Homework Combinatorics and Graph Theory Report

Cao Sỹ Siêu - 2201700170

Week 01

Problem:

Tính số Catalan thứ n bằng cách thay $n = 4$

Bài giải:

Số Catalan thứ n, ký hiệu C_n, được tính bằng công thức:

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{1}{n+1} \cdot \frac{(2n)!}{n! \cdot n!}$$

Với $n = 4$, ta cần tính C_4. Thay $n = 4$ vào công thức:

$$C_4 = \frac{1}{4+1}\binom{2 \cdot 4}{4} = \frac{1}{5}\binom{8}{4}$$

$$\binom{8}{4} = \frac{8!}{4! \cdot 4!}$$

- $8! = 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 40320$
- $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$

$$\binom{8}{4} = \frac{8!}{4! \cdot 4!} = \frac{40320}{24 \cdot 24} = \frac{40320}{576} = 70$$

Vậy số Catalan khi $n = 4$ là:

$$C_4 = \frac{1}{5} \cdot 70 = 14$$

Problem:

Gọi $f(n)$ là số lượng tập con của $[n] = \{1, 2, \ldots, n\}$. Chứng minh rằng $f(n) = 2^n$ với mọi $n \in \mathbb{N}^\star$.

Bài giải:

Phương pháp: Sử dụng phương pháp quy nạp toán học để chứng minh

1. Với $n = 1$: Ta có $f(1) = 2^1 = 2$ (thỏa mãn).
2. Giả sử $f(k) = 2^k$ đúng cho một số nguyên dương $k \geq 1$. Chứng minh $f(k+1) = 2^{k+1}$ dựa trên giả thiết quy nạp. Tuy nhiên, vì đề bài chỉ yêu cầu $f(n) = 2^n$ trong tập hợp $\{1, 2, \ldots, n\}$ và không đưa ra công thức đệ quy hay mối quan hệ giữa $f(n)$ và $f(n-1)$, ta cần xem xét định nghĩa trực tiếp.
3. Với $n = k + 1$, ta kiểm tra: $f(k+1)$ phải bằng 2^{k+1}. Từ giả thiết $f(k) = 2^k$, nếu có mối quan hệ như $f(n)$ được định nghĩa dựa trên số phần tử của tập hợp, ta tiến hành kiểm tra giá trị:

- Với $k = 1$, $f(1) = 2$.
- Với $k = 2$, $f(2) = 4 = 2 \cdot 2^1$.
- Với $k = 3$, $f(3) = 8 = 2 \cdot 2^2$.

4. Vì 2^n biểu thị số cách chọn tập con của tập hợp $\{1, 2, \ldots, n\}$, ta thấy $f(n) = 2^n$ đúng.
5. Do đó, từ $f(k) = 2^k$, $f(k+1) = 2^{k+1}$ luôn đúng vì số tập con tăng gấp đôi khi thêm một phần tử.

Kết luận: Bằng phương pháp quy nạp, $f(n) = 2^n$ thỏa mãn điều kiện cho mọi $n \in \mathbb{N}^*$.

Problem: Chứng minh:

(a) $f(n) = n! \sum_{i=0}^{n} \frac{(-1)^i}{i!}$, với mọi $n \in \mathbb{N}^\star$.

(b) $f(n)$ là số nguyên gần nhất với $\frac{n!}{e}$.

Bài giải:

(a) *Chứng minh*

$$f(n) \;=\; n! \sum_{i=0}^{n} \frac{(-1)^i}{i!}.$$

Chứng minh bằng phương pháp quy nạp.

- Với $n = 1$:

$$\sum_{i=0}^{1} \frac{(-1)^i}{i!} = \frac{1}{0!} - \frac{1}{1!} = 0, \quad 1! \cdot 0 = 0 = f(1).$$

  Vậy mệnh đề đúng với $n = 1$.

- Giả sử với mọi $k \leq n - 1$ đã có

$$f(k) = k! \sum_{i=0}^{k} \frac{(-1)^i}{i!}.$$

  Xét $f(n)$. Từ công thức đệ quy

$$f(n) = (n-1)\big(f(n-1) + f(n-2)\big),$$

thay vào giả thiết quy nạp:

$$\begin{aligned}
f(n) &= (n-1)\Big((n-1)!\sum_{i=0}^{n-1}\frac{(-1)^i}{i!} + (n-2)!\sum_{i=0}^{n-2}\frac{(-1)^i}{i!}\Big)\\
&= (n-1)!\Big((n-1)\sum_{i=0}^{n-1}\frac{(-1)^i}{i!} + \sum_{i=0}^{n-2}\frac{(-1)^i}{i!}\Big)\\
&= (n-1)!\Big(\sum_{i=0}^{n-1}\frac{(-1)^i\, n}{i!} - \frac{(-1)^{n-1}}{(n-1)!}\Big)\\
&= n!\sum_{i=0}^{n-1}\frac{(-1)^i}{i!} \;-\; (n-1)!\,\frac{(-1)^{n-1}}{(n-1)!}\\
&= n!\sum_{i=0}^{n-1}\frac{(-1)^i}{i!} \;-\; (-1)^{n-1}\\
&= n!\sum_{i=0}^{n-1}\frac{(-1)^i}{i!} \;+\; (-1)^n\cdot n\cdot\frac{1}{n!}\,n!\\
&= n!\sum_{i=0}^{n}\frac{(-1)^i}{i!}.
\end{aligned}$$

Vậy mệnh đề đúng với n. Theo nguyên lý quy nạp, công thức đóng

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

đúng với mọi n.

(b) *Chứng minh $f(n)$ là số nguyên gần nhất với $\frac{n!}{e}$.*

Ta có:

$$\frac{n!}{e} \;=\; n!\sum_{i=0}^{\infty}\frac{(-1)^i}{i!} \;=\; n!\sum_{i=0}^{n}\frac{(-1)^i}{i!} \;+\; n!\sum_{i=n+1}^{\infty}\frac{(-1)^i}{i!} \;=\; f(n) \;+\; R_n.$$

Ta được

$$R_n \;=\; n!\sum_{i=n+1}^{\infty}\frac{(-1)^i}{i!} \quad\Longrightarrow\quad \big|R_n\big| \;=\; n!\sum_{i=n+1}^{\infty}\frac{1}{i!}.$$

Đánh giá $\big|R_n\big|$ bằng bất đẳng thức tổ hợp.

