

[SOT_SUM25] Combinatorics & Graph Theory

Vo Ngoc Tram Anh

May 12, 2025

https://github.com/vntanh1406/Graph_SUM2025/blob/main/VNTA_GraphSUM25.pdf

— Update **12/05/2025** —

Bài tập làm thêm ở nhà: subsection 2.1, subsection 2.2, 4.1.8

— Update **07/05/2025** —

Bài tập đã làm trên lớp: 2.1, 2.2

Bài tập làm thêm ở nhà: 1.3.1, 1.4.1, 1.8.1, 1.8.2, 1.8.3

Contents

Contents	1
1 Basic Combinatorics	3
1.1 Inclusion-Exclusion Principle	3
1.2 Problems on inclusion-exclusion principle	3
1.3 Problems on counting	3
1.3.1 Bài toán 2. Số cách đặt dấu ngoặc đúng	3
1.3.2 Bài toán 3: Code tính P_n, A_n^k, C_n^k , số catalan thứ n	3
1.3.3 Code in $n + 1$ dòng đầu tiên của tam giác Pascal và khai triển nhị thức Newton của $(a + b)^n, (a + b + c)^n, (\sum_{i=1}^m a_i)^n$	3
1.4 Method of mathematical induction & recurrence	4
1.4.1 Problem 3	4
1.5 Principle of strong induction	4
1.6 Fibonacci & Lucas numbers	4
1.7 Pigeonhole principle & Ramsey theory	4
1.8 Counting rules & Stirling number of type 1 & type 2	4
1.8.1 Problem 6	4
1.8.2 Problem 7: Prove that the number of subsets of $[n] = 2^n, \forall n \in N^*$	5
1.8.3 Problem 8	5
1.8.4 Problem 9	7
1.9 Permutation & Combination	7
1.9.1 Consecutive 2 Dice Rolls	7
1.9.2 Simultaneous 2 Dice Rolls	9
1.9.3 Consecutive n Dice Rolls	11
1.9.4 Simultaneous n Dice Rolls	11
1.9.5 Prime and Composite	11
1.9.6 Even and Odd	13
2 Nhị thức Newton & đa thức	14
2.1 Bài toán 27	14

2.1.1	Tính $(a + b + c)^n$	14
2.1.2	Tính $(a + b + c + d)^n$	14
2.2	Đẳng thức tổ hợp	15
2.2.1	Bài toán 28	15
2.2.2	Bài toán 29	17
2.2.3	Bài toán 30	18
2.2.4	Bài toán 31	19
3	Basic Graph Theory	20
3.1	Graph representation	20
3.2	Search Algorithm	21
3.2.1	Basic DFS & BFS	21
3.2.2	DFS & BFS for grid	21
3.2.3	Important algorithms	21
4	CSES Problem List	21
4.1	Graph Algorithms	21
4.1.1	Counting Rooms	21
4.1.2	Labyrinth	23
4.1.3	Building Roads	24
4.1.4	Message Routes	25
4.1.5	Building Teams	26
4.1.6	Round Trip	26
4.1.7	Monsters	27
4.1.8	Flight Routes	27

1 Basic Combinatorics

1.1 Inclusion-Exclusion Principle

1.2 Problems on inclusion-exclusion principle

1.3 Problems on counting

1.3.1 Bài toán 2. Số cách đặt dấu ngoặc đúng

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/ValidParentheses.cpp

- Số Catalan thỏa mãn hệ thức truy hồi sau:
 - $C_0 = 1$
 - $C_{n+1} = \sum_{i=0}^n C_i \cdot C_{n-i}$ (Ref: *Wikipedia: Catalan number / Properties*)
- Gọi D_n là số chuỗi đúng đắn gồm n dấu ngoặc mở và n dấu ngoặc đóng. Ta cần chứng minh $D_n = C_n = \frac{1}{n+1} \cdot \binom{2n}{n}$.
- Với $n = 0$, chỉ có 1 cách (chuỗi rỗng): $D_0 = C_0 = \frac{1}{0+1} \cdot \binom{0}{0} = 1$
- Giả sử $D_k = C_k = \frac{1}{k+1} \cdot \binom{2k}{k}$ đúng $\forall k \leq n, k \in \mathbb{N}$. Ta cần chứng minh rằng điều này cũng đúng với D_{n+1} .
- Thật vậy, mỗi chuỗi ngoặc đúng có thể viết thành dạng $(S_1)S_2$, trong đó:
 - S_1 là chuỗi ngoặc đúng với i cặp ngoặc.
 - S_2 là chuỗi ngoặc đúng với $n - i$ cặp ngoặc.

Khi đó, tổng số cặp ngoặc trong chuỗi $(S_1)S_2$ là: $i + (n - i) + 1 = n + 1$ cặp ngoặc.

Vậy số chuỗi ngoặc đúng có thể được biểu diễn thành: $D_{n+1} = \sum_{i=0}^n D_i \cdot D_{n-i}$

Áp dụng giả thiết quy nạp, ta có: $D_{n+1} = \sum_{i=0}^n C_i \cdot C_{n-i} = C_{n+1}$

- Do đó, ta đã chứng minh được rằng $D_{n+1} = C_{n+1}$, hoàn thành chứng minh quy nạp.

1.3.2 Bài toán 3: Code tính P_n, A_n^k, C_n^k , số catalan thứ n

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/CalculateP_A_C_Catalan.cpp

1.3.3 Code in $n + 1$ dòng đầu tiên của tam giác Pascal và khai triển nhị thức Newton của $(a + b)^n, (a + b + c)^n, (\sum_{i=1}^m a_i)^n$

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/PascalTriaAndMultinomial.cpp

1.4 Method of mathematical induction & recurrence

1.4.1 Problem 3

- Gọi $f(n)$ là số vùng mà các đường thẳng tạo ra.
- Với $n = 0$, không có đường thẳng nào, hình vuông là một vùng duy nhất.
$$f(0) = 1 + \frac{0 \cdot 1}{2} = 1$$
- Với $n = 1$, 1 đường thẳng chia hình vuông thành 2 vùng.
$$f(1) = 1 + \frac{1 \cdot 2}{2} = 2$$
- Với $n = 2$, 2 đường thẳng cắt nhau chia hình vuông thành 4 vùng.
$$f(2) = 1 + \frac{2 \cdot 3}{2} = 4$$
- Giả sử $f(k) = 1 + \frac{k \cdot (k+1)}{2}$ đúng, ta cần chứng minh điều này đúng với $k + 1$.
- Thật vậy, khi thêm đường thẳng thứ $(k + 1)$ vào hình vuông đã có k đường thẳng, thì đường thẳng thứ $(k + 1)$ này:
 - Cắt tất cả các đường thẳng trước đó (vì mọi cặp đường thẳng đều giao nhau)
 - Không có 3 đường thẳng nào đồng quy
 - Tạo ra k giao điểm mới, các giao điểm này chia đường thẳng thứ $(k + 1)$ thành $k + 1$ đoạn
 - * 1 đoạn từ điểm bắt đầu trên cạnh đến giao điểm đầu tiên
 - * $k - 1$ đoạn giữa các giao điểm
 - * 1 đoạn từ giao điểm cuối đến điểm kết thúc trên cạnh
 - Số vùng mới tạo ra là:
 - * Mỗi đoạn của đường thẳng thứ $k + 1$ nằm trong một vùng hiện có (do đường thẳng đi qua các vùng được tạo từ k đường thẳng trước đó)
 - * Khi đường thẳng $(k + 1)$ đi qua một vùng, nó chia vùng đó thành hai vùng mới.
 - * Vì đường thẳng $(k + 1)$ có $k + 1$ đoạn, mỗi đoạn chia một vùng thành hai (tức là thêm một vùng mới), nên đường thẳng này tạo ra $k + 1$ vùng mới so với $f(k)$.
 - Do đó: $f(k + 1) = f(k) + (k + 1)$
 - Áp dụng giả thiết quy nạp, ta có:
$$f(k + 1) = 1 + \frac{k \cdot (k+1)}{2} + (k + 1) = 1 + \frac{k(k+1) + 2(k+1)}{2} = 1 + \frac{(k+1)(k+2)}{2} \text{ (đpcm)}$$

