example of the **sieve method** extensively used in **number theory** & is sometimes referred to as the *sieve formula*.

As finite probabilities are computed as counts relative to the cardinality of the **probability space**, the formulas for the principle of inclusion–exclusion remain valid when the cardinalities of the sets are replaced by finite probabilities. More generally, both versions of the principle can be put under the common umbrella of **measure theory**.

In a very abstract setting, the principle of inclusion–exclusion can be expressed as the calculation of the inverse of a certain matrix. This inverse has a special structure, making the principle an extremely valuable technique in combinatorics & related areas of mathematics.

“1 of the most useful principles of enumeration is discrete probability & combinatorial theory is the celebrated principle of inclusion–exclusion. When skillfully applied, this principle has yielded the solution to many a combinatorial problem.” – **GIAN-CARLO ROTA**

For more details, see, e.g., [Wikipedia/inclusion–exclusion principle](#).

Định lý 1 (Nguyên lý bù trừ/nguyên lý bao hàm–loại trừ).

- (i) Với 2 tập hợp hữu hạn A, B bất kỳ, $|A \cup B| = |A| + |B| - |A \cap B|$, $|A \setminus B| = |A| - |A \cap B|$.
- (ii) Với 3 tập hợp hữu hạn A, B, C bất kỳ, $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$.
- (iii) Với 4 tập hợp hữu hạn A, B, C, D bất kỳ, $|A \cap B \cap C \cap D| = |A| + |B| + |C| + |D| - |A \cap B| - |A \cap C| - |A \cap D| - |B \cap C| - |B \cap D| - |C \cap D| + |A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D| - |A \cap B \cap C \cap D|$.
- (iv) Với $n \in \mathbb{N}^*$, A_i , $i = 1, \dots, n$, là n tập hợp hữu hạn bất kỳ:

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{T \subseteq \{1, \dots, n\}, T \neq \emptyset} (-1)^{|T|+1} \left| \bigcap_{i \in T} A_i \right|.$$

Từ đó suy ra,

$$\left| \bigcup_{i=1}^n A_i \right| \geq \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j|.$$

Prove that if you take into account only the 1st $m < n$ sums on the right (in the general form of the principle), then you will get an overestimate if m is odd & an underestimate if m is even.

Chứng minh. Vẽ biểu đồ Venn (Venn diagram) để tiện lập luận.

(i) Các phần tử x của tập hợp A gồm 2 loại:

- $x \in A$ nhưng $x \notin B$, i.e., $x \in A \setminus B$: có đúng $|A \setminus B|$ phần tử x như vậy.
- $x \in A$ & $x \in B$, i.e., $x \in A \cap B$: có đúng $|A \cap B|$ phần tử như vậy.

Suy ra tổng số phần tử của tập A bằng $|A \setminus B| + |A \cap B|$, i.e., $|A| = |A \setminus B| + |A \cap B|$, hay $|A \setminus B| = |A| - |A \cap B|$. Chứng minh tương tự cho tập hợp B được: $|B| = |B \setminus A| + |B \cap A|$, hay $|B \setminus A| = |B| - |A \cap B|$. Áp dụng 2 kết quả này được: $|A \cup B| = |A \setminus B| + |A \cap B| + |B \setminus A| = |A| - |A \cap B| + |A \cap B| + |B| - |A \cap B| = |A| + |B| - |A \cap B|$.

(ii) Áp dụng kết quả ý (i) cho $(A, B) = (A, B \cup C)$ được:

$$\begin{aligned} |A \cup B \cup C| &= |A \cup (B \cup C)| = |A| + |B \cup C| - |A \cap (B \cup C)| = |A| + |B| + |C| - |B \cap C| - |(A \cap B) \cup (A \cap C)| \\ &= |A| + |B| + |C| - |B \cap C| - (|A \cap B| + |A \cap C| - |(A \cap B) \cap (A \cap C)|) \\ &= |A| + |B| + |C| - |B \cap C| - |C \cap A| - |A \cap B| + |A \cap B \cap C|. \end{aligned}$$

(iii) Chứng minh bằng quy nạp. □

Theorem 1 (Inclusion–exclusion principle). For any $n \in \mathbb{N}^*$, & any finite sets A_1, \dots, A_n , one has the identity

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n+1} |A_1 \cap \dots \cap A_n|,$$

which can be compactly written as

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{i=1}^n (-1)^{k+1} \sum_{1 \leq i_1 < \dots < i_k \leq n} |A_{i_1} \cap \dots \cap A_{i_k}|,$$

or

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{\emptyset \neq J \subseteq \{1, \dots, n\}} (-1)^{|J|+1} \left| \bigcap_{j \in J} A_j \right|.$$

In words, to count the number of elements in a finite union of finite sets, 1st sum the cardinalities of the individual sets, then subtract the number of elements that appear in at least 2 sets, then add back the number of elements that appear in at least 3 sets, then subtract the number of elements that appear in at least 4 sets, & so on. This process always ends since there can be no elements that appear in more than the number of sets in the union.

In applications, it is common to see the principle expressed in its complementary form:

Theorem 2 (Complementary form of inclusion–exclusion principle). *Let S be a finite **universal set** containing all of the A_i & letting $\overline{A_i}$ denote the complement of A_i in S , by **De Morgan's laws**, one has*

$$\left| \bigcap_{i=1}^n \overline{A_i} \right| = \left| S \setminus \bigcup_{i=1}^n A_i \right| = |S| - \sum_{i=1}^n |A_i| + \sum_{1 \leq i < j \leq n} |A_i \cap A_j| - \cdots + (-1)^n |A_1 \cap \cdots \cap A_n|.$$

As another variant due to **J. J. SYLVESTER** of the statement, let P_1, \dots, P_n be a list of properties that elements of a set S may or may not have, then the principle of inclusion–exclusion provides a way to calculate the number of elements of S that have none of the properties. Just let A_i be the subset of elements of S which have the property P_i & use the principle in its complementary form.

Theorem 3 (Inclusion–exclusion principle in probability). *For any $n \in \mathbb{N}^*$, & for any events A_1, \dots, A_n in a **probability space** $(\Omega, \mathcal{F}, \mathbb{P})$:*

(i) For $n = 2$, $\mathbb{P}(A_1 \cup A_2) = \mathbb{P}(A_1) + \mathbb{P}(A_2) - \mathbb{P}(A_1 \cap A_2)$.

(ii) For $n = 3$, $\mathbb{P}(A_1 \cup A_2 \cup A_3) = \mathbb{P}(A_1) + \mathbb{P}(A_2) + \mathbb{P}(A_3) - \mathbb{P}(A_1 \cap A_2) - \mathbb{P}(A_2 \cap A_3) - \mathbb{P}(A_3 \cap A_1) + \mathbb{P}(A_1 \cap A_2 \cap A_3)$.

(iii) In general

$$\mathbb{P}\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n \mathbb{P}(A_i) - \sum_{i < j} \mathbb{P}(A_i \cap A_j) + \sum_{i < j < k} \mathbb{P}(A_i \cap A_j \cap A_k) - \cdots + (-1)^{n-1} \mathbb{P}\left(\bigcap_{i=1}^n A_i\right), \quad (1)$$

which can be written in closed form as

$$\mathbb{P}\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n (-1)^{k-1} \sum_{I \subset [n], |I|=k} \mathbb{P}(A_I), \quad (2)$$

where the last sum runs over all subsets I of the indices $1, \dots, n$ which contain exactly k elements, & $A_I := \bigcap_{i \in I} A_i$ denotes the intersection of all those A_i with index in I .

In particular, if the probability of the intersection A_I only depends on the cardinality of I , i.e., for every $k \in [n]$, there is an a_k s.t. $a_k = \mathbb{P}(A_I)$ for every $I \in [n]$ with $|I| = k$, then (2) simplifies to

$$\mathbb{P}\left(\bigcup_{i=1}^n A_i\right) = \sum_{k=1}^n (-1)^{k-1} \binom{n}{k} a_k.$$

In addition, if the events A_i are **independent & identically distributed** (i.i.d.), then $\mathbb{P}(A_i) = p$, $\forall i$, & $a_k = p^k$, hence

$$\mathbb{P}\left(\bigcup_{i=1}^n A_i\right) = 1 - (1 - p)^n.$$

Problem 1. *Prove this theorem (the last result can also be derived more simply by considering the intersection of the complements of the events A_i .)*

According to the **Bonferroni inequalities**, the sum of the 1st terms in the formula is alternately an upper bound & a lower bound for the LHS. This can be used in cases where the full formula is too cumbersome.

For a general **measure space** (S, Σ, μ) & **measurable** subsets A_1, \dots, A_n of **finite measure**, the above identities also hold when the probability measure \mathbb{P} is replaced by the measure μ .

Theorem 4 (General form).

$$g(A) = \sum_{S \subset A} f(S) \Rightarrow f(A) = \sum_{S \subset A} (-1)^{|A|-|S|} g(S).$$

The combinatorial version Thm. 1 & the probabilistic version Thm. 3 of the inclusion–exclusion principle are instances of 4.

1.3 Problems on inclusion–exclusion principle

Problem 2 (Counting derangements – đếm số quân bài đánh tráo). *Suppose there is a deck of n cards numbered from 1 to n . Suppose a card numbered m is in the correct position if it is the m th card in the deck. How many ways, W , can the cards be shuffled with at least 1 card being in the correct position?*

– Giả sử có một bộ bài n lá bài được đánh số từ 1 đến n . Giả sử một lá bài được đánh số m ở đúng vị trí nếu nó là lá bài thứ m trong bộ bài. Có bao nhiêu cách, W , có thể xáo trộn các lá bài sao cho ít nhất 1 lá bài ở đúng vị trí?

1.4 Problems on counting

Bài toán 1. 1 lớp có $n \in \mathbb{N}^*$ sinh viên. Lớp muốn trong bức ảnh có $a \in \mathbb{N}^*$ ngồi hàng đầu & $n - a$ sinh viên ngồi hàng sau. Đếm số cách.