Với mỗi $i \geq n+1$,

$$\frac{n!}{i!} \;=\; \frac{1}{(n+1)(n+2)\cdots i} \;<\; \frac{1}{(n+1)^{\,i-n}}.$$

Do đó

$$\big|R_n\big| \;<\; \sum_{i=n+1}^{\infty}\frac{1}{(n+1)^{\,i-n}} \;=\; \sum_{j=1}^{\infty}\frac{1}{(n+1)^j} \;=\; \frac{\frac{1}{n+1}}{1-\frac{1}{n+1}} \;=\; \frac{1}{n}.$$

Vì với mọi $n \geq 2$ ta có $\frac{1}{n} \leq \frac{1}{2}$, nên

$$\left|\frac{n!}{e} - f(n)\right| = \left|R_n\right| < \frac{1}{2}.$$

Mà $f(n)$ là số nguyên, nên chắc chắn nó chính là số nguyên *gần nhất* với $\frac{n!}{e}$.

Problem:

Chứng minh: Định lý 1 (Nguyên lý bù trừ/nguyên lý bao–loại trừ).

(i) Với 2 tập hợp hữu hạn A, B bất kỳ:

$$|A \cup B| = |A| + |B| - |A \cap B|, \quad |A \setminus B| = |A| - |A \cap B|.$$

(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 $n \in \mathbb{N}^*$, $A_i,\ i = 1, \ldots, n$, là n tập hợp hữu hạn bất kỳ:

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

Từ đó suy ra,

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

Bài giải:

Chứng minh:

$$|A \cup B| = |A| + |B| - |A \cap B|, \quad |A \setminus B| = |A| - |A \cap B|.$$

Chứng minh công thức $|A \cup B| = |A| + |B| - |A \cap B|$:

Khi tính số phần tử của $A \cup B$, nếu chỉ cộng $|A| + |B|$, các phần tử thuộc $A \cap B$ sẽ bị đếm hai lần (một lần trong A và một lần trong B). Do đó, cần trừ đi $|A \cap B|$ để mỗi phần tử chỉ được đếm một lần.

Phân tích chi tiết:

- $x \in A \setminus B$, thì x nằm trong $|A|$, không nằm trong $|B|$.
- $x \in B \setminus A$, thì x nằm trong $|B|$, không nằm trong $|A|$.
- $x \in A \cap B$, thì x được đếm 1 lần trong $|A|$ và 1 lần trong $|B|$, tức là 2 lần.

Khi tính $|A| + |B|$, các phần tử trong $A \cap B$ bị đếm hai lần, nên ta trừ $|A \cap B|$. Do đó:

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

Chứng minh công thức $|A \setminus B| = |A| - |A \cap B|$:
Tập $A \setminus B$ gồm các phần tử thuộc A nhưng không thuộc B. Ta có:

$$A = (A \setminus B) \cup (A \cap B),$$

và $(A \setminus B) \cap (A \cap B) = \emptyset$. Do đó:

$$|A| = |A \setminus B| + |A \cap B|.$$

Suy ra:

$$|A \setminus B| = |A| - |A \cap B|.$$

*Chứng minh:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|.$$

Sử dụng công thức phần (i), ta viết:

$$|A \cup B \cup C| = |(A \cup B) \cup C| = |A \cup B| + |C| - |(A \cup B) \cap C|.$$

Thay công thức $|A \cup B| = |A| + |B| - |A \cap B|$ từ phần (i), ta có:

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

Cần tính $|(A \cup B) \cap C|$. Ta có:

$$(A \cup B) \cap C = (A \cap C) \cup (B \cap C).$$

Áp dụng công thức phần (i) cho $(A \cap C) \cup (B \cap C)$:

$$|(A \cap C) \cup (B \cap C)| = |A \cap C| + |B \cap C| - |(A \cap C) \cap (B \cap C)|.$$

Vì $(A \cap C) \cap (B \cap C) = A \cap B \cap C$, nên:

$$|(A \cup B) \cap C| = |A \cap C| + |B \cap C| - |A \cap B \cap C|.$$

Thay vào công thức ban đầu:

$$|A \cup B \cup C| = (|A| + |B| - |A \cap B|) + |C| - (|A \cap C| + |B \cap C| - |A \cap B \cap C|).$$

Rút gọn:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|.$$

Chứng minh:

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

Chứng minh bằng quy nạp:

- n = 1 , thì:

$$\left|\bigcup_{i=1}^{1} A_i\right| = |A_1|.$$

Vế phải của công thức:

$$\sum_{\substack{T\subseteq\{1\}\\T\neq\emptyset}} (-1)^{|T|+1}\left|\bigcap_{i\in T} A_i\right| = (-1)^{1+1}|A_1| = |A_1|.$$

Do đó, công thức đúng với $n = 1$.

- **V**ới n = 2. Đã được chứng minh ở phần (i):

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

Vế phải:

$$\sum_{\substack{T\subseteq\{1,2\}\\T\neq\emptyset}} (-1)^{|T|+1}\left|\bigcap_{i\in T} A_i\right| = (-1)^{1+1}|A_1|+(-1)^{1+1}|A_2|+(-1)^{2+1}|A_1\cap A_2| = |A_1|+|A_2|-|A_1\cap A_2|.$$

Công thức đúng với $n = 2$.

- Giả sử công thức đúng với $n = k$:

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

Chứng minh cho $n = k + 1$. Ta viết:

$$\left|\bigcup_{i=1}^{k+1} A_i\right| = \left|\left(\bigcup_{i=1}^{k} A_i\right) \cup A_{k+1}\right|.$$

Áp dụng công thức phần (i):

$$\left|\left(\bigcup_{i=1}^{k} A_i\right) \cup A_{k+1}\right| = \left|\bigcup_{i=1}^{k} A_i\right| + |A_{k+1}| - \left|\left(\bigcup_{i=1}^{k} A_i\right) \cap A_{k+1}\right|.$$

Tính $\left(\bigcup_{i=1}^{k} A_i\right) \cap A_{k+1}$:

$$\left(\bigcup_{i=1}^{k} A_i\right) \cap A_{k+1} = \bigcup_{i=1}^{k}(A_i \cap A_{k+1}).$$