1.5 Principle of strong induction

1.6 Fibonacci & Lucas numbers

1.7 Pigeonhole principle & Ramsey theory

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

1.8.1 Problem 6

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/ValidSequences.cpp

- Vì không có hai số 0 đứng cạnh nhau, nên:

- Mỗi cặp số 0 phải được ngăn cách bởi ít nhất một số 1.
- Có m số 0, vậy có ít nhất $m - 1$ số 1 để ngăn cách chúng.
- Có $n - m$ số 1, nên để tồn tại dãy hợp lệ thì $n - m \geq m - 1 \Leftrightarrow n \geq 2m - 1$
- Đặt $n - m$ số 1 vào dãy, chiếm $n - m$ vị trí. Còn lại m vị trí cần điền bằng số 0.
- $n - m$ số 1 tạo ra $n - m + 1$ khoảng trống để có thể đặt số 0 vào:
 - 1 khoảng trống trước số 1 đầu tiên
 - 1 khoảng trống sau số 1 cuối cùng
 - $n - m - 1$ khoảng trống giữa các số 1 (nếu có ít nhất hai số 1)
- Để đặt m số 0 sao cho không có hai số 0 nào liên tiếp, ta đặt tối đa một số 0 vào mỗi khoảng trống. Vậy ta cần chọn ra m trong số $n - m + 1$ khoảng trống để đặt số 0.
- Vậy tổng số cách chọn là: $\binom{n-m+1}{m}$

1.8.2 Problem 7: Prove that the number of subsets of $[n] = 2^n, \forall n \in N^*$

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/SubsetsOf%5Bn%5D.cpp

==Cách 1: Quy nạp==

- Với $n=1$, khi đó $[n] = \{1\}$. Các tập con của $[1]$ là: $\emptyset, \{1\} \Rightarrow$ Có $2 = 2^1$ tập con.
- Giả sử $[k]$ có 2^k tập con đúng $\forall k \leq n, k \in N^*$. Ta cần chứng minh điều này đúng với $n + 1$.
- Xét $[n + 1] = \{1, 2, \dots, n, n + 1\}$
Mỗi tập con của $[n + 1]$ có 2 khả năng với phần tử $n + 1$:
 - Không chứa $n + 1$: Chính là các tập con của $[n]$, có tổng cộng 2^n tập.
 - Có chứa $n + 1$: Chính là các tập con của $[n]$ có thêm phần tử $n + 1$ vào, có tổng cộng 2^n tập.
- Vậy tổng số tập con của $[n + 1]$ là: $2 \cdot 2^n = 2^{n+1}$, hoàn thành chứng minh quy nạp.

==Cách 2: Nguyên lý đếm==

- Xét $[n] = \{1, 2, \dots, n\}$ có tổng cộng n phần tử.
- Mỗi phần tử có 2 lựa chọn khi tạo tập con: Chọn hoặc Không chọn.
- Vậy theo quy tắc nhân, tổng số cách chọn các tập con là: 2^n

1.8.3 Problem 8

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/Derangement.cpp

a)

- Bài toán có thể phát biểu thành:

- Xét $M = \{1, 2, \dots, n\}$ gồm n phần tử. Một hoán vị $M' = \{x_1, x_2, \dots, x_n\}$, $x_i \in M, i = \overline{1, n}$ được gọi là có k điểm bất động nếu có đúng k phần tử $x_i \in M'$ sao cho $x_i = i$.
- Gọi $f(n)$ là số hoán vị không có phần tử nào bất động.

Chứng minh $f(n) = n! \cdot \sum_{i=0}^n \frac{(-1)^i}{i!}$

- Tổng số cách hoán vị n phần tử: $n!$
- Gọi A_i là tập các hoán vị mà phần tử thứ i nằm đúng vị trí i . Khi đó, tập các hoán vị có ít nhất một điểm cố định là $\bigcup_{i=1}^n A_i$
- Áp dụng nguyên lý bù trừ, số hoán vị có ít nhất một điểm bất động là:

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

Nhận thấy, $A_{i_1} \cap \dots \cap A_{i_k}$ hay $\bigcap_{i=1}^k A_{i_k}$ là tập các hoán vị mà tất cả những phần tử i_1, i_2, \dots, i_k đều bất động.

Với mỗi tập hợp có k điểm bất động, $n-k$ phần tử còn lại hoán vị tự do: $\left| \bigcap_{i=1}^k A_{i_k} \right| = (n-k)!$

Số cách chọn k phần tử bất động: $\binom{n}{k}$

Do đó tổng số tập hợp là:

$$|\bigcup_{i=1}^n A_i| = \sum_{k=1}^n (-1)^{k+1} \binom{n}{k} (n-k)! = \sum_{k=1}^n (-1)^n \cdot \frac{n!}{k!}$$

- Vậy số các hoán vị không có điểm bất động nào là:

$$f(n) = n! - \frac{n!}{1!} + \frac{n!}{2!} - \dots + \frac{(-1)^n \cdot n!}{n!} = \sum_{i=0}^n \frac{(-1)^i}{i!} \quad (\text{đpcm})$$

b)

- Khai triển Taylor: $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + o(x^n)$
- Tại $x = -1$: $e^{-1} = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!}$
- Vậy:

- $n! \left(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \dots + \frac{(-1)^n}{n!} \right) > n! \cdot e^{-1}$ nếu n chẵn
- $n! \left(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \dots + \frac{(-1)^n}{n!} \right) < n! \cdot e^{-1}$ nếu n lẻ

Hay:

- $n! \left(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \dots + \frac{(-1)^n}{n!} \right) + e^{-1} > (n! + 1) \cdot e^{-1}$ nếu n chẵn
- $n! \left(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \dots + \frac{(-1)^n}{n!} \right) + e^{-1} < (n! + 1) \cdot e^{-1}$ nếu n lẻ

Suy ra: $f(n) = n! \left(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \dots + \frac{(-1)^n}{n!} \right) = \left\lfloor \frac{n!+1}{e} \right\rfloor \quad (\text{đpcm})$

1.8.4 Problem 9

1.9 Permutation & Combination

1.9.1 Consecutive 2 Dice Rolls

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/Consecutive2DiceRolls.cpp

Let the sample space be denoted by Ω .

We consider the experiment of rolling two distinguishable six-sided dice in sequence. Each die has 6 possible outcomes, and since the rolls are independent and ordered, the sample space consists of all ordered pairs (i, j) where $i, j \in \{1, 2, 3, 4, 5, 6\}$.

Therefore, the total number of outcomes is: $|\Omega| = 6 \cdot 6 = 36$

a)

- Let A_1 be the event that both dice show the same number of dots. This corresponds to the set of outcomes: $A_1 = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)\}$

So, $P(A_1) = \frac{6}{36} = \frac{1}{6}$

- Let A_2 be the event that both dice show different numbers of dots. Since there are 6 outcomes with the same number, there are: $36 - 6 = 30$ outcomes with different numbers

So: $P(A_2) = \frac{30}{36} = \frac{5}{6}$

b)

Even numbers on a die: $\{2, 4, 6\}$

Odd numbers on a die: $\{1, 3, 5\}$

- Let B_1 be the event that both dice have the same parity:

– Both even: $3 \cdot 3 = 9$ outcomes

– Both odd: $3 \cdot 3 = 9$ outcomes

Total favorable outcomes: $9 + 9 = 18$. So: $P(B_1) = \frac{18}{36} = \frac{1}{2}$

- Let B_2 be the event that the two numbers have different parity:

– First even, second odd: $3 \cdot 3 = 9$ outcomes

– First odd, second even: $3 \cdot 3 = 9$ outcomes

Total favorable outcomes: $9 + 9 = 18$. So: $P(B_2) = \frac{18}{36} = \frac{1}{2}$

c)

Prime numbers on a die: $\{2, 3, 5\}$

Composite numbers on a die: $\{4, 6\}$

- Let C_1 be the event that the numbers on both dice are prime numbers:

Both prime: $3 \cdot 3 = 9$ outcomes. So: $P(C_1) = \frac{9}{36} = \frac{1}{4}$

- Let C_2 be the event that the numbers on both dice are composite numbers:

Both composite: $2 \cdot 2 = 4$ outcomes. So: $P(C_2) = \frac{4}{36} = \frac{1}{9}$

- Let C_3 be the event that at least one of the two dice shows a prime number.