Chứng minh. Chọn ra a bạn ngồi hàng đầu & sắp vị trí: có A_n^a cách. Hoán vị $n - a$ bạn còn lại vào hàng sau: có $(n - a)!$ cách. Theo quy tắc nhân, có $A_n^a(n - a)! = \frac{n!}{(n - a)!}(n - a)! = n!$ cách. \square

Bài toán 2 (O, +1). Cho $n \in \mathbb{N}^*$. Chứng minh số cách khác nhau để đặt n dấu ngoặc mở & n dấu ngoặc đóng đúng đắn bằng số Catalan thứ n :

$$C_n := \frac{1}{n+1} C_{2n}^n = \frac{(2n)!}{n!(n+1)!}, \forall n \in \mathbb{N}^*.$$

Bài toán 3 (+0.5). Cho $n \in \mathbb{N}^*$, $k \in \mathbb{N}$, $0 \leq k \leq n$. Viết chương trình C/C++, Pascal, Python để tính & cho biết giá trị nhỏ nhất của n gặp lỗi overflow: (a) $P_n = n!$. (b) A_n^k . (c) C_n^k . (d) Số Catalan thứ n .

1.5 Method of mathematical induction & recurrence – Phương pháp quy nạp toán học & truy hồi/đệ quy

Ideas of method of mathematical induction. In philosophy & in the sciences, the inductive method refers to the process of starting with observations & then looking for general laws & theories. Mathematical induction – which we just called *induction* – is quite different. For us, induction is a method of proof. When we have an infinite sequence $\{P_n\}_{n=1}^\infty = P_1, P_2, \dots$ of (related) mathematical statements that need a proof, then instead of proving each 1 of these statements, we may be able to just prove that whenever 1 of the statements is true, then so is the next statement (i.e., if P_k is true, then so is P_{k+1}). If we manage such a feat, then proving P_1 , the 1st statement, starts a domino effect resulting in all of the statements being true.

– Ý tưởng về phương pháp quy nạp toán học. Trong triết học & trong khoa học, phương pháp quy nạp đề cập đến quá trình bắt đầu bằng các quan sát & sau đó tìm kiếm các định luật & lý thuyết chung. Quy nạp toán học – mà chúng ta chỉ gọi là *quy nạp* – thì khá khác. Đối với chúng ta, quy nạp là một phương pháp chứng minh. Khi chúng ta có một chuỗi vô hạn $\{P_n\}_{n=1}^\infty = P_1, P_2, \dots$ các mệnh đề toán học (có liên quan) cần được chứng minh, thì thay vì chứng minh từng 1 trong số các mệnh đề này, chúng ta có thể chỉ cần chứng minh rằng bất cứ khi nào 1 trong các mệnh đề là đúng, thì mệnh đề tiếp theo cũng vậy (tức là nếu P_k đúng, thì P_{k+1} cũng đúng). Nếu chúng ta thực hiện được kỳ tích như vậy, thì việc chứng minh P_1 , mệnh đề đầu tiên, sẽ bắt đầu một hiệu ứng domino khiến tất cả các mệnh đề đều đúng.

Intuition. A sequence of *consecutive* pieces of domino falling.

– Trực giác. Phương pháp quy nạp toán học hoạt động như cách 1 chuỗi các quân cờ domino *liên tiếp* rơi xuống sau khi đẩy ngã quân domino đầu tiên (1 quân domino bất kỳ bị ngã sẽ khiến quân domino tiếp theo nó bị ngã).

Principle of Mathematical Induction. Given an infinite sequence of propositions $\{P_n\}_{n=1}^\infty = P_1, P_2, \dots, P_n, \dots$, in order to prove that all of them are true, it is enough to show 2 things:

- The base case: P_1 is true.
- The inductive step: For all $k \in \mathbb{N}^*$, if P_k is true, then so is P_{k+1} .

Remark 2. 1 drawback of mathematical induction: to use it, you already have to know the pattern & the answer. Mathematical induction – unlike the inductive method in science – does not help in finding the pattern. 1 strength of induction: it allows you to use P_k to prove P_{k+1} . This is a big advantage – it almost seems like cheating – since often P_k looks very much like P_{k+1} .

– 1 nhược điểm của quy nạp toán học: để sử dụng nó, bạn phải biết mô hình & câu trả lời. Quy nạp toán học – không giống như phương pháp quy nạp trong khoa học – không giúp tìm ra mô hình. 1 điểm mạnh của quy nạp: nó cho phép bạn sử dụng P_k để chứng minh P_{k+1} . Đây là một lợi thế lớn – gần giống như gian lận – vì thường thì P_k trông rất giống P_{k+1} .

Bài toán 4. Chứng minh: (a) (Tổng của n số nguyên dương đầu tiên, n số nguyên lẻ đầu tiên, n số nguyên dương chẵn đầu tiên)

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}, \sum_{i=1}^n (2i-1) = n^2, \sum_{i=1}^n 2i = n(n+1), \forall n \in \mathbb{N}^*. (b) \text{ (Tổng bình phương của } n \text{ số nguyên dương đầu tiên,}$$

$$n \text{ số nguyên lẻ đầu tiên, } n \text{ số nguyên dương chẵn đầu tiên)} \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}, \sum_{i=1}^n (2i-1)^2, \sum_{i=1}^n (2i)^2, \forall n \in \mathbb{N}^*.$$

$$(c) \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 = \frac{n^2(n+1)^2}{4}, \forall n \in \mathbb{N}^*. (d) \text{ Tìm cách tính } S(n, k) := \sum_{i=1}^n i^k, \forall n, k \in \mathbb{N}^*. (e) \text{ Tính } \sum_{i=1}^n \frac{1}{i(i+1)} = \frac{n}{n+1}, \forall n \in \mathbb{N}^*. (f) \text{ Tính } \sum_{i=1}^n \frac{1}{\sqrt{i} + \sqrt{i+1}} = \sqrt{n+1} - 1, \forall n \in \mathbb{N}^*. (g)$$

$$\prod_{i=1}^n \frac{i^3 - 1}{i^3 + 1} = \frac{2(n^2 + n + 1)}{3n(n+1)}, \forall n \in \mathbb{N}^*.$$

See, e.g., [Wikipedia/ring of integers](#).

Bài toán 5. Cho $m \in \mathbb{N}^*$ là 1 số không chính phương. Chứng minh: (a) $\exists A, B \in \mathbb{N}^*$ s.t. $(a + b\sqrt{m})^n = A + B\sqrt{m}, \forall n \in \mathbb{N}^*$. (b) $\exists A, B \in \mathbb{N}^*$ s.t. $(a - b\sqrt{m})^n = A - B\sqrt{m}, \forall n \in \mathbb{N}^*$.

Bài toán 6. Chứng minh: (a) $(a+1)^n - an - 1 : n^2, \forall n \in \mathbb{N}^*, \forall a \in \mathbb{Z}$. (b) $(a+1)^n - \frac{n(n-1)}{2}a^2 - an - 1 : n^3, \forall a \in \mathbb{Z}$.

Problem 3 ([Sha22], p. 6). On a large square piece of paper, draw $n \in \mathbb{N}^*$ straight lines that start from 1 side of the square & end on another side. Each 2 of the lines intersect but no 3 (or more) lines go through the same point. Let $f(n)$ be the number of regions which the lines split the piece of paper. Compute some values of $f(n)$ & predict its general formula.

Problem 4 ([Sha22], P1.1.2., p. 10). Prove: (a) (Formula of triangle numbers) $\sum_{i=1}^n i = \frac{n(n+1)}{2}$, $\forall n \in \mathbb{N}^*$. (b) (Partial sums of geometric sequences)

$$S_n(a) := \sum_{i=0}^n a^i = \begin{cases} n+1 & \text{if } a = 1, \\ \frac{a^{n+1} - 1}{a - 1} & \text{if } a \neq 1, \end{cases} \quad \forall n \in \mathbb{N}^*.$$

(c) Compute $1 + \sum_{i=1}^n i!$, $\forall n \in \mathbb{N}^*$. (d) Compute $3 \sum_{i=1}^n i(i+1)$, $\forall n \in \mathbb{N}^*$. (e) Compute $\sum_{i=1}^n i^2$, $\forall n \in \mathbb{N}^*$.

Hint. (ii) Compute $aS_n(a) - S_n(a)$.

Remark 3. The sequence of integers $\left\{ \frac{n(n+1)}{2} \right\}_{n=1}^{\infty}$ are called triangle numbers since they count the number of dots in progressively larger triangles.

Bài toán 7 ([Sha22], p. 6). Trên một tờ giấy vuông lớn, vẽ $n \in \mathbb{N}^*$ các đường thẳng bắt đầu từ 1 cạnh của hình vuông & kết thúc ở cạnh còn lại. Mỗi 2 đường thẳng cắt nhau nhưng không có 3 (hoặc nhiều hơn) đường thẳng nào đi qua cùng một điểm. Giả sử $f(n)$ là số vùng mà các đường thẳng chia tờ giấy. Tính một số giá trị của $f(n)$ & dự đoán công thức chung của nó.

Problem 5 ([Sha22], p. 6). You have 100 briefcases numbered 1 through 100. For any $n \in \mathbb{N}^*$, if a briefcase numbered n holds cash, then so does the briefcase numbered $n+3$. You open up briefcase numbered 55 & it has a stuffed animal in it. Can you conclude anything about any of the other briefcases?

Bài toán 8 ([Sha22], p. 6). Bạn có 100 cặp được đánh số từ 1 đến 100. Đối với bất kỳ $n \in \mathbb{N}^*$ nào, nếu một cặp được đánh số n đựng tiền mặt, thì cặp được đánh số $n+3$ cũng vậy. Bạn mở cặp được đánh số 55 & bên trong có một con thú nhồi bông. Bạn có thể kết luận điều gì về bất kỳ cặp nào khác không?

1.6 Principle of strong induction – Nguyên lý quy nạp mạnh

Given an infinite sequence of propositions $\{P_n\}_{n=1}^{\infty} = P_1, P_2, \dots, P_n, \dots$ in order to demonstrate that all of them are true, it is enough to know 2 things:

- The base case: P_1 is true.
- The inductive step: $\forall k \in \mathbb{N}^*$, if P_1, P_2, \dots, P_k are true, then so is P_{k+1} .

1.7 Fibonacci & Lucas numbers

Definition 3 (Fibonacci sequences). Fibonacci sequences are defined by

$$\begin{cases} F_0 = 0, F_1 = 1, \\ F_n = F_{n-1} + F_{n-2}, \quad \forall n \in \mathbb{N}, n \geq 2. \end{cases}$$

Definition 4 (Lucas sequences). The sequence of Lucas numbers are defined by

$$\begin{cases} L_0 = 2, L_1 = 1, \\ L_n = L_{n-1} + L_{n-2}, \quad \forall n \in \mathbb{N}, n \geq 2. \end{cases}$$

1.8 Recurrence Relations – Quan hệ truy hồi/hồi quy/đệ quy

1.9 Pigeonhole principle & Ramsey theory – Nguyên lý chuồng bồ câu & lý thuyết Ramsey

1.10 Counting rules & Stirling number of type 1 & type 2

Problem 6 ([Sha22], p. 2). Given $m, n \in \mathbb{N}^*$, $m \leq n$. Count the number of sequences a_1, a_2, \dots, a_n consisting of m 0's & $n-m$ 1's, if no 2 consecutive terms are both 0's?