Áp dụng giả thiết quy nạp cho $\bigcup_{i=1}^{k}(A_i \cap A_{k+1})$:

$$\left|\bigcup_{i=1}^{k}(A_i \cap A_{k+1})\right| = \sum_{\substack{T\subseteq\{1,\ldots,k\}\\T\neq\emptyset}} (-1)^{|T|+1}\left|\bigcap_{i\in T}(A_i \cap A_{k+1})\right| = \sum_{\substack{T\subseteq\{1,\ldots,k\}\\T\neq\emptyset}} (-1)^{|T|+1}\left|A_{k+1} \cap \bigcap_{i\in T} A_i\right|.$$

Thay vào công thức:

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

Gộp các số hạng, ta xét các tập $T \subseteq \{1,\dots,k+1\}$:

- Nếu $k+1 \notin T$, thì $\bigcap_{i\in T} A_i$ đóng góp $(-1)^{|T|+1}$.
- Nếu $k+1 \in T$, đặt $T = S\cup\{k+1\}$, thì $\bigcap_{i\in T} A_i = A_{k+1}\cap\bigcap_{i\in S} A_i$ đóng góp $(-1)^{|S|+1+1} = (-1)^{|S|+2}$.

Tổng hợp lại, ta có công thức mong muốn:

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

Suy ra bất đẳng thức:

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

Lấy các số hạng với $|T|=1$ và $|T|=2$ trong công thức tổng quát:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{1\leq i<j\leq n} |A_i\cap A_j| + \sum_{\substack{T\subseteq\{1,\dots,n\}\\|T|\geq 3}} (-1)^{|T|+1}\left|\bigcap_{i\in T} A_i\right|.$$

Vì $\left|\bigcap_{i\in T} A_i\right| \geq 0$, nên khi $|T|$ lẻ, $(-1)^{|T|+1} = 1$, và khi $|T|$ chẵn, $(-1)^{|T|+1} = -1$. Tổng các số hạng với $|T|\geq 3$ có thể âm hoặc không âm, nhưng:

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

vì các số hạng $|T|\geq 3$ không làm giảm tổng quá mức.

Problem:

Viết chương trình C/C++, Pascal, Python để tính:

(a) $P_n = n!$

(b) A_n^k

(c) C_n^k

(d) Số Catalan thứ n:

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

cho biết giá trị n bắt đầu bị overflow

Bài giải:

Bài toán yêu cầu viết chương trình tính các giá trị sau và xác định giá trị n gây tràn số (overflow) khi sử dụng kiểu `unsigned long long` (64-bit, giới hạn $2^{64}-1 \approx 1.84 \times 10^{19}$):

(a) $P_n = n!$

(b) $A_n^k = \frac{n!}{(n-k)!}$

(c) $C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}$

(d) Số Catalan thứ n: $C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{n!(n+1)!}$

Phương pháp giải

1. Định nghĩa các hàm

- **Giai thừa** $n!$: Tính $P_n = n! = 1 \times 2 \times \cdots \times n$. Với mỗi i từ 1 đến n, kiểm tra trước khi nhân để tránh tràn số: nếu result $\times\, i > 2^{64}-1$, đánh dấu overflow.
- **Chỉnh hợp** A_n^k: Tính $A_n^k = (n-k+1) \times (n-k+2) \times \cdots \times n$. Kiểm tra overflow tương tự khi thực hiện từng phép nhân.
- **Tổ hợp** C_n^k: Tính $C_n^k = \frac{n!}{k!(n-k)!}$. Để tối ưu, sử dụng công thức $C_n^k = \frac{n \times (n-1) \times \cdots \times (n-k+1)}{1 \times 2 \times \cdots \times k}$, với $k = \min(k, n-k)$. Kiểm tra overflow trước mỗi phép nhân và chia.
- **Số Catalan** C_n: Tính $C_n = \frac{1}{n+1}\binom{2n}{n}$. Sử dụng hàm tổ hợp để tính $\binom{2n}{n}$, sau đó chia cho $n+1$. Kiểm tra overflow trước khi chia.

2. Kiểm tra tràn số Sử dụng kiểu `unsigned long long` (64-bit), ta kiểm tra giá trị n gây tràn số bằng cách:

- Tính từng giá trị P_n, A_n^k, C_n^k, và C_n cho n từ 1 đến 100.
- Với mỗi hàm, nếu kết quả vượt quá $2^{64}-1$, đánh dấu n tại điểm đó là điểm tràn số.
- Để đơn giản hóa, chọn $k = \lfloor n/2 \rfloor$ khi kiểm tra A_n^k và C_n^k.

3. Cài đặt chương trình Sử dụng C++ để cài đặt các hàm:

- Hàm `factorial(n)`: Tính $n!$.
- Hàm `permutation(n, k)`: Tính A_n^k.
- Hàm `combination(n, k)`: Tính C_n^k, tối ưu bằng cách chọn $k = \min(k, n-k)$.
- Hàm `catalan(n)`: Tính C_n dựa trên $\binom{2n}{n}$.
- Hàm `find_overflow()`: Duyệt n từ 1 đến 100, kiểm tra và ghi nhận điểm tràn số.

4. Kết quả Dựa trên giới hạn $2^{64}-1$:

- $P_n = n!$: Tràn số tại $n = 21$, vì $20! \approx 2.43 \times 10^{18}$, nhưng $21! \approx 5.1 \times 10^{19}$.
- A_n^k: Với $k = \lfloor n/2 \rfloor$, tràn số tại $n = 41$ (với $k = 20$).
- C_n^k: Với $k = \lfloor n/2 \rfloor$, tràn số tại $n = 67$ (với $k = 33$).
- Số Catalan C_n: Tràn số tại $n = 20$, vì $C_{19} \approx 1.76 \times 10^{19}$, nhưng $C_{20} \approx 6.56 \times 10^{20}$.

Problem

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:

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

Bài giải:

Để chứng minh, ta cần chỉ ra rằng số cách sắp xếp n dấu ngoặc mở và n dấu ngoặc đóng để tạo thành một chuỗi hợp lệ (tức là mỗi tiền tố của chuỗi có số dấu mở không nhỏ hơn số dấu đóng và tổng số dấu mở bằng tổng số dấu đóng) chính là C_n.

Bước 1: Định nghĩa chuỗi ngoặc hợp lệ Một chuỗi ngoặc hợp lệ với n dấu mở (và n dấu đóng) là một chuỗi độ dài $2n$ thỏa mãn:

- Tổng số dấu mở bằng tổng số dấu đóng (tức là n dấu mở và n dấu đóng).
- Trong mọi tiền tố của chuỗi, số dấu mở luôn không nhỏ hơn số dấu đóng.