The number of outcomes where neither die shows a prime number, or both dice show non-prime numbers: $3 \cdot 3 = 9$ (*non-prime*: $\{1, 4, 6\}$)

Therefore, the number of favorable outcomes: $36 - 9 = 27$. So: $P(C_3) = \frac{27}{36} = \frac{3}{4}$

- Let C_4 be the event that at least one of the two dice shows a composite number.

The number of outcomes where neither die shows a composite number, or both dice show non-composite numbers: $4 \cdot 4 = 16$ (*non-composite*: $\{1, 2, 3, 5\}$)

Therefore, the number of favorable outcomes: $36 - 16 = 20$. So: $P(C_4) = \frac{20}{36} = \frac{5}{9}$

d)

Let D be the event that one of the two numbers is a divisor or a multiple of the other.

Let each outcome be represented as an ordered pair (a, b) , where $a, b \in \{1, 2, 3, 4, 5, 6\}$.

We are interested in counting the number of outcomes where: $a \mid b$ or $b \mid a$

The valid pairs:

- When $a = 1$: $b = 1, 2, 3, 4, 5, 6$ (6 outcomes)
- When $a = 2$: $b = 1, 2, 4, 6$ (4 outcomes)
- When $a = 3$: $b = 1, 3, 6$ (3 outcomes)
- When $a = 4$: $b = 1, 2, 4$ (3 outcomes)
- When $a = 5$: $b = 1, 5$ (2 outcomes)
- When $a = 6$: $b = 1, 2, 3, 6$ (4 outcomes)

Total favorable outcomes: $6 + 4 + 3 + 3 + 2 + 4 = 22$. So: $P(D) = \frac{22}{36} = \frac{11}{18}$

e)

Let E_n be the event that the sum of the two dice is equal to n .

For $n \in \{2, 3, \dots, 12\}$, we define the function: $f(n) = \min\{n - 1, 6\} - \max\{n - 6, 1\} + 1$

This function counts the number of integer pairs $(a, b) \in \{1, 2, 3, 4, 5, 6\}^2$ such that $a + b = n$.

- The smallest possible sum is $1 + 1 = 2$, and the largest is $6 + 6 = 12$, so $n \in \{2, 3, \dots, 12\}$.
- For a fixed n , valid pairs (a, b) must satisfy: $a \in [\max(1, n - 6), \min(6, n - 1)]$, and then $b = n - a$.
- Therefore, the number of such values of a is: $f(n) = \min(n - 1, 6) - \max(n - 6, 1) + 1$
- This formula works because:
 - $\min(n - 1, 6)$ gives the largest possible value of a such that $b = n - a \geq 1$
 - $\max(n - 6, 1)$ gives the smallest possible value of a such that $b = n - a \leq 6$
 - The total number of integers a in that interval is: upper bound – lower bound + 1

So: $P(E_n) = \frac{f(n)}{36} \cdot \mathbf{1}_{\{2 \leq n \leq 12\}}$

1.9.2 Simultaneous 2 Dice Rolls

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/Consecutive2DiceRolls.cpp

Let the sample space be denoted by Ω .

We consider the experiment of rolling two indistinguishable six-sided dice simultaneously. Each die has 6 possible outcomes, and since the dice are indistinguishable and rolled at the same time, the sample space consists of all unordered pairs (i, j) where $i, j \in \{1, 2, 3, 4, 5, 6\}$ and $i \leq j$.

Therefore, the total number of outcomes is: $|\Omega| = 6 + \binom{6}{2} = 21$

a)

- Let A_1 be the event that both dice show the same number of dots. This corresponds to the set of outcomes: $A_1 = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)\}$

So, $P(A_1) = \frac{6}{21} = \frac{2}{7}$

- Let A_2 be the event that both dice show different numbers of dots. Since there are 6 outcomes with the same number, there are: $21 - 6 = 15$ outcomes with different numbers

So: $P(A_2) = \frac{15}{21} = \frac{5}{7}$

b)

Even numbers on a die: $\{2, 4, 6\}$

Odd numbers on a die: $\{1, 3, 5\}$

- Let B_1 be the event that both dice have the same parity:

– Both even: $3 + \binom{3}{2} = 6$ outcomes

– Both odd: $3 + \binom{3}{2} = 6$ outcomes

Total favorable outcomes: $6 + 6 = 12$. So: $P(B_1) = \frac{12}{21} = \frac{4}{7}$

- Let B_2 be the event that the two numbers have different parity:

Total favorable outcomes: $3 \cdot 3 = 9$. So: $P(B_2) = \frac{9}{21} = \frac{3}{7}$

c)

Prime numbers on a die: $\{2, 3, 5\}$

Composite numbers on a die: $\{4, 6\}$

- Let C_1 be the event that the numbers on both dice are prime numbers:

Both prime: $3 + \binom{3}{2} = 6$ outcomes. So: $P(C_1) = \frac{6}{21} = \frac{2}{7}$

- Let C_2 be the event that the numbers on both dice are composite numbers:

Both composite: $2 + \binom{2}{2} = 3$ outcomes. So: $P(C_2) = \frac{3}{21} = \frac{1}{7}$

- Let C_3 be the event that at least one of the two dice shows a prime number.

The number of outcomes where neither die shows a prime number, or both dice show non-prime numbers: $3 + \binom{3}{2} = 6$ (*non-prime*: $\{1, 4, 6\}$)

Therefore, the number of favorable outcomes: $21 - 6 = 15$. So: $P(C_3) = \frac{15}{21} = \frac{5}{7}$

- Let C_4 be the event that at least one of the two dice shows a composite number.

The number of outcomes where neither die shows a composite number, or both dice show non-composite numbers: $4 + \binom{4}{2} = 10$ (*non-composite*: $\{1, 2, 3, 5\}$)

Therefore, the number of favorable outcomes: $21 - 10 = 11$. So: $P(C_4) = \frac{11}{21}$

d)

Let D be the event that one of the two numbers is a divisor or a multiple of the other.

Let each outcome be represented as an unordered pair (a, b) , where $a, b \in \{1, 2, 3, 4, 5, 6\}$.

We are interested in counting the number of outcomes where: $a \mid b$ or $b \mid a$

The valid pairs:

- When $a = 1$: $b = 1, 2, 3, 4, 5, 6$ (6 outcomes)
- When $a = 2$: $b = 2, 4, 6$ (3 outcomes)
- When $a = 3$: $b = 3, 6$ (2 outcomes)
- When $a = 4$: $b = 4$ (1 outcome)
- When $a = 5$: $b = 5$ (1 outcome)
- When $a = 6$: $b = 6$ (1 outcome)

Total favorable outcomes: $6 + 3 + 2 + 1 + 1 + 1 = 14$. So: $P(D) = \frac{14}{21} = \frac{2}{3}$

e)

Let E_n be the event that the sum of the two dice is equal to n .

For $n \in \{2, 3, \dots, 12\}$, we define the function: $f(n) = \left\lfloor \frac{\min\{n-1, 6\} - \max\{n-6, 1\}}{2} \right\rfloor + 1$

This function counts the number of unordered integer pairs $(a, b) \in \{1, 2, 3, 4, 5, 6\}^2$ such that $a + b = n$ and $a \leq b$.