Problem 7 ([Sha22], p. 3). Let $f(n)$ be the number of subsets of $[n]$. Prove that $f(n) = 2^n$, $\forall n \in \mathbb{N}^*$.

Problem 8 ([Sha22], p. 3). Assume $n \in \mathbb{N}$, $n \geq 2$, people given their n hats to a hat-check person. Let $f(n)$ be the number of ways that the hats can be returned, so that everyone has 1 hat, but no one has their own hat. Prove: (a) $f(n) = n! \sum_{i=0}^n \frac{(-1)^i}{i!}$, $\forall n \in \mathbb{N}^*$. (b) $f(n)$ is the nearest integer to $\frac{n!}{e}$.

Problem 9 ([Sha22], p. 3). Given $n \in \mathbb{N}^*$. Let $f(n)$ be the number of subsets of $[n]$ that do not contain 2 consecutive integers. (a) Compute $f(1), f(2), f(3), f(4)$. (b) Prove: $f(n) = f(n-1) + f(n-2)$, $\forall n \in \mathbb{N}$, $n \geq 3$. (c) Prove:

$$f(n) = \frac{1}{\sqrt{5}} (\tau^{n+2} - \bar{\tau}^{n+2}), \quad \text{where } \tau = \frac{1 + \sqrt{5}}{2}, \bar{\tau} = \frac{1 - \sqrt{5}}{2}.$$

1.11 Hoán vị & tổ hợp

Bài toán 9 (Consecutive coin toss – Gieo các đồng xu liên tiếp). Cho $n, k \in \mathbb{N}^*$, $k \leq n$. Tung 1 đồng xu đồng chất ngẫu nhiên n lần. Tính xác suất lý thuyết của sự kiện: (a) Toàn bộ đều là mặt sấp (ngửa). (b) Có đúng k lần xuất hiện mặt sấp (ngửa). (c) Có ít nhất k lần xuất hiện mặt sấp (ngửa). (d) Có đúng k lần xuất hiện mặt sấp (ngửa) liên tiếp nhau. (e) Có ít nhất k lần xuất hiện mặt sấp (ngửa) liên tiếp nhau.

Giải. Gọi $X_i \in \{S, N\}$ là biến cố ngẫu nhiên biểu diễn mặt đồng xu trong lần tung thứ i , $\forall i = 1, \dots, n$. Không gian mẫu: $|\Omega| = \prod_{i=1}^n 2 = 2^n$. (a) Vì chỉ có 1 trường hợp thuận lợi là (S, S, \dots, S) nên $\mathbb{P}(X_i = S, \forall i = 1, \dots, n) = \mathbb{P}(\{|i; X_i = S\}| = n) = \frac{1}{2^n}$. Tương tự, vì chỉ có 1 trường hợp thuận lợi là (N, N, \dots, N) nên $\mathbb{P}(X_i = N, \forall i = 1, \dots, n) = \mathbb{P}(\{|i; X_i = N\}| = n) = \frac{1}{2^n}$. (b) $\mathbb{P}(\{|i; X_i = S\}| = k) = \mathbb{P}(\{|i; X_i = N\}| = k) = \frac{C_n^k}{2^n}$, $\forall k = 0, \dots, n$. (c) $\mathbb{P}(\{|i; X_i = S\}| \geq k) = \mathbb{P}(\{|i; X_i = N\}| \geq k) = \frac{C_n^k + C_n^{k+1} + \dots + C_n^n}{2^n} = \frac{\sum_{i=k}^n C_n^i}{2^n}$, $\forall k = 0, \dots, n$. (d) $\mathbb{P} = \frac{n-k+1}{2^n}$. (e) $\mathbb{P} = \frac{\sum_{i=k}^n (n-i+1)}{2^n} = \frac{(n+1)(n-k+1) - \frac{(n+k)(n-k+1)}{2}}{2^n}$. □

Bài toán 10 (Simultaneous coin toss – Gieo các đồng xu đồng thời). Cho $n, k \in \mathbb{N}^*$, $k \leq n$. Tung đồng thời n đồng xu đồng chất ngẫu nhiên. Tính xác suất lý thuyết của sự kiện: (a) Toàn bộ đều là mặt sấp (ngửa). (b) Có đúng k lần xuất hiện mặt sấp (ngửa). (c) Có ít nhất k lần xuất hiện mặt sấp (ngửa).

Giải. Gọi X là biến cố ngẫu nhiên chỉ số mặt S xuất hiện khi tung đồng thời n đồng xu. (a) $\mathbb{P}(X = n) = \mathbb{P}(X = 0) = \frac{1}{n+1}$. (b) $\mathbb{P}(X = k) = \frac{1}{n+1}$ □

Bài toán 11 (Consecutive 2 dice rolls – Gieo 2 xúc xắc lần lượt). Gieo lần lượt 2 con xúc xắc. Tính xác suất lý thuyết của sự kiện: (a) 2 mặt có cùng số chấm, khác số chấm. (b) Số chấm 2 mặt có cùng tính chẵn lẻ, khác tính chẵn lẻ. (c) Số chấm 2 mặt đều là số nguyên tố, đều là hợp số, có ít nhất 1 số nguyên tố, có ít nhất 1 hợp số. (d) Số chấm 1 mặt là ước (bội) của số chấm trên mặt còn lại. (e) Tổng số chấm 2 mặt bằng $n \in \mathbb{N}$.

Ans. (e) $f(n) = (\min\{n-1, 6\} - \max\{n-6, 1\} + 1)\mathbf{1}_{n \in \{2, 3, \dots, 12\}}$.

Bài toán 12 (Simultaneous 2 dice rolls – Gieo 2 xúc xắc đồng thời). Gieo đồng thời 2 con xúc xắc. Tính xác suất lý thuyết của sự kiện: (a) 2 mặt có cùng số chấm, khác số chấm. (b) Số chấm 2 mặt có cùng tính chẵn lẻ, khác tính chẵn lẻ. (c) Số chấm 2 mặt đều là số nguyên tố, đều là hợp số, có ít nhất 1 số nguyên tố, có ít nhất 1 hợp số. (d) Số chấm 1 mặt là ước (bội) của số chấm trên mặt còn lại. (e) Tổng số chấm 2 mặt bằng $n \in \mathbb{N}$.

Bài toán 13 (Consecutive n dice rolls – Gieo n xúc xắc lần lượt). Gieo lần lượt $n \in \mathbb{N}^*$ con xúc xắc. Tính xác suất lý thuyết của sự kiện: (a) n mặt có cùng số chấm. (b) n mặt có khác số chấm. (c) Số chấm n mặt có cùng tính chẵn lẻ. (d) Số chấm 1 mặt là ước (bội) của số chấm trên các mặt còn lại. (e) Tổng số chấm n mặt bằng $a \in \mathbb{N}$.

Bài toán 14 (Simultaneous n dice rolls – Gieo n xúc xắc đồng thời). Gieo đồng thời $n \in \mathbb{N}^*$ con xúc xắc. Tính xác suất lý thuyết của sự kiện: (a) n mặt có cùng số chấm. (b) n mặt có khác số chấm. (c) Số chấm n mặt có cùng tính chẵn lẻ. (d) Số chấm 1 mặt là ước (bội) của số chấm trên các mặt còn lại. (e) Tổng số chấm n mặt bằng $a \in \mathbb{N}$.

Bài toán 15 (Squares & rectangles with same perimeter – Hình vuông & hình chữ nhật cùng chu vi). Cho $n \in \mathbb{N}^*$. Viết n thành tổng 2 số: $n = a + b$. Tính xác suất để a, b cùng là độ dài cạnh của 1 hình vuông, xác suất để a, b là độ dài 2 cạnh của 1 hình chữ nhật nếu: (a) $a, b \in \mathbb{N}^*$. (b) $a, b \in \mathbb{N}$.

Bài toán 16 (Squares & rectangles with same area – hình vuông & hình chữ nhật cùng diện tích). Cho $a \in \mathbb{N}^*$, $a \geq 2$ có phân tích thừa số nguyên tố $a = \prod_{i=1}^n p_i^{a_i} = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$ với p_i là số nguyên tố, $a_i \in \mathbb{N}^*$, $\forall i = 1, 2, \dots, n$. (a) Viết ngẫu nhiên a thành tích của 2 số: $a = bc$. Tính xác suất để b, c là độ dài 2 cạnh của 1 hình chữ nhật, xác suất để b, c cùng là độ dài cạnh của 1 hình vuông nếu: (i) $b, c \in \mathbb{N}$. (ii) $b, c \in \mathbb{Z}$. (b) Lấy ngẫu nhiên 2 số $b, c \in \mathcal{U}(a)$. Tính xác suất để phân số $\frac{b}{c}$: (i) tối giản. (ii) không tối giản.

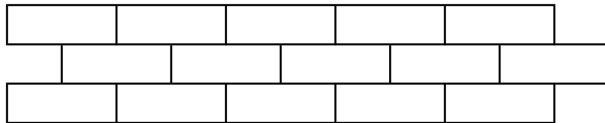
Definition 5 (Prime-counting function). The prime-counting function is the function counting the number of prime numbers less than or equal to some real number x , denoted by $\pi(x) := |\{p \in \mathbb{N}^* | p \text{ is a prime, } p \leq x\}|$.

Định nghĩa 3 (Hàm đếm số số nguyên tố). Hàm đếm số số nguyên tố là hàm đếm số số nguyên tố nhỏ hơn hoặc bằng $x \in \mathbb{R}$, ký hiệu là $\pi(x) := |\{p \in \mathbb{N}^* | p \text{ là số nguyên tố, } p \leq x\}|$.