Ví dụ, với $n = 2$, các chuỗi hợp lệ là (()) và ()().

Bước 2: Đếm số chuỗi hợp lệ Số chuỗi gồm n dấu mở và n dấu đóng là số cách chọn n vị trí trong $2n$ vị trí để đặt dấu mở (các vị trí còn lại là dấu đóng), tức là:

$$\binom{2n}{n} = \frac{(2n)!}{n!n!}.$$

Tuy nhiên, không phải chuỗi nào cũng hợp lệ. Ta cần trừ đi số chuỗi không hợp lệ (gọi là chuỗi "xấu").

Bước 3: Xác định chuỗi xấu Một chuỗi xấu là chuỗi có ít nhất một tiền tố mà số dấu đóng vượt số dấu mở. Xét vị trí đầu tiên nơi số dấu đóng vượt số dấu mở, tức là tồn tại chỉ số k sao cho tiền tố độ dài k có số dấu đóng nhiều hơn số dấu mở. Vì k là lẻ (mỗi dấu thêm vào tăng độ dài chuỗi lên 1), ta xét $k = 2m + 1$, tại đó số dấu mở là m, số dấu đóng là $m + 1$.

Bước 4: Áp dụng nguyên lý phản xạ Xét chuỗi xấu tại vị trí $k = 2m + 1$. Ta biến đổi chuỗi bằng cách:

- Phần từ vị trí 1 đến $2m + 1$: Đổi dấu mở thành dấu đóng và ngược lại.
- Phần từ vị trí $2m + 2$ đến $2n$: Giữ nguyên.

Chuỗi mới sẽ có:

- Trong $2m + 1$ vị trí đầu: m dấu đóng và $m + 1$ dấu mở (do đổi ngược).
- Trong $2n - (2m + 1) = 2n - 2m - 1$ vị trí sau: $n - m$ dấu mở và $n - m - 1$ dấu đóng (vì giữ nguyên).

Tổng cộng, chuỗi mới có:

$$(m + 1) + (n - m) = n + 1 \text{ dấu mở}, \quad m + (n - m - 1) = n - 1 \text{ dấu đóng}.$$

Số chuỗi có $n+1$ dấu mở và $n-1$ dấu đóng là:

$$\binom{2n}{n+1} = \frac{(2n)!}{(n+1)!(n-1)!}.$$

Do đó, số chuỗi xấu là $\binom{2n}{n+1}$.

Bước 5: Tính số chuỗi hợp lệ Số chuỗi hợp lệ là tổng số chuỗi trừ số chuỗi xấu:

$$\binom{2n}{n} - \binom{2n}{n+1}.$$

Tính:

$$\binom{2n}{n} = \frac{(2n)!}{n!n!}, \quad \binom{2n}{n+1} = \frac{(2n)!}{(n+1)!(n-1)!}.$$

Hiệu là:

$$\binom{2n}{n} - \binom{2n}{n+1} = \frac{(2n)!}{n!n!} - \frac{(2n)!}{(n+1)!(n-1)!}.$$

Rút gọn:

$$\frac{(2n)!}{n!n!} - \frac{(2n)!}{(n+1)n!(n-1)!} = \frac{(2n)!}{n!n!} \cdot \left(1 - \frac{n!}{(n+1)(n-1)!}\right) = \frac{(2n)!}{n!n!} \cdot \frac{1}{n+1} = \frac{1}{n+1}\binom{2n}{n}.$$

Đây chính là số Catalan C_n.

Kết luận Số cách sắp xếp n dấu ngoặc mở và n dấu ngoặc đóng đúng đắn bằng số Catalan thứ n:

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{n!(n+1)!}.$$

Vậy, điều phải chứng minh đã được hoàn tất.

Problem (Pascal triangle & Newton binomial expansion)

Given $m, n \in \mathbb{N}^*$. Write Pascal/Python/C/C++ programs to print the first $n+1$ lines of the Pascal triangle & Newton binomial expansion of

$$(a+b)^n, \quad (a+b+c)^n, \quad \left(\sum_{i=1}^{m} a_i\right)^n, \quad \forall a, b, c, a_i \in \mathbb{R},\ \forall i = 1, \ldots, m.$$

Phương pháp giải:

1. Tam giác Pascal Tam giác Pascal là một mảng số trong đó dòng thứ k (bắt đầu từ $k=0$) chứa các hệ số tổ hợp $\binom{k}{j}$ với $j = 0, 1, \ldots, k$. Mỗi hệ số được tính bằng công thức:

$$\binom{k}{j} = \frac{k!}{j!(k-j)!}.$$

Để tối ưu, ta sử dụng tính chất:

$$\binom{k}{j} = \binom{k-1}{j-1} + \binom{k-1}{j}, \quad \text{với } \binom{k}{0} = \binom{k}{k} = 1.$$

Tuy nhiên, để đơn giản và tránh tràn số, ta tính trực tiếp:

$$\binom{k}{j} = \prod_{i=1}^{j} \frac{k-i+1}{i}, \quad \text{với } j = \min(j, k-j).$$

Chương trình tạo ma trận $n+1$ dòng, mỗi dòng k chứa $k+1$ hệ số $\binom{k}{j}$, và in ra với định dạng căn giữa dựa trên độ rộng của số lớn nhất.

2. Mở rộng nhị thức $(a+b)^n$ Theo định lý nhị thức Newton:

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k.$$

Mỗi hạng tử có dạng $\binom{n}{k} a^{n-k} b^k$. Chương trình:

- Tính hệ số $\binom{n}{k}$ cho $k = 0$ đến n.
- Tạo chuỗi biểu diễn hạng tử, bỏ qua hệ số 1 và lũy thừa 1 (ví dụ, $1a^2b$ được viết là a^2b).
- Nối các hạng tử bằng dấu $+$.

3. Mở rộng $(a+b+c)^n$ Mở rộng đa thức:

$$(a+b+c)^n = \sum_{i+j+k=n} \binom{n}{i,j,k} a^i b^j c^k,$$

trong đó $\binom{n}{i,j,k} = \frac{n!}{i!j!k!}$ là hệ số đa thức, với $i, j, k \geq 0$ và $i+j+k = n$. Chương trình:

- Duyệt tất cả cặp (i, j) sao cho $i + j \leq n$, đặt $k = n - i - j$.
- Tính hệ số $\binom{n}{i,j,k} = \binom{n}{i} \cdot \binom{n-i}{j}$.
- Tạo chuỗi hạng tử, bỏ qua hệ số và lũy thừa 1.
- Nối các hạng tử bằng dấu $+$.