- The smallest possible sum is $1 + 1 = 2$, and the largest is $6 + 6 = 12$, so $n \in \{2, 3, \dots, 12\}$.
- For a fixed n , valid unordered pairs (a, b) must satisfy:

$$a \in \left[\max(1, n-6), \left\lfloor \frac{n}{2} \right\rfloor \right], \quad b = n - a, \quad \text{and } a \leq b$$

- Therefore, the number of such values of a is given by:

$$f(n) = \left\lfloor \frac{\min(n-1, 6) - \max(n-6, 1)}{2} \right\rfloor + 1$$

- This formula works because:

- $\min(n-1, 6)$ gives the largest possible value of a such that $b = n - a \in [1, 6]$
- $\max(n-6, 1)$ gives the smallest possible value of a such that $b = n - a \in [1, 6]$
- We divide the range by 2 and take floor to count only unordered pairs (i.e., $a \leq b$)
- Adding 1 accounts for inclusive bounds

So: $P(E_n) = \frac{f(n)}{21} \cdot \mathbf{1}_{\{2 \leq n \leq 12\}}$.

1.9.3 Consecutive n Dice Rolls

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/ConsecutiveNDiceRolls.ipynb

Let the sample space be denoted by Ω .

We consider the experiment of rolling n distinguishable six-sided dice in sequence. Each die has 6 possible outcomes, and since the rolls are independent and ordered, the sample space consists of all ordered pairs (i, j) where $i, j \in \{1, 2, 3, 4, 5, 6\}$.

Therefore, the total number of outcomes is: $|\Omega| = 6^n$

a) Let A be the event that all dice show the same number of dots.

This corresponds to the set of outcomes where $x_1 = x_2 = \dots = x_n$.

There are exactly 6 such outcomes (all 1's, all 2's, ..., all 6's), so: $P(A) = \frac{6}{6^n}$

b) Let B be the event that all dice show different numbers of dots.

This is only possible when $1 \leq n \leq 6$ (since there are only 6 distinct values from 1 to 6).

- If $n > 6$ or $n < 2$, then clearly: $P(B) = 0$
- For $2 \leq n \leq 6$: The number of favorable outcomes as the number of one-to-one mappings from n dice to 6 values, i.e., number of permutations: $P(6, n) = 6 \cdot 5 \cdot 4 \cdots (6 - n + 1)$.
So: $P(B) = \frac{P(6, n)}{6^n} = \frac{6!}{(6-n)! \cdot 6^n}$

c) Let C be the event that all dice have the same parity.

- All even: 3^n outcomes
- All odd: 3^n outcomes

Total favorable outcomes: $2 \cdot 3^n$. So: $P(C) = \frac{2 \cdot 3^n}{6^n} = \frac{1}{2^{n-1}}$

1.9.4 Simultaneous n Dice Rolls

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/ConsecutiveNDiceRolls.ipynb

Let the sample space be denoted by Ω .

We consider the experiment of rolling n indistinguishable six-sided dice simultaneously.

Each die has 6 possible outcomes, and the order of dice does not matter.

So the sample space consists of all multisets of n values chosen from $\{1, 2, 3, 4, 5, 6\}$.

Therefore, the total number of outcomes is: $|\Omega| = \binom{n+5}{5}$

a) Let A be the event that all dice show the same number of dots.

This corresponds to the set of outcomes where $x_1 = x_2 = \dots = x_n$.

There are exactly 6 such outcomes (all 1's, all 2's, ..., all 6's), so: $P(A) = \frac{6}{6^n}$

1.9.5 Prime and Composite

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/PrimeAndComposite.ipynb

Let $A_n = \{1, 2, \dots, n\} \subset \mathbb{N}^*$ be the set of the first n positive integers.

Let $m \in \mathbb{N}^*$, and suppose we randomly select m distinct elements from A_n .

Let $T = \binom{n}{m}$ be the total number of ways to choose m distinct elements from A_n .

Suppose $k \in \{0, 1, \dots, m\}$

a)

Let:

- $d_e = \left\lfloor \frac{n}{2} \right\rfloor$ be the number of even numbers in A_n ,
- $d_o = n - d_e$ be the number of odd numbers in A_n ,

We have the following probabilities:

- $P_{\text{All even}} = \frac{\binom{d_e}{m}}{T}$
- $P_{\text{All odd}} = \frac{\binom{d_o}{m}}{T}$
- $P_{\text{At least one even}} = 1 - P_{\text{All odd}} = 1 - \frac{\binom{d_o}{m}}{T}$
- $P_{\text{At least one odd}} = 1 - P_{\text{All even}} = 1 - \frac{\binom{d_e}{m}}{T}$
- $P_{\text{Exactly k even}} = \frac{\binom{d_e}{k} \cdot \binom{d_o}{m-k}}{T}$
- $P_{\text{Exactly k odd}} = \frac{\binom{d_o}{k} \cdot \binom{d_e}{m-k}}{T}$
- $P_{\text{At least k even}} = \frac{1}{T} \sum_{i=k}^m \binom{d_e}{i} \cdot \binom{d_o}{m-i}$
- $P_{\text{At least k odd}} = \frac{1}{T} \sum_{i=k}^m \binom{d_o}{i} \cdot \binom{d_e}{m-i}$

b)

Let:

- $p = \pi(n)$: the number of prime numbers less than or equal to n
- $c = n - 1 - \pi(n)$: the number of composite numbers in A_n

We have the following probabilities:

- $P_{\text{All prime}} = \frac{\binom{p}{m}}{T}$
- $P_{\text{All composite}} = \frac{\binom{c}{m}}{T}$
- $P_{\text{At least one prime}} = \frac{1}{T} \sum_{i=1}^m \binom{p}{i} \cdot \binom{n-p}{m-i} = 1 - \frac{\binom{c}{m} + \binom{c}{m-1}}{T}$
- $P_{\text{At least one composite}} = \frac{1}{T} \sum_{i=1}^m \binom{c}{i} \cdot \binom{n-c}{m-i} = 1 - \frac{\binom{p}{m} + \binom{p}{m-1}}{T}$
- $P_{\text{Exactly k prime}} = \frac{\binom{p}{k} \cdot \binom{n-p}{m-k}}{T}$
- $P_{\text{Exactly k composite}} = \frac{\binom{c}{k} \cdot \binom{n-c}{m-k}}{T}$
- $P_{\text{At least k prime}} = \frac{1}{T} \sum_{i=k}^m \binom{p}{i} \cdot \binom{n-p}{m-i}$
- $P_{\text{At least k composite}} = \frac{1}{T} \sum_{i=k}^m \binom{c}{i} \cdot \binom{n-c}{m-i}$

1.9.6 Even and Odd

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/EvenAndOdd.ipynb

Let $a, b \in \mathbb{Z}$ with $a < b$, and let $n, k \in \mathbb{N}^*$ such that $n \geq 2$ and $k \leq n$.

Define the set $A = \{a, a+1, a+2, \dots, b\} \subset \mathbb{Z}$ with total size $N = |A| = b - a + 1$.