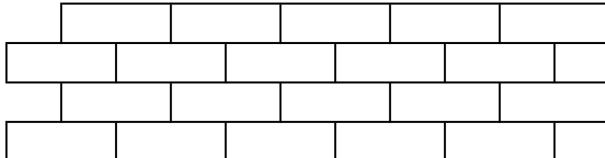
Bài toán 17 (Prime, composite – số nguyên tố, hợp số). Cho $m, n, k \in \mathbb{N}^*$. Đặt $A_n = \{1, 2, \dots, n\}$ là tập hợp n số nguyên dương đầu tiên, $\forall n \in \mathbb{N}^*$. (a) Lấy m số từ A_n . Tính xác suất để m số này cùng chẵn, cùng lẻ, có ít nhất 1 số chẵn, có ít nhất 1 số lẻ, có đúng k số chẵn, có đúng k số lẻ, có ít nhất k số chẵn, có ít nhất k số lẻ. (b) Lấy m số phân biệt từ A_n . Tính xác suất để m số này đều là số nguyên tố, đều là hợp số, có đúng k số nguyên tố, có đúng k hợp số, có ít nhất 1 số nguyên tố, có ít nhất 1 hợp số, có ít nhất k số nguyên tố, có ít nhất k hợp số. (c) Viết chương trình Pascal, Python C/C++ để mô phỏng việc tính các xác suất đó.

Bài toán 18 (Odd, even – chẵn, lẻ). Cho $a, b \in \mathbb{Z}$, $a < b$, $n, k \in \mathbb{N}^*$, $n \geq 2$, $k \leq n$. Đặt $A = [a, b] \cap \mathbb{Z} = \{a, a+1, a+2, \dots, b-1, b\}$.
(a) Lấy 2 số từ tập A . Xét 2 trường hợp phân biệt, không nhất thiết phân biệt. Tính xác suất để 2 số này cùng tính chẵn lẻ, khác tính chẵn lẻ. (b) Lấy n số từ tập A . Tính xác suất để n số này đều chẵn, đều lẻ, cùng tính chẵn lẻ, có đúng k số chẵn, k số lẻ, có ít nhất k số chẵn, k số lẻ. (c) Viết chương trình Pascal, Python C/C++ để mô phỏng việc tính các xác suất đó.

Bài toán 19 (VMC2024B4). (a) Đếm số cách chọn ra 3 viên gạch, mỗi viên từ 1 hàng trong 3×5 viên gạch xếp xen kẽ, sao cho không có 2 viên gạch nào được lấy ra nằm kề nhau (2 viên gạch được gọi là kề nhau nếu có chung 1 phần của 1 cạnh).



(b) Đếm số cách chọn ra 4 viên gạch, mỗi viên từ 1 hàng trong 4×5 viên gạch xếp xen kẽ, sao cho không có 2 viên gạch nào được lấy ra nằm kề nhau.



(c) Cho $m, n \in \mathbb{N}^*$. Đếm số cách chọn ra m viên gạch, mỗi viên từ 1 hàng trong $m \times n$ viên gạch xếp xen kẽ, sao cho không có 2 viên gạch nào được lấy ra nằm kề nhau. (d) Cho $m, n, k \in \mathbb{N}^*$. Đếm số cách chọn ra k viên gạch, không nhất thiết mỗi viên từ 1 hàng trong $m \times n$ viên gạch xếp xen kẽ, sao cho không có 2 viên gạch nào được lấy ra nằm kề nhau. (e*) Mở rộng cho trường hợp $m \times n$ với số gạch mỗi hàng có thể khác nhau, cụ thể là hàng i chứa $a_i \in \mathbb{N}^*$ viên gạch, $\forall i = 1, \dots, m$ với 2 trường hợp: (i) Mỗi viên từ 1 hàng. (ii) Lấy $k \in \mathbb{N}^*$ viên gạch, mỗi hàng có thể lấy nhiều viên.

Nhận xét 1 (Left-right symmetry – Đối xứng trái phải). Nếu số viên gạch của mỗi hàng bằng nhau & được sắp xếp xen kẽ như (a) & (b), thì thứ tự viên gạch đầu tiên từ bên trái của mỗi hàng lòi ra hay thụt vào không quan trọng, vì có thể lấy đối xứng gương trái-phải để chuyển đổi 2 trường hợp đó. Cũng chú ý đến tính đối xứng trên-dưới (top-bottom symmetry).

Chứng minh. Số cách chọn gạch từ 3 hàng, mỗi hàng n viên gạch: $(n-1)(n-2)^2 + (n-1)^2 = (n-1)(n^2 - 3n + 3)$, $\forall n \in \mathbb{N}^*$, $n \geq 2$. Số cách chọn gạch từ 4 hàng, mỗi hàng $n \in \mathbb{N}^*$ viên gạch: $(n^2 - 3n + 3)^2$, $\forall n \in \mathbb{N}^*$, $n \geq 2$. \square

• C++ codes:

- (DKAK): https://github.com/NQBH/advanced_STEM_beyond/blob/main/VMC/C++/brick_DPAK.cpp.
- (NLDK): https://github.com/NQBH/advanced_STEM_beyond/blob/main/VMC/C++/brick_NLDK.cpp.

1.12 Hệ số nhị thức & đa thức

1.13 Phân vùng số nguyên & nguyên tắc loại suy

1.14 Generating function – Hàm sinh

Idea of generating functions. If we want to find a formula for some function $f(n)$ with $n \in \mathbb{N}$, then we can form the generating function of $f(n)$:

$$F(x) := \sum_{n \geq 0} f(n)x^n = f(0) + f(1)x + f(2)x^2 + \dots + f(n)x^n + \dots, \quad (\text{genf})$$

so that a_n is the coefficient of x^n in the Taylor series expansion of $F(x)$ defined by (genf).

Here we are concerned with formal power series, & questions of convergence do not come up (at least for elementary applications). Sometimes this power series has a nice closed form & then we can manipulate this function & get information about $f(n)$, $\forall n \in \mathbb{N}$.

– Ở đây chúng ta quan tâm đến chuỗi lũy thừa chính thức, & các câu hỏi về sự hội tụ không xuất hiện (ít nhất là đối với các ứng dụng cơ bản). Đôi khi chuỗi lũy thừa này có dạng đẹp & sau đó chúng ta có thể thao tác hàm này & lấy thông tin về $f(n)$, $\forall n \in \mathbb{N}$.

Problem 10 ([Sha22], p. 4). Let $\{a_n\}_{n=0}^\infty$ be a sequence defined by

$$a_0 = 1, \quad \sum_{i=0}^n a_i a_{n-i} = 1, \quad \forall n \in \mathbb{N}^*.$$

(a) Let $F(x) := \sum_{i=0}^\infty a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots$ be the generating function of $\{a_n\}_{n=0}^\infty$. Prove: $F(x)F(x) = \frac{1}{1-x}$, thus

$F(x) = \frac{1}{\sqrt{1-x}}$. (b) Prove: $a_n = \frac{(2n-1)!!}{2^n n!} = \frac{1 \cdot 3 \cdot 5 \cdot (2n-1)}{2^n n!}$, $\forall n \in \mathbb{N}^*$.

2 Graph Theory – Lý Thuyết Đồ Thị

Resources – Tài nguyên.

- [AD10]. TITU ANDREESCU, GABRIEL DOSPINESCU. *Problems From the Book*. Chap. 6: *Some Classical Problems in Extremal Graph Theory – Vài Bài Toán Cổ Điển trong Lý Thuyết Đồ Thị Cực Trị*, pp. 119–136.
- [Val02; Val21]. GABRIEL VALIENTE. *Algorithms on Trees & Graphs With Python Code*. 2e.

“In mathematics & computer science, *graph theory* is the study of **graphs**, which are **mathematical structures** used to model pairwise relations between objects. A graph in this context is made up of **vertices** (also called *nodes* or *points*) which are connected by **edges** (also called *arcs*, *links* or *lines*). A distinction is made between *undirected graphs*, where edges link 2 vertices symmetrically, & *directed graphs*, where edges link 2 vertices asymmetrically. Graphs are 1 of the principal objects of study in **discrete mathematics**.” – [Wikipedia/graph theory](#).

Graphs are such a good way of organizing certain kinds of information & formulating problems in discrete mathematics in general & combinatorics & related fields of mathematics in particular. Intuitively, a graph is a set of dots – called *vertices* – where some of the dots are connected to some of the other dots. The connections are called *edges*. Each edge of a graph connects just 2 vertices & so it can be represented as a pair of vertices.

2.1 Trees & graphs: Some basic concepts – Cây & đồ thị: Vài khái niệm cơ bản

Definitions in graph theory vary. For a comprehensive list of concepts in graph theory, see, e.g., [Wikipedia/glossary of graph theory](#). A graph consists of a set of vertices and a set of edges. Intuitively (and visually), an edge is a direct connection between two vertices. More formally, an edge is a set consisting of a pair of vertices.

The notion of graph which is most useful in computer science is that of a directed graph 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.

– Khái niệm đồ thị hữu ích nhất trong khoa học máy tính là đồ thị có hướng hoặc chỉ là đồ thị. Đồ thị là 1 cấu trúc tổ hợp bao gồm 1 tập hợp hữu hạn không rỗng các đối tượng, được gọi là các đỉnh, cùng với 1 tập hợp hữu hạn (có thể rỗng) các cặp đỉnh có thứ tự, được gọi là các cạnh có hướng hoặc cung.

Definition 6 (Graphs & subgraphs, [Sha22], Def. 2.17, p. 54). *A simple graph (or just a graph) G is a pair of sets (V, E) where V is a nonempty set called the set of vertices of G , & E is a (possibly empty) set of unordered pairs of distinct elements of V . The set E is called the set of edges of G . If the number of vertices of G is finite, then G is a finite graph (or a finite simple graph).*

If $G = (V, E)$ & $H = (V', E')$ are both graphs, then H is a subgraph of G if $V' \subseteq V$ & $E' \subseteq E$.

Bài toán 20 (+0.5). *Vẽ đồ thị Peterson $G_{\text{Peterson}} = (V, E)$ & mô tả cụ thể V, E .*

To construct a subgraph of a graph given, we take some of the vertices & some of the edges, but we have to make sure that we have included all the vertices of the edges that we chose.