4. Mở rộng $\left(\sum_{i=1}^{m} a_i\right)^n$ Mở rộng tổng quát:

$$\left(\sum_{i=1}^{m} a_i\right)^n = \sum_{k_1+k_2+\cdots+k_m=n} \binom{n}{k_1, k_2, \ldots, k_m} a_1^{k_1} a_2^{k_2} \ldots a_m^{k_m},$$

với hệ số đa thức:

$$\binom{n}{k_1, k_2, \ldots, k_m} = \frac{n!}{k_1! k_2! \ldots k_m!}.$$

Chương trình:

- Sử dụng đệ quy để sinh tất cả các bộ chỉ số $(k_1, k_2, \ldots, k_m)$ sao cho $k_1 + k_2 + \cdots + k_m = n$.
- Tính hệ số bằng cách nhân các $\binom{n'}{k_i}$ tuần tự, với n' là tổng còn lại.

- Tạo chuỗi hạng tử với các biến a_i, bỏ qua hệ số và lũy thừa 1.
- Nối các hạng tử bằng dấu +.

5. Cài đặt chương trình Sử dụng C++ với các hàm:

- `combination(n, k)`: Tính $\binom{n}{k}$ bằng cách tối ưu hóa với $k = \min(k, n-k)$.
- `pascal_triangle(n)`: Tạo ma trận $n+1$ dòng của tam giác Pascal.
- `print_pascal_triangle`: In tam giác Pascal với định dạng căn giữa.
- `binomial_expansion(n, a, b)`: Tạo chuỗi mở rộng $(a+b)^n$.
- `trinomial_expansion(n, a, b, c)`: Tạo chuỗi mở rộng $(a+b+c)^n$.
- `multinomial_expansion(n, m, terms)`: Tạo chuỗi mở rộng $(\sum_{i=1}^{m} a_i)^n$, với đệ quy để sinh các bộ chỉ số.

Sử dụng kiểu `unsigned long long` để tính hệ số, phù hợp cho n nhỏ (ví dụ, $n \leq 20$). Với n lớn, cần kiểm tra tràn số.

6. Ví dụ kết quả Với $n = 4$, $m = 3$:

- Tam giác Pascal:

$$\begin{array}{ccccc} & & & & 1 \\ & & & 1 & 1 \\ & & 1 & 2 & 1 \\ & 1 & 3 & 3 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

- Mở rộng $(a+b)^4$:

$$a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

- Mở rộng $(a+b+c)^4$:

$$a^4+4a^3b+4a^3c+6a^2b^2+12a^2bc+6a^2c^2+4ab^3+12ab^2c+12abc^2+4ac^3+b^4+4b^3c+6b^2c^2+4bc^3+c^4$$

- Mở rộng $(a_1+a_2+a_3)^4$:

$$a_1^4+4a_1^3a_2+4a_1^3a_3+6a_1^2a_2^2+12a_1^2a_2a_3+6a_1^2a_3^2+4a_1a_2^3+12a_1a_2^2a_3+12a_1a_2a_3^2+4a_1a_3^3+a_2^4+4a_2^3a_3+6a_2^2a_3^2+$$

Week 02

[P 10.1.1.]

A simple graph G has nine edges and the degree of each vertex is at least 3. What are the possibilities for the number of vertices? Give an example, for each possibility.

Bài giải:

Ta có, tổng bậc của tất cả các đỉnh là $2 \times 9 = 18$. Gọi n là số đỉnh, mỗi đỉnh có bậc ít nhất 3, nên tổng bậc ít nhất là $3n$. Do đó, $3n \leq 18$, suy ra $n \leq 6$. Vì bậc của bất kỳ đỉnh nào tối đa là $n-1$, ta có $3 \leq n-1$, nên $n \geq 4$. Vậy các giá trị khả dĩ cho n là $n = 4, 5, 6$.

- **Trường hợp** $n = 4$: Tổng bậc $= 18$, bậc trung bình $= \frac{18}{4} = 4.5$. Vì bậc trung bình không phải số nguyên và tất cả các bậc đều ít nhất là 3, ta kiểm tra xem có dãy bậc nào khả thi không. Dãy khả năng: $4, 4, 5, 5$ (không khả thi vì bậc tối đa là $n - 1 = 3$). Thử $3, 3, 3, 9$: tổng là 18, nhưng bậc 9 vượt quá $n - 1 = 3$. Không có dãy bậc hợp lệ, nên $n = 4$ không khả thi.
- **Trường hợp** $n = 5$: Tổng bậc $= 18$, bậc trung bình $= \frac{18}{5} = 3.6$. Dãy khả thi: $4, 4, 4, 3, 3$, tổng $= 18$. Đây là dãy bậc của $K_5 - e$ (đồ thị đầy đủ 5 đỉnh trừ đi một cạnh). Ví dụ: Các đỉnh $\{v_1, v_2, v_3, v_4, v_5\}$, các cạnh $\{v_1v_2, v_1v_3, v_1v_4, v_1v_5, v_2v_3, v_2v_4, v_2v_5, v_3v_4, v_3v_5\}$.
- **Trường hợp** $n = 6$: Tổng bậc $= 18$, bậc trung bình $= 3$. Dãy khả thi: $3, 3, 3, 3, 3, 3$, tổng $= 18$. Đây là dãy bậc của một đồ thị đều bậc 3, ví dụ, đồ thị Petersen hoặc đồ thị với hai tam giác rời nhau được nối phù hợp. Ví dụ: Các đỉnh $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, các cạnh tạo thành hai tam giác v_1v_2, v_2v_3, v_3v_1 và v_4v_5, v_5v_6, v_6v_4, cộng thêm các cạnh v_1v_4, v_2v_5, v_3v_6.

Vậy, số đỉnh khả dĩ là 5 hoặc 6.

[P 10.1.2.]

Is $7, 7, 6, 5, 5, 4, 4, 4, 3, 2$ a graphic sequence?