Let:

- $d_o = \left\lfloor \frac{b-a}{2} \right\rfloor + 1$ be the number of odd numbers in A_n
- $d_e = N - d_o$
- T be the total number of possible selections.

a)

- **Distinct** ($T = \binom{N}{2}$):
 - Probability that both numbers have the same parity: $P_{\text{same parity}} = \frac{\binom{d_e}{2} + \binom{d_o}{2}}{T}$
 - Probability that the two numbers have different parity: $P_{\text{different parity}} = \frac{d_e \cdot d_o}{T}$
- **With replacement** ($T = N^2$):
 - Probability that both numbers have the same parity: $P_{\text{same parity}} = \frac{d_e^2 + d_o^2}{T}$
 - Probability that the two numbers have different parity: $P_{\text{different parity}} = \frac{2 \cdot d_e \cdot d_o}{T}$

b)

Suppose $d_e, d_o \geq n$

- **Distinct** ($T = \binom{N}{n}$):
 - $P_{\text{all even}} = \frac{\binom{d_e}{n}}{T}$
 - $P_{\text{all odd}} = \frac{\binom{d_o}{n}}{T}$
 - $P_{\text{same parity}} = \frac{\binom{d_e}{n} + \binom{d_o}{n}}{T}$
 - $P_{\text{exactly } k \text{ even}} = \frac{\binom{d_e}{k} \cdot \binom{d_o}{n-k}}{T}$
 - $P_{\text{exactly } k \text{ odd}} = \frac{\binom{d_o}{k} \cdot \binom{d_e}{n-k}}{T}$
 - $P_{\text{at least } k \text{ even}} = \frac{1}{T} \sum_{i=k}^n \binom{d_e}{i} \cdot \binom{d_o}{n-i}$
 - $P_{\text{at least } k \text{ odd}} = \frac{1}{T} \sum_{i=k}^n \binom{d_o}{i} \cdot \binom{d_e}{n-i}$
- **With replacement** ($T = N^n$):
 - $P_{\text{all even}} = \frac{d_e^n}{T}$

$$\begin{aligned}
- P_{\text{all odd}} &= \frac{d_o^n}{T} \\
- P_{\text{same parity}} &= \frac{d_e^n + d_o^n}{T} \\
- P_{\text{exactly } k \text{ even}} &= \frac{\binom{n}{k} \cdot d_e^k \cdot d_o^{n-k}}{T} \\
- P_{\text{exactly } k \text{ odd}} &= \frac{\binom{n}{k} \cdot d_o^k \cdot d_e^{n-k}}{T} \\
- P_{\text{at least } k \text{ even}} &= \frac{1}{T} \sum_{i=k}^n \binom{n}{i} \cdot d_e^i \cdot d_o^{n-i} \\
- P_{\text{at least } k \text{ odd}} &= \frac{1}{T} \sum_{i=k}^n \binom{n}{i} \cdot d_o^i \cdot d_e^{n-i}
\end{aligned}$$

2 Nhị thức Newton & đa thức

2.1 Bài toán 27

2.1.1 Tính $(a + b + c)^n$

$$S_n = (a + b + c)^n = [(a + b) + c]^n = (A + c)^n = \sum_{k=0}^n \binom{n}{k} A^{n-k} c^k$$

Xét: $(a + b)^{n-k} = \sum_{i=0}^{n-k} \binom{n-k}{i} a^i b^{n-k-i} \quad (1)$

Thay (1) vào S_n :

$$S_n = \sum_{k=0}^n \left[\binom{n}{k} c^k \sum_{i=0}^{n-k} \binom{n-k}{i} a^i b^{n-k-i} \right] = \sum_{k=0}^n \left[\sum_{i=0}^{n-k} \binom{n}{k} \binom{n-k}{i} a^i b^{n-k-i} c^k \right] \quad (2)$$

Đặt $j = n - k - i$. Ta có: $i + j + k = i + n - k - i + k = n \quad (3)$

Lại có: $\binom{n}{k} \binom{n-k}{i} = \frac{n!}{k!(n-k)!} \cdot \frac{(n-k)!}{i!(n-k-i)!} = \frac{n!}{i! \cdot j! \cdot k!} \quad (4)$

Từ (2), (3), (4) suy ra:

$$S_n = (a + b + c)^n = \sum_{i+j+k=n} \frac{n!}{i! \cdot j! \cdot k!} a^i b^j c^k \quad (i, j, k \geq 0)$$

2.1.2 Tính $(a + b + c + d)^n$

$$S_n = (a + b + c + d)^n = \sum_{r=0}^n \binom{n}{r} (a + b + c)^{n-r} d^r$$

Áp dụng cách tính $(a + b + c)^n$ đã chứng minh ở bài trước, với $i, j, k \geq 0$:

$$S_n = \sum_{r=0}^n \left\{ \binom{n}{r} \left[\sum_{i+j+k=n-r} \frac{(n-r)!}{i! \cdot j! \cdot k!} a^i b^j c^k \right] d^r \right\} = \sum_{r=0}^n \left[\sum_{i+j+k=n-r} \binom{n}{r} \frac{(n-r)!}{i! \cdot j! \cdot k!} a^i b^j c^k d^r \right]$$

Đặt $l = r$. Do $i + j + k = n - r$ nên $i + j + k + l = n$

Xét: $\binom{n}{r} \cdot \frac{(n-r)!}{i! \cdot j! \cdot k!} = \frac{n!}{r!(n-r)!} \cdot \frac{(n-r)!}{i! \cdot j! \cdot k!} = \frac{n!}{i! \cdot j! \cdot k! \cdot l!} \quad (\text{do } l = r = n - (i + j + k))$

Thay vào S_n :

$$S_n = \sum_{i+j+k+l=n} \frac{n!}{i! \cdot j! \cdot k! \cdot l!} a^i b^j c^k d^l$$

2.2 Dạng thức tổ hợp

2.2.1 Bài toán 28

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/Baitoan28.cpp

$$\sum_{i=1}^n i C_n^i = n 2^{n-1} \quad (*)$$

Một số công thức:

$$(C1) \quad \binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}$$

$$(C2) \quad \sum_{i=0}^n \binom{n}{i} = 2^n$$

a) Quy nạp toán học

- Với $n=1$: $C_1^1 = 1 \cdot 2^0 = 1$
 - Giả sử $(*)$ đúng với $n = k$. Ta chứng minh $(*)$ đúng với $n = k+1$, tức $\sum_{i=1}^{k+1} i C_{k+1}^i = (k+1) 2^k$
- Thật vậy:

$$\sum_{i=1}^{k+1} i \binom{k+1}{i} = \sum_{i=1}^k i \binom{k+1}{i} + (k+1) \binom{k+1}{k+1} = \sum_{i=1}^k i \binom{k+1}{i} + (k+1) \quad (1)$$

Thay $\binom{k+1}{i} = \binom{k}{i} + \binom{k}{i-1}$ vào (1), ta được:

$$\sum_{i=1}^k i \left(\binom{k}{i} + \binom{k}{i-1} \right) + (k+1) = \sum_{i=1}^k i \binom{k}{i} + \sum_{i=1}^k i \binom{k}{i-1} + (k+1) \quad (2)$$

Xét $\sum_{i=1}^k i \binom{k}{i} = k 2^{k-1}$ (Giả thiết quy nạp) (3)

Xét

$$\sum_{i=1}^k i \binom{k}{i-1} = \sum_{j=0}^{k-1} (j+1) \binom{k}{j} = \sum_{j=0}^{k-1} j \binom{k}{j} + \sum_{j=0}^{k-1} \binom{k}{j} = \left(k 2^{k-1} - k \binom{k}{k} \right) + \left(2^k - \binom{k}{k} \right)$$

$$= (k 2^{k-1} - k) + (2^k - 1) \quad (4)$$

Thay (3), (4) vào (2), ta được:

$$k 2^{k-1} + k 2^{k-1} - k + 2^k - 1 + k + 1 = 2k 2^{k-1} + 2^k = (k+1) 2^k$$

Vậy theo nguyên lý quy nạp, ta được đpcm.

b) Biến đổi số hạng tổng quát

- Ta có:

$$i \binom{n}{i} = i \cdot \frac{n!}{i!(n-i)!} = n \cdot \frac{n!}{n(i-1)!(n-i)!} = n \cdot \frac{(n-1)!}{(i-1)!(n-i)!} = n \binom{n-1}{i-1}$$