Definition 7 (Directed graph, [Val21], Def. 1.1, p. 3). *A graph $G = (V, E)$ consists of a finite nonempty set V of vertices & a finite set $E \subseteq V \times V$ of edges. The order of a graph $G = (V, E)$, denoted by n , is the number of vertices, $n = |V|$ & the size, denoted by m , is the number of edges, $m = |E|$. An edge $e = (v, w)$ is said to be incident with vertices v & w , where v is the source & w the target of edge e , & vertices v, w are said to be adjacent. Edges $(u, v), (v, w)$ are said to be adjacent, as are edges $(u, v), (w, v)$, & also edges $(v, u), (v, w)$.*

Định nghĩa 4 (Đồ thị có hướng). *1 đồ thị $G = (V, E)$ bao gồm 1 tập hữu hạn không rỗng V các đỉnh & 1 tập hữu hạn $E \subseteq V \times V$ các cạnh. Bậc của 1 đồ thị $G = (V, E)$, ký hiệu là n , là số đỉnh, $n = |V|$ & size, ký hiệu là m , là số cạnh, $m = |E|$. Một cạnh $e = (v, w)$ được gọi là incident với các đỉnh v & w , trong đó v là source & w target của cạnh e , & các đỉnh v, w được gọi là kề. Các cạnh $(u, v), (v, w)$ được gọi là kề, cũng như các cạnh $(u, v), (w, v)$, & cũng vậy các cạnh $(v, u), (v, w)$.*

Graphs are often drawn as a set of points in the plane & a set of arrows, each of which joins 2 (not necessarily different) points. In a drawing of a graph $G = (V, E)$, each vertex $v \in V$ is drawn as a point or a small circle & each edge $(v, w) \in E$ is drawn as an arrow from point or circle of vertex v to the point or circle corresponding to vertex w .

– Đồ thị thường được vẽ như một tập hợp các điểm trên mặt phẳng & một tập hợp các mũi tên, mỗi mũi tên nối 2 điểm (không nhất thiết phải khác nhau). Trong bản vẽ đồ thị $G = (V, E)$, mỗi đỉnh $v \in V$ được vẽ như một điểm hoặc một đường tròn nhỏ & mỗi cạnh $(v, w) \in E$ được vẽ như một mũi tên từ điểm hoặc đường tròn của đỉnh v đến điểm hoặc đường tròn tương ứng với đỉnh w .

A vertex has 2 degrees in a graph, one given by the number of edges coming into the vertex & the other given by the number of edges in the graph going out of the vertex.

– Mỗi đỉnh có 2 bậc trong đồ thị, một bậc được xác định bởi số cạnh đi vào đỉnh & bậc còn lại được xác định bởi số cạnh trong đồ thị đi ra khỏi đỉnh.

Definition 8 ([Val21], Def. 1.2, p. 4). *The indegree of a vertex v in a graph $G = (V, E)$ is the number of edges in G whose target is v , i.e., $\text{indeg}(v) = |\{(u, v) | (u, v) \in E\}|$. The outdegree of a vertex v in a graph $G = (V, E)$ is the number of edges in G whose source is v , i.e., $\text{outdeg}(v) = |\{(v, w) | (v, w) \in E\}|$. The degree of a vertex v in a graph $G = (V, E)$ is the sum of the indegree & the outdegree of the vertex, i.e., $\text{deg}(v) = \text{indeg}(v) + \text{outdeg}(v)$.*

A basic relationship between the size of a graph & the degree of its vertices, which will prove to be very useful in analyzing the computational complexity of algorithms on graphs:

– Mỗi quan hệ cơ bản giữa kích thước của đồ thị & bậc của các đỉnh, sẽ rất hữu ích trong việc phân tích độ phức tạp tính toán của các thuật toán trên đồ thị:

Theorem 5. *Let $G = (V, E)$ be a graph with n vertices & m edges, & let $V = \{v_1, \dots, v_n\}$. Then*

$$\sum_{i=1}^n \text{indeg}(v_i) = \sum_{i=1}^n \text{outdeg}(v_i) = m.$$

Definition 9 (Multigraphs, general graphs, [Sha22], Def. 2.20, p. 54). *In the definition of a graph, if we allow E to be a multiset – i.e., allow repeated edges – then G is called a multigraph or a graph with repeated edges. If we also allow pairs of non-distinct elements in E – i.e., allow loops – then G is called a general graph or a graph with repeated edges & loops.*

Định nghĩa 5 (Đa đồ thị, đồ thị tổng quát). *Trong định nghĩa của đồ thị, nếu chúng ta cho phép E là một đa tập – tức là cho phép các cạnh lặp lại – thì G được gọi là đa đồ thị hoặc đồ thị có các cạnh lặp lại. Nếu chúng ta cũng cho phép các cặp phần tử không phân biệt trong E – tức là cho phép các vòng lặp – thì G được gọi là đồ thị tổng quát hoặc đồ thị có các cạnh lặp lại & vòng lặp.*

If a graph G does not have any double edges or loops then G is a simple graph. If a graph G has repeated edges but no loops, then G is a multigraph. If G has both repeated edges & repeated loops, then G would be a general graph.

Remark 4 ([Sha22], Rmk. 2.22, p. 55). *If $G = (V, E)$ is a general graph & $e \in E$ is an edge of G , then, unlike in the case of simple graphs, we cannot necessarily identify e by its vertices. I.e., it will be inadequate to say $e = \{v, w\}$, where $v, w \in V$. This is because there may be multiple edges connecting v & w . Hence, to be formally correct, we should say that a general graph consists of 2 sets V & E & an incidence map that gives 2 vertices for each $e \in E$. I.e., every edge will have its own name in addition to being associated with its 2 vertices. To avoid unnecessary clutter, in what follows we will continue to identify edges of general graphs as pairs of vertices unless there is a need to be more formal.*

– Nếu $G = (V, E)$ là một đồ thị tổng quát & $e \in E$ là một cạnh của G , thì, không giống như trong trường hợp của đồ thị đơn giản, chúng ta không nhất thiết phải xác định e theo các đỉnh của nó. Tức là, sẽ không đủ nếu nói $e = \{v, w\}$, trong đó $v, w \in V$. Điều này là do có thể có nhiều cạnh kết nối v & w . Do đó, để chính xác về mặt hình thức, chúng ta phải nói rằng một đồ thị tổng quát bao gồm 2 tập hợp V & E & một bản đồ tới cung cấp 2 đỉnh cho mỗi $e \in E$. Tức là, mỗi cạnh sẽ có tên riêng của nó ngoài việc được liên kết với 2 đỉnh của nó. Để tránh sự lộn xộn không cần thiết, trong phần sau, chúng ta sẽ tiếp tục xác định các cạnh của đồ thị tổng quát là các cặp đỉnh trừ khi cần phải chính thức hơn.

Definition 10 (Order, adjacency, incidence, [Sha22], Def. 2.23, p. 55). *Let $G = (V, E)$ be a general graph. Then $|V|$, the number of vertices, is called the order of G . If $\alpha = \{x, y\} \in E$ – i.e., α is an edge connecting the vertices x, y – then we say that x & y are vertices or ends of α , that α joints x & y , that x & y are adjacent, & that x & α are incident.*

Definition 11 (Degree of a vertex, [Sha22], Def. 2.24, p. 55). *If x is a vertex of a graph G , then the number of edges incident with x is called the degree (or the valence) of x . In the case of general graphs, a loop at a vertex x contributes 2 to the degree of x .*

The degree of a vertex is the number of edges incident with it & note that, in a general graph, a loop contributes 2 to the degree of its vertex. One usually records the degrees of each vertices of a graph in a table.

– Bậc của một đỉnh là số cạnh liên quan đến nó & lưu ý rằng, trong một đồ thị tổng quát, một vòng lặp đóng góp 2 vào bậc của đỉnh của nó. Người ta thường ghi lại bậc của mỗi đỉnh của đồ thị trong một bảng.

Roughly speaking, in graph theory, we are not concerned with the labels of the vertices. So, if we keep the vertices & edges intact, but rearrange the names of the vertices, we get a new graph that is basically the same graph as G . We say that these 2 graphs are *isomorphic*.

– Nói một cách đại khái, trong lý thuyết đồ thị, chúng ta không quan tâm đến nhãn của các đỉnh. Vì vậy, nếu chúng ta giữ nguyên các đỉnh & cạnh, nhưng sắp xếp lại tên của các đỉnh, chúng ta sẽ có một đồ thị mới về cơ bản là cùng một đồ thị với G . Chúng ta nói rằng 2 đồ thị này là *đẳng cấu*.

More technically, in combinatorial graph theory – as opposed to topological graph theory – we are not concerned with how a graph is drawn. The drawing is just a representation that helps us “see” the vertices & the edges. All that matters is the number of vertices & their adjacencies. Hence, we consider 2 graphs the same if, after we appropriately relabel the vertices of 1 of the graphs, the set of edges of the 2 graphs become the same. 2 such graphs are called isomorphic.

Definition 12 (Graph isomorphism). *Let G, H be graphs. The graph G is isomorphic to the graph H if there exists a 1-1, onto map $f : V(G) \rightarrow V(H)$ s.t. $\{x, y\}$ is an edge of G iff $\{f(x), f(y)\}$ is an edge of H . Such a map f is called an isomorphism between G & H .*

Example 2. *In the Peterson graph, every vertex has degree 3.*

A few oft-used terms to graph vocabularies.

Definition 13 (Isolated vertex, leaf, [Sha22], Def. 10.2, p. 362). *Let G be a graph. A vertex with degree 0 is called an isolated vertex, while a vertex with degree 1 is called a leaf.*

Definition 14 (Regular graph, [Sha22], Def. 10.3, p. 362). A graph is called regular of degree d or d -regular if each vertex has degree equal to d .

In mathematical notation:

$$G = (V, E) \text{ is a } d\text{-regular} \Leftrightarrow \deg v = d, \forall v \in V.$$

Definition 15 (Cubic graph, [Sha22], Def. 10.4, p. 362). A graph is called cubic if it is regular of degree 3 (i.e., if each vertex has degree equal to 3).

2.1.1 Walks, trails, paths, cycles, & complete graphs

Walks, trails, & paths in a graph are alternating sequences of vertices & edges in the graph s.t. each edge in the sequence is preceded by its source vertex & followed by its target vertex. Trails are walks having no repeated edges, & paths are trails having no repeated vertices.

– Đường đi, đường mòn, & đường đi trong đồ thị là chuỗi xen kẽ các đỉnh & cạnh trong đồ thị, tức là mỗi cạnh trong chuỗi được đi trước bởi đỉnh nguồn & theo sau bởi đỉnh đích. Đường mòn là đường đi không có cạnh lặp lại, & đường đi là đường mòn không có đỉnh lặp lại.