Bài giải:

Dãy có 10 số hạng, tổng $7 + 7 + 6 + 5 + 5 + 4 + 4 + 4 + 3 + 2 = 47$, là số lẻ. Theo Bổ đề Bắt tay, tổng bậc của một đồ thị phải là số chẵn (vì nó bằng hai lần số cạnh). Vì 47 là lẻ, dãy này không thể là graphic sequence của bất kỳ đồ thị nào.

Vậy, dãy không phải là graphic sequence.

[P 10.1.3.]

Is there a simple regular graph (*Definition 10.3*) of degree 5 with eight vertices? Why?

Bài giải:

Một đồ thị đều bậc k với n đỉnh có tổng bậc $n \cdot k$. Theo Bổ đề Bắt tay, tổng này bằng $2e$, nên $n \cdot k$ phải chẵn, tức là n hoặc k phải chẵn. Ở đây, $k = 5$ (lẻ) và $n = 8$ (chẵn), nên $8 \cdot 5 = 40$ là chẵn. Số cạnh là $e = \frac{8 \cdot 5}{2} = 20$.

Đối với đồ thị đơn, bậc của mỗi đỉnh tối đa là $n - 1 = 8 - 1 = 7$. Vì $k = 5 \leq 7$, điều này là khả thi. Một đồ thị đều bậc 5 với 8 đỉnh tồn tại, ví dụ, phần bù của một ghép đôi hoàn hảo trong K_8. Xây dựng K_8 và loại bỏ một đồ thị con đều bậc 1 (bốn cạnh rời nhau). Đồ thị còn lại có bậc $7 - 2 = 5$ cho mỗi đỉnh.

Vậy, đồ thị như vậy tồn tại.

[P 10.1.4.]

Is there a simple graph on eight vertices where half of the degrees are 5 and the other half are 3?

Bài giải:

Số đỉnh $n = 8$, với 4 đỉnh bậc 5 và 4 đỉnh bậc 3. Tổng bậc $= 4 \cdot 5 + 4 \cdot 3 = 20 + 12 = 32$, là số chẵn. Số cạnh $= \frac{32}{2} = 16$. Bậc tối đa là 5, nhỏ hơn $n - 1 = 7$, nên khả thi.

Xây dựng đồ thị: Gọi các đỉnh $\{v_1, v_2, v_3, v_4\}$ có bậc 5 và $\{v_5, v_6, v_7, v_8\}$ có bậc 3. Tạo một đồ thị đầy đủ K_4 trên $\{v_1, v_2, v_3, v_4\}$ (6 cạnh, mỗi đỉnh bậc 3). Thêm các cạnh từ mỗi đỉnh v_1, v_2, v_3, v_4 đến hai đỉnh trong $\{v_5, v_6, v_7, v_8\}$, ví dụ, $v_1v_5, v_1v_6, v_2v_5, v_2v_7, v_3v_6, v_3v_8, v_4v_7, v_4v_8$ (8 cạnh). Lúc này, bậc là: $v_1, v_2, v_3, v_4 : 3 + 2 = 5$; $v_5, v_6, v_7, v_8 : 2$. Thêm hai cạnh trong $\{v_5, v_6, v_7, v_8\}$, ví dụ, v_5v_8, v_6v_7. Bậc cuối cùng: $v_5, v_6, v_7, v_8 : 2 + 1 = 3$. Tổng số cạnh $= 6 + 8 + 2 = 16$.

Vậy, đồ thị như vậy tồn tại.

[P 10.1.5.]

The sequence $7, 5, 5, 4, 3, 3, 3, 3, 3, 2$ is graphic because it is the degree sequence of the simple graph of *Figure 10.1*. Apply the Havel–Hakimi algorithm of *Theorem 10.10* to construct a graph with this degree sequence. Is the resulting graph isomorphic – see *Definition 2.29* – to the graph of *Figure 10.1*?

Bài giải:

Áp dụng thuật toán Havel–Hakimi:

- Bắt đầu: $7, 5, 5, 4, 3, 3, 3, 3, 3, 2$.
- Loại đỉnh có bậc cao nhất (7), giảm 7 số tiếp theo đi 1: $4, 4, 3, 2, 2, 2, 2, 3, 2$. Sắp xếp lại: $4, 4, 3, 3, 2, 2, 2, 2$.
- Loại đỉnh bậc 4, giảm 4 số tiếp theo đi 1: $3, 2, 2, 1, 2, 2, 2$. Sắp xếp lại: $3, 2, 2, 2, 2, 1, 2$.
- Loại đỉnh bậc 3, giảm 3 số tiếp theo đi 1: $1, 1, 1, 2, 1, 2$. Sắp xếp lại: $2, 1, 1, 1, 1, 2$.
- Loại đỉnh bậc 2, giảm 2 số tiếp theo đi 1: $0, 0, 1, 1, 2$. Sắp xếp lại: $2, 1, 1, 0, 0$.
- Loại đỉnh bậc 2, giảm 2 số tiếp theo đi 1: $0, 0, 0, 0$. Sắp xếp lại: $0, 0, 0, 0$.

Dãy cuối là dãy bậc của một đồ thị (tất cả 0). Xây dựng ngược lại:

- Dãy $0, 0, 0, 0$: Đồ thị rỗng với 4 đỉnh.
- Thêm đỉnh v_5, bậc 2, nối với v_6, v_7: Dãy $2, 1, 1, 0, 0$.
- Thêm đỉnh v_4, bậc 2, nối với v_6, v_8: Dãy $2, 1, 1, 1, 1, 2$.
- Thêm đỉnh v_3, bậc 3, nối với v_4, v_5, v_6: Dãy $3, 2, 2, 2, 2, 1, 2$.
- Thêm đỉnh v_2, bậc 4, nối với v_3, v_4, v_5, v_6: Dãy $4, 4, 3, 3, 2, 2, 2, 2$.
- Thêm đỉnh v_1, bậc 7, nối với $v_2, v_3, v_4, v_5, v_6, v_7, v_8$: Dãy $7, 5, 5, 4, 3, 3, 3, 3, 3, 2$.

[P 10.1.6.]

Assume that you applied the Havel–Hakimi algorithm to a given sequence, and, at the end of the process, you arrived at a graphic sequence. You draw a simple graph corresponding to this final sequence, and work your way back to construct a simple graph with the original sequence as its degree sequence. As you work your way back, is it the case that, at every step, after adding a new vertex, you add edges between this new vertex and those existing vertices that have the highest degrees? Either prove that you do or provide an example where you don't.