Thay vào (*):

$$\sum_{i=1}^n i \binom{n}{i} = \sum_{i=1}^n n \binom{n-1}{i-1} = n \cdot \sum_{i=1}^n \binom{n-1}{i-1} = n \cdot \sum_{j=0}^{n-1} \binom{n-1}{j} = n2^{n-1} \quad (\text{đpcm})$$

c) Khai triển $(1+x)^n$

- Áp dụng nhị thức Newton:

$$f(x) = (1+x)^n = \sum_{i=0}^n \binom{n}{i} x^i \Rightarrow f'(x) = \sum_{i=0}^n i \binom{n}{i} x^{i-1} = \sum_{i=1}^n i \binom{n}{i} x^{i-1}$$

- Thay $x = 1$:

$$f'(1) = \sum_{i=1}^n i \binom{n}{i} = \left. \frac{d}{dx} (1+x)^n \right|_{x=1} = n(1+x)^{n-1} \Big|_{x=1} = n2^{n-1}$$

- Vậy

$$\sum_{i=1}^n i \binom{n}{i} = n2^{n-1} \quad (\text{đpcm})$$

d) Lý luận tổ hợp

- Gọi S là một tập gồm n phần tử. Ta cần đếm số cách chọn ra một tập con khác rỗng của S và chọn thêm một phần tử trong tập con đó.
- Về trái: $\sum_{i=1}^n i \binom{n}{i}$
 - Với mỗi i từ 1 đến n : Có $\binom{n}{i}$ cách chọn tập con A có đúng i phần tử.
 - Từ tập A , số cách chọn 1 phần tử là i cách.
 - Tổng số cách: $\sum_{i=1}^n i \binom{n}{i}$
- Về phải: $n2^{n-1}$
 - Bài toán tương đương với việc ta đếm số cách chọn 1 phần tử $x \in S$ rồi sau đó chọn tập con $A \subseteq S$ sao cho $x \in A$
 - Số cách chọn $x \in S$ là n cách.
 - Phần còn lại, ta có thể chọn bất cứ tập con nào của $S \setminus \{x\}$, số cách chọn là 2^{n-1}
 - Tổng số cách: $n2^{n-1}$
- Vậy về trái = về phải (đpcm)

2.2.2 Bài toán 29

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/Baitoan29.cpp

$$\sum_{i=0}^n \frac{C_n^i}{i+1} = \frac{2^{n+1} - 1}{n+1} \quad (*)$$

a) Quy nạp toán học:

- Với $n = 1$:

$$\sum_{i=0}^1 \frac{C_n^i}{i+1} = \frac{\binom{1}{0}}{0+1} + \frac{\binom{1}{1}}{1+1} = 1 + \frac{1}{2} = \frac{2^{1+1} - 1}{1+1} = \frac{3}{2}$$

- Giả sử $(*)$ đúng với $n = k$. Ta chứng minh $(*)$ cũng đúng với $n = k+1$

Tức là $\sum_{i=0}^{k+1} \frac{\binom{k+1}{i}}{i+1} = \frac{2^{k+2}-1}{k+2}$

Thật vậy:

$$\sum_{i=0}^{k+1} \frac{\binom{k+1}{i}}{i+1} = \sum_{i=0}^k \frac{\binom{k+1}{i}}{i+1} + \frac{\binom{k+1}{k+1}}{k+2} = \sum_{i=0}^k \frac{\binom{k+1}{i}}{i+1} + \frac{1}{k+2} \quad (1)$$

Thay $\binom{k+1}{i} = \binom{k}{i} + \binom{k}{i-1}$ vào (1), ta được:

$$\sum_{i=0}^k \frac{\binom{k+1}{i}}{i+1} + \frac{1}{k+2} = \sum_{i=0}^k \frac{\binom{k}{i} + \binom{k}{i-1}}{i+1} + \frac{1}{k+2} = \sum_{i=0}^k \frac{\binom{k}{i}}{i+1} + \sum_{i=1}^k \frac{\binom{k}{i-1}}{i+1} + \frac{1}{k+2} \quad (2)$$

Xét $\sum_{i=0}^k \frac{\binom{k}{i}}{i+1} = \frac{2^{k+1}-1}{k+1}$ (Giả thiết quy nạp) (3)

Xét $\sum_{i=1}^k \frac{\binom{k}{i-1}}{i+1} = \sum_{j=0}^{k-1} \frac{\binom{k}{j}}{j+2}$ (4)

Thay (3), (4) vào (2), ta được:

$$\frac{2^{k+1} - 1}{k+1} + \sum_{j=0}^{k-1} \frac{\binom{k}{j}}{j+2} + \frac{1}{k+2} = \frac{2^{k+1} - 1}{k+1} + \sum_{j=0}^{k-1} \frac{\binom{k}{j}}{j+2} + \frac{\binom{k}{k}}{k+2} = \frac{2^{k+1} - 1}{k+1} + \sum_{j=0}^k \frac{\binom{k}{j}}{j+2}$$

b) Biến đổi số hạng tổng quát

- Ta có

$$\frac{1}{i+1} \binom{n}{i} = \frac{1}{i+1} \cdot \frac{n!}{i!(n-i)!} = \frac{1}{n+1} \cdot \frac{(n+1)!}{(i+1)!(n+1-i-1)!} = \frac{1}{n+1} \binom{n+1}{i+1} \quad (1)$$

- Thay (1) vào $(*)$, ta được:

$$\sum_{i=0}^n \frac{C_n^i}{i+1} = \sum_{i=0}^n \frac{1}{n+1} \binom{n+1}{i+1} = \frac{1}{n+1} \sum_{i=0}^n \binom{n+1}{i+1} \quad (2)$$

- Xét

$$\sum_{i=0}^n \binom{n+1}{i+1} = \sum_{j=1}^{n+1} \binom{n+1}{j} = \sum_{j=0}^{n+1} \binom{n+1}{j} - \binom{n+1}{0} = 2^{n+1} - 1 \quad (3)$$

- Thay (3) vào (2), ta được:

$$\frac{1}{n+1} \sum_{i=0}^n \binom{n+1}{i+1} = \frac{1}{n+1} \cdot (2^{n+1} - 1) = \frac{2^{n+1} - 1}{n+1} \quad (\text{đpcm})$$

c) Khai triển $(1+x)^n$

- Áp dụng nhị thức Newton: $(1+x)^n = \sum_{i=0}^n \binom{n}{i} x^i$
- Lấy tích phân từ 0 đến 1 hai vế:

$$\int_0^1 (1+x)^n dx = \int_0^1 \sum_{i=0}^n \binom{n}{i} x^i dx = \sum_{i=0}^n \left[\binom{n}{i} \int_0^1 x^i dx \right] = \sum_{i=0}^n \frac{\binom{n}{i}}{i+1} \quad (\text{Vế trái})$$

- Đặt $u = 1+x$, ta được $du = d(1+x) = dx$. Xét:

$$\int_0^1 (1+x)^n dx = \int_1^2 u^n du = \frac{u^{n+1}}{n+1} \Big|_1^2 = \frac{2^{n+1}}{n+1} - \frac{1}{n+1} = \frac{2^{n+1} - 1}{n+1} \quad (\text{Vế phải})$$

- Vậy

$$\sum_{i=0}^n \frac{C_n^i}{i+1} = \frac{2^{n+1} - 1}{n+1} \quad (\text{đpcm})$$

d) Lý luận tổ hợp

2.2.3 Bài toán 30

https://github.com/vntanh1406/Graph_SUM2025/blob/main/Combinatorics/Baitoan30.cpp

$$\sum_{i=0}^n (C_n^i)^2 = C_{2n}^n$$

a) Khai triển nhị thức Newton

- Xét $(x+1)^{2n} = \sum_{i=0}^{2n} \binom{2n}{i} x^i$, hệ số của x^n là: $\binom{2n}{n}$
- Xét $(x+1)^{2n} = (x+1)^n (x+1)^n = (\sum_{i=0}^n \binom{n}{i} x^i) \cdot (\sum_{j=0}^n \binom{n}{j} x^j)$, hệ số của x^n là:

$$\sum_{\substack{i+j=n \\ 0 \leq i, j \leq n}} \binom{n}{i} \cdot \binom{n}{j} = \sum_{i=0}^n \binom{n}{i} \binom{n}{n-i} = \sum_{i=0}^n \binom{n}{i}^2$$