Definition 16 (Walk, trail, path, [Val21], Def. 1.3). A walk from vertex v_i to vertex v_j in a graph is an alternating sequence $[v_i, e_{i+1}, v_{i+1}, e_{i+2}, \dots, v_{j-1}, e_j, v_j]$ of vertices & edges in the graph, s.t. $e_k = (v_{k-1}, v_k)$ for $k = i + 1, \dots, j$. A trail is a walk with no repeated edges, & a path is a trail with no repeated vertices (except, possibly, the initial & final vertices). The length of a walk, trail, or path is the number of edges in the sequence.

Định nghĩa 6 (Đường đi dạo, đường mòn, đường đi). 1 đường đi dạo từ đỉnh v_i đến đỉnh v_j trong một đồ thị là một chuỗi xen kẽ $[v_i, e_{i+1}, v_{i+1}, e_{i+2}, \dots, v_{j-1}, e_j, v_j]$ các đỉnh & cạnh trong đồ thị, s.t. $e_k = (v_{k-1}, v_k)$ với $k = i + 1, \dots, j$. Một đường mòn là một cuộc đi bộ không có cạnh nào lặp lại, & một đường đi là một cuộc đi bộ không có đỉnh nào lặp lại (ngoại trừ, có thể là, các đỉnh đầu & cuối). Độ dài của một cuộc đi bộ, trail hoặc path là số cạnh trong chuỗi.

Since an edge in a graph is uniquely determined by its source & target vertices, a walk, trail, or path can be abbreviated by just enumerating either the vertices $[v_i, v_{i+1}, \dots, v_{j-1}, v_j]$ or the edges $[e_{i+1}, e_{i+2}, \dots, e_j]$ in the alternating sequence $[v_i, e_{i+1}, v_{i+1}, e_{i+2}, \dots, v_{j-1}, e_j, v_j]$ of vertices & edges.

– Vì một cạnh trong đồ thị được xác định duy nhất bởi các đỉnh nguồn & đích của nó, nên một đường đi, đường mòn hoặc đường dẫn có thể được rút gọn chỉ bằng cách liệt kê các đỉnh $[v_i, v_{i+1}, \dots, v_{j-1}, v_j]$ hoặc các cạnh $[e_{i+1}, e_{i+2}, \dots, e_j]$ trong chuỗi xen kẽ $[v_i, e_{i+1}, v_{i+1}, e_{i+2}, \dots, v_{j-1}, e_j, v_j]$ các đỉnh & cạnh.

Walks are closed if their initial & final vertices coincide. – Đường đi sẽ khép lại nếu đỉnh đầu & đỉnh cuối trùng nhau.

Definition 17 (Cycle, [Val21], Def. 1.4). A walk, trail, or path $[v_i, e_{i+1}, v_{i+1}, e_{i+2}, \dots, v_{j-1}, e_i, v_j]$ is said to be closed if $v_i = v_j$. A cycle is a closed path of length at least 1.

The combinatorial structure of a graph encompasses 2 notions of the substructure. A subgraph of a graph is just a graph whose vertex & edge sets are contained in the vertex & edge sets of the given graph, resp. The subgraph of a graph induced by a subset of its vertices has as edges the set of edges in the given graph whose source & target belong to the subset of vertices.

– Cấu trúc tổ hợp của một đồ thị bao gồm 2 khái niệm về cấu trúc con. Một đồ thị con của một đồ thị chỉ là một đồ thị có tập đỉnh & cạnh được chứa trong tập đỉnh & cạnh của đồ thị đã cho, tương ứng. Đồ thị con của một đồ thị được tạo ra bởi một tập con các đỉnh của nó có các cạnh là tập các cạnh trong đồ thị đã cho có nguồn & đích thuộc về tập con các đỉnh.

Definition 18 (Subgraph, [Val21], Def. 1.5, p. 6). Let $G = (V, E)$ be a graph, & let $W \subseteq V$. A graph (W, S) is a subgraph of G if $S \subseteq E$. The subgraph of G induced by W is the graph $(W, E \cap W \times W)$.

Định nghĩa 7 (Đồ thị con). Cho $G = (V, E)$ là một đồ thị, & cho $W \subseteq V$. Một đồ thị (W, S) là một đồ thị con của G nếu $S \subseteq E$. Đồ thị con của G được tạo ra bởi W là đồ thị $(W, E \cap W \times W)$.

The notion of graph which is most often found in mathematics is that of an undirected graph. Unlike the directed edges or edges of a graph, edges of an undirected graph have no direction association with them & therefore, no distinction is made between the source & target vertices of an edge. In a mathematical sense, an undirected graph consists of a set of vertices & a finite set of undirected edges, where each edge has a set of 1 or 2 vertices associated with it. In the computer science view of undirected graphs, though, an undirected graph is the particular case of a directed graph in which for every edge (v, w) of the graph, the reversed edge (w, v) also belongs to the graph. Undirected graphs are also called *bidirected*.

– Khái niệm đồ thị thường thấy nhất trong toán học là đồ thị vô hướng. Không giống như các cạnh hoặc cạnh có hướng của đồ thị, các cạnh của đồ thị vô hướng không có liên kết hướng nào với chúng & do đó, không có sự phân biệt nào được tạo ra giữa các đỉnh nguồn & đích của một cạnh. Theo nghĩa toán học, đồ thị vô hướng bao gồm một tập hợp các đỉnh & một tập hợp hữu hạn các cạnh vô hướng, trong đó mỗi cạnh có một tập hợp gồm 1 hoặc 2 đỉnh được liên kết với nó. Tuy nhiên, theo quan điểm khoa học máy tính về đồ thị vô hướng, đồ thị vô hướng là trường hợp cụ thể của đồ thị có hướng trong đó đối với mọi cạnh (v, w) của đồ thị, cạnh đảo ngược (w, v) cũng thuộc về đồ thị. Đồ thị vô hướng cũng được gọi là *song hướng*.

Definition 19 (Undirected graph, [Val21], Def. 1.5, p. 6). A graph $G = (V, E)$ is undirected if $(v, w) \in E \Rightarrow (w, v) \in E$, $\forall v, w \in V$.

Định nghĩa 8 (Đồ thị vô hướng). Đồ thị $G = (V, E)$ là vô hướng nếu $(v, w) \in E \Rightarrow (w, v) \in E$, $\forall v, w \in V$.

Bài toán 21 ([Sha22], p. 361). A soccer ball is often tiled with 12 pentagons & 20 hexagons. Is there anything special about those numbers? If you tile a soccer ball with pentagons & hexagons, what are the possibilities for the number of pentagons & the number of hexagons?

– 1 quả bóng đá thường được lát bằng 12 hình ngũ giác & 20 hình lục giác. Có điều gì đặc biệt về những con số đó không? Nếu bạn lát một quả bóng đá bằng hình ngũ giác & hình lục giác, thì khả năng có bao nhiêu hình ngũ giác & số hình lục giác?

2.1.2 Graphic sequences

Problem 11 ([Sha22], Warm-Up 10.5, p. 363). Is there a simple graph with 4 vertices s.t. the degrees of the vertices are 3, 2, 1, 1/ What if the degrees were 2, 2, 1, 1?

Definition 20 (Degree sequence of a graph; graphic sequences, [Sha22], Def. 10.6, p. 363). The degree sequence of a graph is the list of the degrees of its vertices in non-increasing order.

A non-increasing sequence of nonnegative integers is called graphic if there exists a simple graph whose degree sequence is precisely that sequence.

Question 1. Given a non-increasing sequence of nonnegative integers, when is the sequence graphic?

– Với một dãy số nguyên không âm không tăng, khi nào thì dãy số này là dãy đồ họa?

Problem 12 ([Sha22], Ques. 10.8, p. 363). Which of the following sequences are graphic? (a) 7, 5, 5, 4, 3, 3, 3, 3, 2. (b) 1, 1, 1. (c) 5, 5, 4, 3, 3, 2, 2, 1. (d) 7, 5, 5, 4, 3, 2, 2, 0. (e) 6, 6, 6, 6, 4, 3, 3, 0. (f) 7, 6, 5, 5, 4, 4, 2, 1.

Theorem 6 ([Sha22], Thm. 10.9, p. 363). Let $G = (V, E)$ be a general graph. Let $\{d_i\}_{i=1}^{|V|}$ be the degrees of the vertices. Then

$$\sum_{i=1}^{|V|} d_i = 2|E|.$$

In particular, the number of vertices of G with odd degree is even.

Theorem 7 (Havel–Hakimi algorithm). Consider the following 2 sequences of nonnegative integers. Assume the 1st sequence is in nonincreasing order & $t_s \geq 1$: (a) $s, t_1, \dots, t_s, d_1, \dots, d_n$. (b) $t_1 - 1, t_2 - 1, \dots, t_s - 1, d_1, \dots, d_n$. Sequence (a) is graphic iff sequence (b) is graphic (after possibly rearranging it to make it non-decreasing).

Because of this theorem, if you are given a sequence of nonnegative integers & want to know if it is graphic or not, you repeatedly use Thm. 7 to reduce your sequence to simpler sequences. If, at any point, the simpler sequence is graphic (or not graphic), then so is the original sequence.

– Vì định lý này, nếu bạn được cung cấp một chuỗi các số nguyên không âm & muốn biết nó có đồ họa hay không, bạn sử dụng Thm. 7 nhiều lần để rút gọn chuỗi của bạn thành các chuỗi đơn giản hơn. Nếu, tại bất kỳ thời điểm nào, chuỗi đơn giản hơn là đồ họa (hoặc không đồ họa), thì chuỗi ban đầu cũng vậy.

2.1.3 Miscellaneous: Graph theory

Denote by $d(V)$, $C(V)$ the number, & the set of vertices adjacent to a vertex V , respectively. A graph is said to have a *complete k -subgraph* if there are k vertices any 2 of which are connected. A graph is said to be *k -free* if it does not contain a complete k -subgraph.

Lemma 1 ([AD10], Example 1, p. 121, Zarankiewicz’s lemma). If G is a k -free graph, then there exists a vertex having degree at most $\left\lfloor \frac{k-2}{k-1}n \right\rfloor$.

Zarankiewicz’s lemma is the main step in the proof of Turan’s theorem – a famous classical result about k -free graphs.

Theorem 8 ([AD10], Example 2, p. 123, Turan’s theorem). The greatest number of edges of a k -free graph with n vertices is

$$\frac{k-2}{k-1} \cdot \frac{n^2 - r^2}{2} + \binom{r}{2},$$

where r is the remainder left by n when divided to $k-1$.