Bài giải:

Xét dãy $3, 3, 2, 2$. Bước đầu tiên: loại đỉnh bậc 3, giảm 3 số tiếp theo: $2, 1, 1$, sắp xếp lại: $2, 1, 1$. Loại đỉnh bậc 2: $0, 0$. Dãy $0, 0$ là dãy bậc. Xây dựng ngược:

- Dãy $0, 0$: Hai đỉnh v_3, v_4, không có cạnh.
- Thêm đỉnh v_2, bậc 2, nối với v_3, v_4: Dãy $2, 1, 1$.
- Thêm đỉnh v_1, bậc 3. Các đỉnh hiện có: v_2 (bậc 2), v_3, v_4 (bậc 1). Nếu luôn chọn đỉnh bậc cao nhất, nối v_1 với v_2 và hai đỉnh khác, nhưng có thể nối v_1 với v_3, v_4 và một đỉnh khác, ví dụ, tạo vòng $v_1 - v_3 - v_2 - v_4 - v_1$.

Trong trường hợp này, ta có thể không chọn đỉnh bậc cao nhất mà vẫn thu được đồ thị hợp lệ.

Vậy, không phải lúc nào cũng thêm cạnh tới các đỉnh bậc cao nhất.

[P 10.1.7.]

Assume that you have a sequence d_1, d_2, d_3, d_4 (with $d_1 \geq d_2 \geq d_3 \geq d_4 \geq 0$) that you know is *not* graphic. You consider the sequence $d_1, d_2, d_3, d_4 + 1, 1$. Can this sequence be graphic (after possible rearranging so that it is in non-increasing order)? Either prove that it is never graphic or give an example where it becomes graphic.

Bài giải:

Dãy $3, 3, 3, 1$ không phải dãy bậc (tổng 10, chẵn, nhưng áp dụng Havel–Hakimi: loại 3, giảm 3 số tiếp theo: $2, 2, 0$, không thể tạo đồ thị với một đỉnh bậc 0 và hai đỉnh bậc 2).

Xét dãy mới $3, 3, 3, 2, 1$, sắp xếp: $3, 3, 3, 2, 1$. Tổng $= 12$, chẵn. Áp dụng Havel–Hakimi:

- Loại 3: $2, 2, 1, 0$.
- Loại 2: $1, 0, 0$.
- Loại 1: $0, 0$.

Dãy cuối là dãy bậc. Vậy, dãy mới có thể là dãy bậc.

[P 10.1.8.]

A sequence is graphic if it is the degree sequence of a *simple* graph. Is there a sequence that is not graphic, and yet is the degree sequence of a *multigraph*? Either prove that there are no such sequences or give a specific example.

Bài giải:

Có, ví dụ dãy $3, 3, 3, 3$. Tổng $= 12$, chẵn. Đối với đồ thị đơn trên 4 đỉnh, bậc tối đa là 3. Thử Havel–Hakimi:

- Loại 3: $2, 2, 2$.
- Loại 2: $1, 1$, không thể tạo đồ thị với hai đỉnh bậc 1 (tổng lẻ).

Dãy không phải dãy bậc của đồ thị đơn. Nhưng trong đa đồ thị, có thể có 4 đỉnh, mỗi đỉnh bậc 3, ví dụ, một đồ thị với các cạnh kép.

Vậy, tồn tại dãy như vậy.

[P 10.1.9.]

Can you find a sequence that is *not* the degree sequence of a multigraph, but is the degree sequence of a general graph?

Bài giải:

Có, ví dụ dãy 1. Trong đa đồ thị, tổng bậc phải chẵn, nhưng 1 là lẻ, nên không thể. Trong đồ thị tổng quát, một đỉnh với một vòng (loop) có bậc 1 (vì vòng góp 2 vào bậc).

Vậy, tồn tại dãy như vậy.

[P 10.1.10.]

Let $d_1, d_2, \ldots, d_n$ be a non-increasing sequence of n non-negative integers. By *Theorem 10.9*, if this sequence is the degree sequence of a general graph, then the sum $d_1 + d_2 + \cdots + d_n$ is even. What about the converse?

Bài giải:

Tổng chẵn là điều kiện cần (theo Định lý 10.9). Nhưng không đủ. Ví dụ: dãy 3 (một đỉnh). Tổng $= 3$, lẻ, không thể là dãy bậc. Với tổng chẵn, ví dụ $3, 1$, tổng $= 4$, nhưng không thể tạo đồ thị tổng quát vì một đỉnh bậc 3 cần ít nhất hai đỉnh khác (hoặc vòng và cạnh kép, nhưng tổng vẫn lẻ).

Vậy, điều ngược lại không đúng.

[P 10.1.11.]

Find two non-isomorphic regular simple graphs of degree 3 and order 6.

Bài giải:

Đồ thị đều bậc 3 với 6 đỉnh có tổng bậc $6 \cdot 3 = 18$, số cạnh $\frac{18}{2} = 9$. Hai đồ thị không đồng cấu:

- **Đồ thị 1**: $K_{3,3}$, đồ thị hai phía đầy đủ với hai tập 3 đỉnh. Mỗi đỉnh nối với 3 đỉnh ở tập kia.

- **Đồ thị 2**: Đồ thị với một chu trình 6 đỉnh và các dây bổ sung, ví dụ, đỉnh $v_1, v_2, \ldots, v_6$ tạo chu trình $v_1 - v_2 - v_3 - v_4 - v_5 - v_6 - v_1$, thêm dây v_1v_4, v_2v_5, v_3v_6. Mỗi đỉnh có bậc 3.

Không đồng cấu vì $K_{3,3}$ là hai phía, không chứa chu trình lẻ, trong khi đồ thị thứ hai có chu trình 3 đỉnh.

P 10.1.12

Apply the Havel–Hakimi algorithm of Theorem 10.10 to the sequence 5, 4, 4, 2, 2, 1. Is the sequence graphic? Can you use the Havel–Hakimi algorithm to find a general graph with this degree sequence that has exactly one loop?

Bài giải:

Tổng $= 5 + 4 + 4 + 2 + 2 + 1 = 18$, chẵn. Áp dụng Havel–Hakimi:

- Loại 5: $3, 3, 1, 1, 0$.
- Loại 3: $2, 0, 0, 0$.
- Loại 2: $0, 0, 0$.