- Do hệ số của x^n từ cả hai cách tính phải bằng nhau nên ta được đpcm.

b) Lý luận tổ hợp

- Đếm số cách chọn n phần tử từ một tập hợp gồm $2n$ phần tử, ta có 2 cách đếm sau.
- Vế trái: Đếm trực tiếp, số cách chọn là $\binom{2n}{n}$
- Vế phải: Ta chia $2n$ phần tử thành 2 tập A và B, mỗi tập có n phần tử. Ta chọn i phần tử từ A và $(n-i)$ phần tử từ B, với $0 \leq i \leq n$. Với mỗi giá trị của i , số cách chọn là $\binom{n}{i} \binom{n}{n-i} = \binom{n}{i}^2$. Vậy tổng cộng có $\sum_{i=0}^n \binom{n}{i}^2$.
- Với 2 cách đếm phải ra cùng một kết quả, nên vế trái bằng vế phải (đpcm)

2.2.4 Bài toán 31

a) Tìm hệ số của x^{26}

$$\left(\frac{1}{x^4} + x^7\right)^n = \sum_{i=0}^n C_n^i \frac{x^{7i}}{x^{4n-4i}} = \sum_{i=0}^n C_n^i x^{11i-4n} \quad (1)$$

Lại có:

$$\begin{aligned} \sum_{i=1}^n C_{2n+1}^i &= \sum_{i=n+1}^{2n} C_{2n+1}^i = \frac{1}{2} \cdot \sum_{i=1}^{2n} C_{2n+1}^i \\ &= \frac{1}{2} \cdot \left(\sum_{i=0}^{2n+1} C_{2n+1}^i - C_{2n+1}^0 - C_{2n+1}^{2n+1} \right) \\ &= \frac{1}{2} \cdot (2^{2n+1} - 2) = 2^{2n} - 1 = 2^{20} - 1 \end{aligned}$$

Vậy $2n = 20 \Leftrightarrow n = 10$ (2)

Thay (2) vào (1), ta cần tìm i để $11i - 4n = 26 \Leftrightarrow 11i = 26 + 4 \cdot 10 = 66 \Leftrightarrow i = 6$
 Vậy hệ số của x^{26} là $C_{10}^6 = 210$

b) Tìm hệ số của x^k

Ta có:

$$\begin{aligned} \sum_{i=1}^n C_{2n+1}^i &= \sum_{i=n+1}^{2n} C_{2n+1}^i = \frac{1}{2} \cdot \sum_{i=1}^{2n} C_{2n+1}^i \\ &= \frac{1}{2} \cdot \left(\sum_{i=0}^{2n+1} C_{2n+1}^i - C_{2n+1}^0 - C_{2n+1}^{2n+1} \right) \\ &= \frac{1}{2} \cdot (2^{2n+1} - 2) = 2^{2n} - 1 = 2^{n_0} - 1 \end{aligned}$$

Vậy với n_0 là số chẵn, $2n = n_0 \Leftrightarrow n = \frac{n_0}{2}$

Xét khai triển:

$$(x^a + x^n)^n = \sum_{i=0}^n C_n^i \cdot x^{ai} \cdot x^{n(n-i)} = \sum_{i=0}^n x^{n^2+i(a-n)}$$

Ta cần tìm i để $n^2 + i(a-n) = k \Leftrightarrow i = \frac{k-n^2}{a-n} = \frac{k-\frac{n_0^2}{4}}{a-\frac{n_0}{2}}$, với $0 \leq i \leq n$

Và $i \in \mathbb{N}$ nên $\left(k - \frac{n_0^2}{4}\right)$ phải chia hết cho $\left(a - \frac{n_0}{2}\right)$

• Nếu $a > \frac{n_0}{2}$:

$$0 \leq \frac{k-\frac{n_0^2}{4}}{a-\frac{n_0}{2}} \leq \frac{n_0}{2} \Leftrightarrow \frac{n_0^2}{4} \leq k \leq \frac{n_0^2}{4} + \frac{n_0}{2} \left(a - \frac{n_0}{2}\right) \Leftrightarrow \frac{n_0^2}{4} \leq k \leq \frac{n_0 a}{2}$$

• Nếu $a < \frac{n_0}{2}$:

$$0 \leq \frac{k-\frac{n_0^2}{4}}{a-\frac{n_0}{2}} \leq \frac{n_0}{2} \Leftrightarrow \frac{n_0^2}{4} \geq k \geq \frac{n_0^2}{4} + \frac{n_0}{2} \left(a - \frac{n_0}{2}\right) \Leftrightarrow \frac{n_0^2}{4} \geq k \geq \frac{n_0 a}{2}$$

- Nếu $a = \frac{n_0}{2}$
 $(x^n + x^n)^n = (2x^n)^n = 2^n x^{n^2} = 2^n x^{\frac{n_0^2}{4}}$
Hệ số của x^k là:
 - 2^n nếu $k = \frac{n_0^2}{4}$
 - 0 nếu $k \neq \frac{n_0^2}{4}$
Vậy hệ số của x^k là:
- Nếu $a \neq \frac{n_0}{2}$: $C_{\frac{n_0}{2}}^i$ với $i = \frac{k - \frac{n_0^2}{4}}{a - \frac{n_0}{2}}$, k thỏa:
 - $\left(k - \frac{n_0^2}{4}\right)$ phải chia hết cho $\left(a - \frac{n_0}{2}\right)$
 - $a > \frac{n_0}{2}$ thì $\frac{n_0^2}{4} \leq k \leq \frac{n_0 a}{2}$
 - $a < \frac{n_0}{2}$ thì $\frac{n_0^2}{4} \geq k \geq \frac{n_0 a}{2}$
- Nếu $a = \frac{n_0}{2}$ thì hệ số của x^k là:
 - 2^n nếu $k = \frac{n_0^2}{4}$
 - 0 nếu $k \neq \frac{n_0^2}{4}$

3 Basic Graph Theory

Link to C++ Sources: https://github.com/vntanh1406/Graph_SUM2025/tree/main/BasicGraphTheory

3.1 Graph representation

- Adjacency Matrix to Edge List : https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/AdjacencyMatrixToEdgeList.cpp
- Adjacency Matrix to Adjacency List: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/AdjacencyMatrixToAdjacencyList.cpp
- Edge List to Adjacency Matrix: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/EdgeListToAdjacencyMatrix.cpp
- Edge List to Adjacency List: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/EdgeListToAdjacencyList.cpp
- Adjacency List to Adjacency Matrix: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/AdjacencyListToAdjacencyMatrix.cpp
- Adjacency List To Edge List: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/AdjacencyListToEdgeList.cpp

3.2 Search Algorithm

3.2.1 Basic DFS & BFS

- Basic Depth First Search: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/BasicDFS.cpp
- Basic Breadth First Search: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/BasicBFS.cpp
- Counting Connected Components: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/CountingConnectedComponents.cpp
- Find Path From **s** to **e**: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/FindPath_Basic.cpp

3.2.2 DFS & BFS for grid

- Counting Connected Components and Checking Path Existence: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/BFS_DFS_OnGrid.cpp
- Find The Shortest Path: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/FindShortestPath.cpp

3.2.3 Important algorithms

- Topological Sort: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/TopologicalSort.cpp
- Detect Cycles in Undirected Graph: https://github.com/vntanh1406/Graph_SUM2025/blob/main/BasicGraphTheory/UndirectedGraphCycle.cpp

4 CSES Problem List

4.1 Graph Algorithms

4.1.1 Counting Rooms

Problem Description

- <https://cses.fi/problemset/task/1192>
- The task is to determine the number of rooms in a building.
- A room is defined as a maximal connected area of floor tiles (denoted by `'.'`) in a 2D map.
- We can move up, down, left, and right between adjacent floor tiles. Wall tiles are represented by `'#'` and cannot be walked through.
- **Input:**
 - The first line contains two integers n and m ($1 \leq n, m \leq 1000$), denoting the height and width of the map.
 - The next n lines each contain a string of m characters representing the map.