Định nghĩa 9 ([HT24], Def. 7.2, p. 249, Đỉnh cô lập, lá). Cho G là 1 đồ thị. Đỉnh có bậc 0 được gọi là đỉnh cô lập, đỉnh có bậc 1 được gọi là lá.

Định nghĩa 10 ([HT24], Def. 7.3, p. 249, Đồ thị chính quy). 1 đồ thị được gọi là chính quy bậc d hoặc d -chính quy nếu mỗi đỉnh có bậc bằng $d \in \mathbb{N}$.

Định nghĩa 11 ([HT24], Def. 7.3, p. 249, Đỉnh thị khối). 1 đồ thị được gọi là đồ thị bậc 3 nếu nó chính quy bậc 3, i.e., mỗi đỉnh đồ thị có bậc bằng 3.

Goal 1 (Tính khả dĩ của dãy bậc của đồ thị). Tìm vài dấu hiệu hoặc vài điều kiện cần & đủ để có thể quyết định liệu 1 dãy số nguyên dương $(a_i)_{i=1}^n \subset \mathbb{N}$ cho trước có thể thể là dãy bậc của đồ thị mà không phải vẽ biểu đồ.

Định nghĩa 12 ([HT24], Def. 7.6, p. 249, Dãy bậc của đồ thị, chuỗi đồ thị). Chuỗi bậc của đồ thị là dãy bậc của các đỉnh của nó theo thứ tự không tăng. 1 dãy số nguyên không âm không tăng được gọi là đồ thị nếu tồn tại 1 đồ thị có chuỗi bậc chính xác là dãy số nguyên không âm đó.

Ví dụ 1 (Sequence 1, 1, ..., 1). 1, 1, 1 không phải là 1 dãy đồ thị vì không thể xây dựng 1 đồ thị có 3 đỉnh sao cho tất cả 3 bậc là 1. Nhưng 1, 1 & 1, 1, 1, 1, hay nói chung các dãy chỉ toàn số 1 với độ dài là 1 số chẵn, i.e., $\{1\}_{i=1}^{2n}, \forall n \in \mathbb{N}^*$, là các dãy đồ thị, nhưng bất kỳ dãy chỉ toàn số 1 với độ dài là 1 số lẻ, i.e., $\{1\}_{i=1}^{2n+1}, \forall n \in \mathbb{N}^*$, thì không phải là 1 dãy đồ thị (why?)

Định lý 2 (Euler's, [HT24], Thm. 7.9, p. 250). Cho $G = (V, E)$ là đồ thị tổng quát với $d_1, \dots, d_{|V|} \in \mathbb{N}$ là bậc của các đỉnh. Khi đó $\sum_{i=1}^{|V|} d_i = d_1 + d_2 + \dots + d_{|V|} = 2|E|$. Nói riêng, số đỉnh của G có bậc lẻ là số chẵn.

Briefly:

$$d_1, \dots, d_{|V|} \text{ are degrees of vertices of a graph } G = (V, E) \Rightarrow \sum_{i=1}^{|V|} d_i = 2|E| \Rightarrow |\{i; d_i \not\equiv 2\}| : 2.$$

Chú ý chiều ngược lại chưa chắc đúng:

Ví dụ 2. Dãy số 7, 5, 5, 4, 3, 2, 2, 0 không mâu thuẫn với Định lý 2 nhưng nó không phải là đồ thị (why?).

Question 2. Có thể suy ra được những hệ quả nào từ đẳng thức $\sum_{i=1}^{|V|} d_i = 2|E|$?

Bài toán 22. Cho $G = (V, E)$ là đồ thị tổng quát với $d_1, \dots, d_p \in \mathbb{N}$ là bậc của các đỉnh. Chứng minh: Bậc cao nhất $d_{\max} := \max_{1 \leq i \leq p} d_i$ thỏa $d_{\max} \geq \frac{2|E|}{|V|}$.

3 Algorithms on Graphs – Các Thuật Toán Trên Đồ Thị

3.1 Shortest path problem – Bài toán tìm đường đi ngắn nhất

In graph theory, the *shortest path problem* is the problem of finding a **path** between 2 **vertices** (or nodes) in a graph s.t. the sum of weights of its constituent edges is minimized.

The problem of finding the shortest path between 2 intersections on a road map may be modeled as a special case of the shortest path problem in graphs, where the vertices correspond to intersections & the edges correspond to road segments, each weighted by the length or distance of each segment.

The shortest path problem can be defined for graphs whether undirected, directed, or **mixed**. The definition for undirected graphs states that every edge can be traversed in either direction. Directed graphs require that consecutive vertices be connected by an appropriate directed edge.

Recall that 2 vertices are adjacent when they are both incident to a common edge. A *path* in an undirected graph is a sequence of vertices $P = (v_1, v_2, \dots, v_n) \in V^n = V \times V \times \dots \times V$ s.t. v_i is adjacent to v_{i+1} , $\forall i = 1, \dots, n-1$. Such a path P is called a path of length $n-1$ from v_1 to v_n . (The v_i are variables; their numbering relates to their position in the sequence & need not relate to a canonical labeling.)

Let $E = \{e_{i,j}\}$ where $e_{i,j}$ is the edge incident to both v_i & v_j . Given a real-valued weight function $f : E \rightarrow \mathbb{R}$, & an undirected simple graph G , the shortest path from v to v' is the path $P = (v_1, \dots, v_n)$ (where $v_1 = v$ & $v_n = v'$) that over all possible n minimizes the sum $\sum_{i=1}^{n-1} f(e_{i,i+1})$. When each edge in the graph has unit weight or $f : E \rightarrow \{1\}$, this is equivalent to finding the path with fewest edges.

$$\min_{P=(v_1, \dots, v_n) \in V^n : \text{path}, v_1=v, v_n=v'} \sum_{i=1}^n f(e_{i,i+1}).$$

The problem is also sometimes called the *single-pair shortest path problem*, to distinguish it from the following variations:

- The *single-source shortest path problem*, in which we have to find shortest paths from a source vertex v to all other vertices in the graph.

$$\min_{P=(v_1, \dots, v_n) \in V^n : \text{path}, v_1=v} \sum_{i=1}^n f(e_{i,i+1}).$$

- The *single-destination shortest path problem*, in which we have to find shortest paths from all vertices in the directed graph to a single destination vertex v . This can be reduced to the single-source shortest path problem by reversing the arcs in the directed graph.

$$\min_{P=(v_1, \dots, v_n) \in V^n : \text{path}, v_n=v'} \sum_{i=1}^n f(e_{i,i+1}).$$

- The *all-pairs shortest path problem*, in which we have to find shortest paths between every pair of vertices v, v' in the graph.

$$\min_{P=(v_1, \dots, v_n) \in V^n : \text{path}} \sum_{i=1}^n f(e_{i,i+1}).$$

These generalizations have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest path algorithm on all relevant pairs of vertices.

– Những tổng quát hóa này có thuật toán hiệu quả hơn đáng kể so với cách tiếp cận đơn giản là chạy thuật toán đường dẫn ngắn nhất một cặp trên tất cả các cặp đỉnh có liên quan.

3.1.1 Algorithms

Several well-known algorithms exist for solving the shortest path problem & its variants.

- **Dijkstra's algorithm** solves the single-source shortest path problem with only nonnegative edge weights.
- **Bellman–Ford algorithm** solves the single-source problem if edge weights may be negative.
- **A* search algorithm** solves for single-pair shortest path using heuristics to try to speed up the search.
- **Floyd–Warshall algorithm** solves all pairs shortest paths.
- **Johnson's algorithm** solves all pairs shortest paths, & may be faster than Floyd–Warshall on **sparse graphs**.
- **Viterbi algorithm** solves the shortest stochastic path problem with an additional probabilistic weight on each node.

3.2 Dijkstra's algorithm – Thuật toán Dijkstra

“*Dijkstra's algorithm* is an algorithm for finding the **shortest paths** between **nodes** in a weighted graph, which may represent, e.g., a **road network**. It was conceived by computer scientist **EDSGER W. DIJKSTRA** in 1956 & published 3 years later.

Dijkstra's algorithm finds the shortest path from a given source code to every node. It can be used to find the shortest path to a specific destination node, by terminating the algorithm after determining the shortest path to the destination node. E.g., if nodes

4 CSES Problem Set/Graph Algorithms

Problem 13 (CSES Problem Set/counting rooms, <https://cses.fi/problemset/task/1192>). *You are given a map of a building, & your task is to count the number of its rooms. The size of the map is $n \times m$ squares, & each square is either floor or wall. You can walk left, right, up, & down through the floor squares.*

Input. *The 1st input lines has 2 integers n, m : the height & width of the map. Then there are n lines of m characters describing the map. Each character is either . (floor) or # (wall).*

Output. *Print 1 integer: the number of rooms.*

Constraints. $1 \leq n, m \leq 10^3$.

Sample. Input:

```
5 8
#####
#.#...#
####.#.#
#.#...#
#####
```

Output: 3.

Problem 14 (CSES Problem Set/labyrinth, <https://cses.fi/problemset/task/1193>). *You are given a map of a labyrinth, task: find a path from start to end. You can walk left, right, up, & down.*

Input. *The 1st input line has 2 integers n, m : the height & width of the map. Then there are n lines m characters describing the labyrinth. Each character is . (floor), # (wall), A (start), or B (end). There is exactly 1 A & 1 B in the input.*

Output. *1st print YES, if there is a path, & No otherwise. If there is a path, print the length of the shortest such path & its description as a string consisting of characters L (left), R (right), U (up), & D (down). You can print any valid solution.*

Constraints. $1 \leq n, m \leq 1000$.

Sample. Input:

```
5 8
#####
#.A#...#
#...#B#
#.....#
#####
```

Output:

```
YES
9
LDDRRRRRU
```

Problem 15 (CSES Problem Set/building roads, <https://cses.fi/problemset/task/1666>). Byteland has n cities, & m roads between them. Goal: construct new roads so that there is a route between any 2 cities. Task: find out the minimum number of roads required, & also determine which roads should be built.

Input. The 1st input line has 2 integers n, m : the number of cities & roads. The cities are numbered $1, 2, \dots, n$. After that, there are m lines describing the roads. Each line has 2 integers a, b : there is a road between those cities. A road always connects 2 different cities, & there is at most 1 road between any 2 cities.

Output. 1st print an integer k : the number of required roads. Then, print k lines that describe the new roads. You can print any valid solution.

Constraints. $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq a, b \leq n$.