- **Output:** The number of distinct rooms.
- **Example**

Input	Output
<pre> 5 8 ##### #..#...# ####.#.# #..#...# ##### </pre>	3

Algorithm Explanation

- <https://cses.fi/paste/6fa005ccb9388ed1c2d9a9/>
- Use DFS to find the number of connected components in a grid where we treat floor tiles '.' as nodes in a graph. (2 floor tiles are connected if they are adjacent: up/down/left/right).
- Implementation:
 - **n, m**: the dimensions of the grid (height and width).
 - **grid**: a vector of strings representing the map, where each character is either '.' (floor) or '#' (wall).
 - **visited[n][m]**: a 2D boolean array used to track visited floor tiles.
 - **dx[4], dy[4]**: Two arrays representing the relative movement in four directions:
 - * Up: $(-1, 0)$
 - * Down: $(+1, 0)$
 - * Left: $(0, -1)$
 - * Right: $(0, +1)$
 - **void dfs(int x, int y)**
 - * Given a starting floor tile at position (x, y) :
 1. Mark **visited[x][y] = true**.
 2. For each of the 4 directions:
 - New coordinates $(nx, ny) = (x + dx[i], y + dy[i])$.
 - If (nx, ny) is within bounds, not visited, and is a floor tile, recursively call **dfs(nx, ny)**.
 - **Main loop:**
 - * Iterate over all grid cells.
 - * For each unvisited floor tile:
 - Call DFS from that tile to explore its connected room.
 - Increment the room counter.
- **Time complexity:** The algorithm visits each cell at most once, and each DFS runs in time proportional to the number of floor tiles in a room. Hence, the overall complexity is $\mathcal{O}(n \cdot m)$.

- **Step-by-Step Example** (using the sample input): We visualize the grid and track DFS visits:

Initial Grid:

#	#	#	#	#	#	#	#
#	.	.	#	.	.	.	#
#	#	#	#	.	#	.	#
#	.	.	#	.	.	.	#
#	#	#	#	#	#	#	#

Room 1: Start DFS at (1,1)

DFS visits: (1,1) → (1,2)

→ Room 1 completed. Total rooms = 1

Room 2: Start DFS at (1,4)

DFS visits: (1,4) → (2,4) → (3,4) → (3,5) → (3,6) → (2,6) → (1,6) → (1, 5)

→ Room 2 completed. Total rooms = 2

Room 3: Start DFS at (3,1)

DFS visits: (3,1) → (3,2)

→ Room 3 completed. Total rooms = 3

4.1.2 Labyrinth

Problem Description

- <https://cses.fi/problemset/task/1193>
- There is a map of a labyrinth, and we need to find the shortest path from a start point A to an endpoint B.
- Movement is allowed in four directions: up, down, left, right.
- The labyrinth is represented by a grid:
 - '.' denotes an empty tile (floor).
 - '#' denotes a wall.
 - 'A' is the starting point.
 - 'B' is the target.
- **Input:**
 - The first line contains two integers n and m ($1 \leq n, m \leq 1000$), the dimensions of the map.
 - The next n lines contain m characters each, describing the map.
- **Output:**
 - First print "YES" if a path exists, and "NO" otherwise.
 - If a path exists, print its length and then a string consisting of the steps: L, R, U, D.

- **Example**

Input	Output
<pre> 5 8 ##### #.A#...# #.#.#B# #.....# ##### </pre>	<pre> YES 9 LDDRRRRRU </pre>

Algorithm Explanation:

- <https://cses.fi/paste/261fd1d6d36a0c05c32743/>
- **Initialize:**
 - `visited[i][j] = false` for all cells
 - `d[i][j] = 0` (distance from 'A')
 - `parent[i][j] = previous cell in path`
- **BFS(start):**
 - Enqueue start cell, mark as visited
 - While queue not empty:
 - * Dequeue (x, y)
 - * For each direction (U, L, R, D): If neighbor (nx, ny) is valid and unvisited:
 - Mark visited, set parent, update distance
 - If cell is 'B': stop search
- **Trace Path:**
 - If distance to 'B' is 0: print "NO"
 - Else:
 - * Backtrack from 'B' to 'A' using `parent`
 - * Record directions (U, L, R, D)
 - * Reverse the path and print "YES", distance, and path
- **Time Complexity:** $\mathcal{O}(n \cdot m)$

4.1.3 Building Roads

Problem Description

- <https://cses.fi/problemset/task/1666>
- Given n cities and m roads, determine the minimum number of new roads required to connect all cities, and specify which roads to build. Each existing road connects two different cities.
- **Input:**

- The first line contains two integers n and m ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$) (the number of cities and existing roads).
- The next m lines contain two integers a and b ($1 \leq a, b \leq n$) (meaning there is a road between cities a and b).

- **Output:**

- First, print an integer k (the minimum number of new roads needed).
- Then print k lines, each containing two integers u and v , indicating a road to build between cities u and v .
- Any valid solution is accepted.
- **Example**

Input	Output
4 2 1 2 3 4	1 2 3

Algorithm Explanation

- <https://cses.fi/paste/bdeee055189e59e5c2f5d8/>
- Use DFS to find all connected components.
- For each new component found, save a representative city.
- To connect the components:
 - If there are k components, we need $k - 1$ roads.
 - Connect representative cities linearly: $res[i]$ with $res[i + 1]$.
- **Time Complexity:** $\mathcal{O}(n + m)$

4.1.4 Message Routes

Problem Description

- <https://cses.fi/problemset/task/1667>
- n computers and m connections.
- Each connection links two distinct computers directly.
- Check if there is a path exists from computer 1 to computer n , find the minimum number of computers on the route, and output one such route.
- **Input:**
 - The first line contains two integers n and m
 - Then follow m lines, each with two integers a and b : there is a connection between computers a and b .
- **Output:**

- If there exists a route from computer 1 to computer n , first print k , then print k space-separated integers representing the computers along this path.
- If no such route exists, print IMPOSSIBLE.

- **Example**

Input	Output
5 5 1 2 1 3 1 4 2 3 5 4	3 1 4 5

Algorithm Explanation

- <https://cses.fi/paste/4e4274bd1cea72b6c2f69d/>
- The problem is to find the shortest path from node 1 to node n in an undirected graph.
- Use BFS starting from node 1 to:
 - Mark visited nodes (`visited[i]`).
 - Record the parent of each node during traversal (`parent[i]`) to reconstruct the path.
 - Count the number of steps from the start node to each node (`d[i]`).
- After BFS:
 - If `visited[n]` is false, there is no path from 1 to n , so the output is IMPOSSIBLE.
 - Otherwise, we reconstruct the shortest path using the `parent[]` array starting from node n back to 1.
 - Finally, we print the path length (`d[n] + 1`, because the path includes both end-points), and the path in correct order.
- **Time Complexity:** $\mathcal{O}(n + m)$

4.1.5 Building Teams

Problem Description

- <https://cses.fi/problemset/task/1668/>

Algorithm Explanation

- <https://cses.fi/paste/4ac88035ca891502c2f78e/>

4.1.6 Round Trip

Problem Description

- <https://cses.fi/problemset/task/1669>

Algorithm Explanation

- <https://cses.fi/paste/c716877ebfb8afaec307d9/>

4.1.7 Monsters

Problem Description

- <https://cses.fi/problemset/task/1194>

Algorithm Explanation

- <https://cses.fi/paste/1a35c0381d423f67c3084e/>

4.1.8 Flight Routes

Problem Description

- <https://cses.fi/problemset/task/1196/>

Algorithm Explanation

- <https://cses.fi/paste/c4a13f10d301c5ffc45604/>