Sample.

build_road.inp	build_road.out
4 2	1
1 2	2 3
3 4	

Problem 16 (CSES Problem Set/message route, <https://cses.fi/problemset/task/1667>). Syrjälä's network has n computers & m connections. Task: find out if Uolevi can send a message to Maija, & if it is possible, what is the minimum number of computers on such a route.

Input. The 1st input line has 2 integers n, m : the number of computers & connections. The computers are numbered $1, 2, \dots, n$. Uolevi's computer is 1 & Maija's computer is n . Then, there are m lines describing the connections. Each line has 2 integers a, b : there is a connection between those computers. Every connection is between 2 different computers, & there is at most 1 connection between any 2 computers.

Output. If it is possible to send a message, 1st print k : the minimum number of computers on a valid route. After this, print an example of such a route. You can print any valid solution. If there are no routes, print IMPOSSIBLE.

Constraints. $2 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq a, b \leq n$.

Sample.

message_route.inp	message_route.out
5 5	3
1 2	1 4 5
1 3	
1 4	
2 3	
5 4	

Problem 17 (CSES Problem Set/building team, <https://cses.fi/problemset/task/1668>). There are n pupils in Uolevi's class, & m friendships between them. Task: divide pupils into 2 teams in such a way that no 2 pupils in a team are friends. You can freely choose the sizes of the teams.

Input. The 1st input line has 2 integers n, m : the number of pupils & friendships. The pupils are numbered $1, 2, \dots, n$. Then, there are m lines describing the friendships. Each line has 2 integers a, b : pupils a, b are friends. Every friendship is between 2 different pupils. You can assume that there is at most 1 friendship between any 2 pupils.

Output. Print an example of how to build the teams. For each pupil, print 1 or 2 depending on to which team the pupil will be assigned. You can print any valid team. If there are no solutions, print IMPOSSIBLE.

Constraints. $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq a, b \leq n$.

Sample.

build_team.inp	build_team.out
5 3	1 2 2 1 2
1 2	
1 3	
4 5	

Problem 18 (CSES Problem Set/round trip, <https://cses.fi/problemset/task/1669>). Byteland has n cities & m roads between them. Task: design a round trip that begins in a city, goes through 2 or more other cities, & finally returns to starting city. Every intermediate city on the route has to be distinct.

Input. The 1st input line has 2 integers n, m : the number of cities & roads. The cities are numbered $1, 2, \dots, n$. Then, there are m lines describing the roads. Each line has 2 integers a, b : there is a road between those cities. Every road is between 2 different cities, & there is at most 1 road between any 2 cities.

Output. 1st print an integer k : the number of cities on the route. Then print k cities in order they will be visited. You can print any valid solution. If there are no solutions, print IMPOSSIBLE.

Constraints. $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq a, b \leq n$.

Sample.

build_team.inp	build_team.out
5 6	4
1 3	3 5 1 3
1 2	
5 3	
1 5	
2 4	
4 5	

Problem 19 (CSES Problem Set/monsters, <https://cses.fi/problemset/task/1194>). You & some monsters are in a labyrinth. When taking a step to some direction in the labyrinth, each monster may simultaneously take 1 as well. Goal: reach 1 of the boundary squares without ever sharing a square with a monster. Task: find out if your goal is possible, & if it is, print a path that you can follow. Your plan has to work in any situation; even if the monsters know your path beforehand.

Input. The 1st input line has 2 integers n, m : the height & width of the map. After this there are n lines of m characters describing the map. Each character is . (floor), # (wall), A (start), or M (monster). There is exactly 1 A in the input.

Output. 1st print YES if your goal is possible, & NO otherwise. If your goal is possible, also print an example of a valid path (the length of the path & its description using characters D, U, L, R). You can print any path, as long as its length is at most mn steps.

Constraints. $1 \leq m, n \leq 10^3$.

Sample. Input:

```
5 8
#####
#M..A..#
#.#M#.#
#M#...#
#.#####
```

Output:

```
YES
5
RRDDR
```

Problem 20 (CSES Problem Set/shortest routes I, <https://cses.fi/problemset/task/1671>). There are n cities & m flight connections between them. Task: determine the length of the shortest route from Syrjälä to every city.

Input. The 1st input line has 2 integers n, m : the number of cities & flight connections. The cities are numbered $1, 2, \dots, n$, & city 1 is Syrjälä. After that, there are m lines describing the flight connections. Each line has 3 integers a, b, c : a flight begins at city a , ends at city b , & its length is c . Each flight is a 1-way flight. You can assume that it is possible to travel from Syrjälä to all other cities.

Output. Print n integers: the shortest route lengths from Syrjälä to cities $1, 2, \dots, n$.

Constraints. $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq a, b \leq n, 1 \leq c \leq 10^9$.

Sample.

shortest_route_I.inp	shortest_route_I.out
3 4	0 5 2
1 2 6	
1 3 2	
3 2 3	
1 3 4	

Problem 21 (CSES Problem Set/shortest routes II, <https://cses.fi/problemset/task/1672>). There are n cities & m roads between them. Task: process q queries where you have to determine the length of the shortest route between 2 given cities.

Input. The 1st input line has 3 integers n, m, q : the number of cities, roads, & queries. Then, there are m lines describing the roads. Each line has 3 integers a, b, c : there is a road between cities a & b whose length is c . All roads are 2-way roads. Finally, there are q lines describing the queries. Each line has 2 integers a, b : determine the length of the shortest route between cities a & b .

Output. Print the length of the shortest route for each query. If there is no route, print -1 instead.

Constraints. $1 \leq n \leq 500, 1 \leq m \leq n^2, 1 \leq q \leq 10^5, 1 \leq a, b \leq n, 1 \leq c \leq 10^9$.

Sample.

shortest_route_II.inp	shortest_route_II.out
4 3 5	5
1 2 5	5
1 3 9	8
2 3 3	-1
1 2	3
2 1	
1 3	
1 4	
3 2	

Problem 22 (CSES Problem Set/high score, <https://cses.fi/problemset/task/1673>). You play a game consisting of n rooms & m tunnels. Your initial score is 0, & each tunnel increases your score by x where x may be both positive or negative. You may go through a tunnel several times. Task: walk from room 1 to room n . What is the maximum score you can get?

Input. The 1st input line has 2 integers n, m : the number of rooms & tunnels. The rooms are numbered $1, 2, \dots, n$. Then, there are m lines describing the tunnels. Each line has 3 integers a, b, x : the tunnel starts at room a , ends at room b , & it increases your score by x . All tunnels are 1-way tunnels. You can assume that it is possible to get from room 1 to room n .

Output. Print 1 integer: the maximum score you can get. However, if you can get an arbitrarily large score, print -1.

Constraints. $1 \leq n \leq 2500, 1 \leq m \leq 5000, 1 \leq a, b \leq n, -10^9 \leq x \leq 10^9$.

Sample.

high_score.inp	high_score.out
4 5	5
1 2 3	
2 4 -1	
1 3 -2	
3 4 7	
1 4 4	

Problem 23 (CSES Problem Set/flight discount, <https://cses.fi/problemset/task/1195>). Task: find a minimum-price flight route from Syrjälä to Metsälä. You have 1 discount coupon, using which you can halve the price of any single flight during the route. However, you can only use the coupon once. When you use the discount coupon for a flight whose price is x , its price becomes $\lfloor \frac{x}{2} \rfloor$.

Input. The 1st input line has 2 integers n, m : the number of cities & flight connections. The cities are numbered $1, 2, \dots, n$. City 1 is Syrjälä, & city n is Metsälä. After this there are m lines describing the flights. Each line has 3 integers a, b, c : a flight begins at city a , ends at city b , & its price is c . Each flight is unidirectional. You can assume that it is always possible to get from Syrjälä to Metsälä.

Output. Print 1 integer: the price of the cheapest route from Syrjälä to Metsälä.

Constraints. $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq a, b \leq n, 1 \leq c \leq 10^9$.

Sample.

flight_discount.inp	flight_discount.out
3 4	2
1 2 3	
2 3 1	
1 3 7	
2 1 5	

Problem 24 (CSES Problem Set/cycle finding, <https://cses.fi/problemset/task/1197>). You are given a directed graph, & task: find out if it contains a negative cycle, & also give an example of such a cycle.

Input. The 1st input line has 2 integers n, m : the number of nodes & edges. The nodes are numbered $1, 2, \dots, n$. After this, the input has m lines describing the edges. Each line has 3 integers a, b, c : there is an edge from node a to node b whose length is c .

Output. If the graph contains a negative cycle, print 1st YES, & then the nodes in the cycle in their correct order. If there are several negative cycles, you can print any of them. If there are no negative cycles, print NO.

Constraints. $1 \leq n \leq 2500, 1 \leq m \leq 5000, 1 \leq a, b \leq n, -10^9 \leq c \leq 10^9$.

Sample.

cycle_finding.inp	cycle_finding.out
4 5	YES
1 2 1	1 2 4 1
2 4 1	
3 1 1	
4 1 -3	
4 3 -2	

5 CSES Problem Set/Tree Algorithms

6 Posets, Kết Nối, Lưới Boolean

7 Miscellaneous

Tài liệu

- [AD10] Titu Andreescu and Gabriel Dospinescu. *Problems from the Book*. 2nd. XYZ Press, 2010, p. 571. ISBN: 978-0979926907.
- [Hal60] Paul Richard Halmos. *Naive set theory*. The University Series in Undergraduate Mathematics. D. Van Nostrand Co., Princeton, N.J.-Toronto-London-New York, 1960, pp. vii+104.
- [Hal74] Paul Richard Halmos. *Naive set theory*. Undergraduate Texts in Mathematics. Reprint of the 1960 edition. Springer-Verlag, New York-Heidelberg, 1974, pp. vii+104.
- [HT24] Bùi Việt Hà and Vương Trọng Thanh. *Các Vấn Đề Trong Tổ Hợp*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2024, p. 429.
- [Kap72] Irving Kaplansky. *Set theory and metric spaces*. Allyn and Bacon Series in Advanced Mathematics. Allyn and Bacon, Inc., Boston, Mass., 1972, pp. xii+140.
- [Kap77] Irving Kaplansky. *Set theory and metric spaces*. Second. Chelsea Publishing Co., New York, 1977, xii+140 pp. ISBN 0-8284-0298-1.
- [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: [10.1007/978-3-030-81885-2](https://doi.org/10.1007/978-3-030-81885-2). URL: <https://doi.org/10.1007/978-3-030-81885-2>.