Dãy là dãy bậc. Để có một vòng, trong đồ thị tổng quát, một vòng tại đỉnh góp 2 vào bậc. Thử xây dựng với một vòng tại đỉnh cuối (bậc 1 trong đồ thị đơn, thêm vòng để bậc 2). Tuy nhiên, dãy yêu cầu bậc 1, không thể có vòng. Do đó, không thể có đúng một vòng.

Vậy, dãy là dãy bậc, nhưng không thể có đồ thị tổng quát với đúng một vòng.

P 10.1.13

(a) Prove that a sequence $d_1, d_2, \ldots, d_p$ is a graphic sequence if and only if the sequence $p - d_p - 1, \ldots, p - d_2 - 1, p - d_1 - 1$ is graphic.
(b) Is 9,9,9,9,9,9,9,8,8,8 a graphic sequence?

Bài giải:

(a) Đây là định lý về đồ thị phần bù. Nếu G có dãy bậc $d_1, \ldots, d_p$, thì phần bù $\overline{G}$ có bậc của đỉnh v_i là $p - 1 - d_i$. Dãy mới là dãy bậc của $\overline{G}$, và ngược lại. (b) Tổng $= 7 \cdot 9 + 3 \cdot 8 = 63 + 24 = 87$, lẻ, nên không phải dãy bậc.

P 10.1.14

I have a simple graph with six vertices. The degrees of five of the vertices are 5, 4, 4, 2, and 2. What are the possibilities for the degree of the sixth vertex?

Bài giải:

Tổng bậc của 5 đỉnh $= 5 + 4 + 4 + 2 + 2 = 17$, lẻ. Tổng bậc cả đồ thị phải chẵn, nên bậc của đỉnh thứ 6 phải lẻ. Bậc tối đa $= 5$. Các TH có thể xảy ra là: 1, 3, 5.

- Bậc 1: Tổng $= 18$, khả thi.
- Bậc 3: Tổng $= 20$, khả thi.
- Bậc 5: Tổng $= 22$, khả thi.

Vậy, các giá trị bậc của đỉnh thứ 6 là 1, 3, 5.

P 10.1.15

Can we have a simple graph where the degree sequence consists of all distinct integers? What about a multigraph?

Bài giải:

- **Đồ thị đơn**: Không, vì nếu các bậc khác nhau, chúng là $0, 1, \ldots, n-1$, nhưng đỉnh bậc $n-1$ phải nối với tất cả, không thể có đỉnh bậc 0.
- **Đa đồ thị**: Có, ví dụ $1, 2, 3$ với cạnh kép và cạnh đơn.

Bài toán 35

Viết chương trình C/C++, Python sử dụng định lý Euler và thuật toán Havel–Hakimi để kiểm tra 1 dãy số nguyên không âm được nhập có phải là 1 graphical sequence hay không.

Input. Dòng 1 chứa số bộ test $t \in \mathbb{N}^*$. Tiếp theo, mỗi bộ test gồm 2 dòng: Dòng 1 chứa $n := |V| \in \mathbb{N}^*$: số đỉnh của đồ thị $G = (V, E)$. Dòng 2 chứa dãy $d_1, \ldots, d_n \in \mathbb{N}$.

Output. Xuất ra 1 nếu dãy đó là dãy graphical sequence, 0 nếu dãy đó không phải là dãy graphical sequence, nếu chưa quyết định được thì xuất ra dãy không thể giảm được nữa thu được từ thuật toán Havel–Hakimi **??**.

graphical_sequence.inp	graphical_sequence.out
3 4 3 2 1 1 4 2 2 1 1 10 7 7 6 6 6 5 4 3 1 9 7 7 6 5 4 4 4 3 2 10 7 5 5 4 3 3 3 3 2 2	

Bài giải:

Để kiểm tra một dãy số nguyên không âm $d_1, d_2, \ldots, d_n$ có phải là dãy graphical sequence (dãy bậc của một đồ thị đơn) hay không, ta sử dụng định lý Euler và thuật toán Havel-Hakimi. Dưới đây là mô tả chi tiết phương pháp.

Định lý Euler Một dãy số nguyên không âm $d_1, d_2, \ldots, d_n$ là graphical sequence nếu thỏa mãn các điều kiện sau:

- Không có số âm: $d_i \geq 0, \forall i = 1, 2, \ldots, n$.
- Tổng các bậc phải chẵn: $\sum_{i=1}^{n} d_i$ là số chẵn (do mỗi cạnh trong đồ thị đóng góp vào tổng bậc của hai đỉnh).

Thuật toán Havel-Hakimi Thuật toán Havel-Hakimi cung cấp một cách kiểm tra đệ quy xem dãy có phải là graphical sequence hay không. Các bước thực hiện như sau:

Algorithm 1 Thuật toán Havel-Hakimi

Input: Dãy số nguyên không âm $d_1, d_2, \ldots, d_n$.
Output: 1 nếu dãy là graphical sequence, 0 nếu không phải, hoặc dãy không thể giảm được nữa.
function IsGraphicalSequence(d, n)
    if có $d_i < 0$ cho một số i **then**
        return 0 ▷ Dãy không hợp lệ do có số âm
    end if
    if $\sum_{i=1}^{n} d_i$ là số lẻ **then**
        return 0 ▷ Tổng bậc không chẵn
    end if
    while dãy không rỗng và $d_1 > 0$ **do**
        Sắp xếp dãy $d_1, d_2, \ldots, d_n$ theo thứ tự giảm dần.
        Lấy d_1, đặt $d = d_1$, và xóa d_1 khỏi dãy.
        if $d > |d|$ (số lượng phần tử còn lại) **then**
            return 0 ▷ Không đủ phần tử để giảm
        end if
        for $i = 1$ đến d **do**
            Giảm d_i đi 1.
        end for
        if có $d_i < 0$ cho một số i **then**
            return 0 ▷ Dãy không hợp lệ do có số âm
        end if
        Xóa các số 0 ở cuối dãy (nếu có).
    end while
    if dãy rỗng hoặc toàn số 0 **then**
        return 1 ▷ Dãy là graphical sequence
    else
        return dãy hiện tại ▷ Dãy không thể giảm được nữa
    end if
end function

Giải thích các bước

- **Kiểm tra Euler**: Đầu tiên, kiểm tra xem dãy có số âm nào không và tổng các số trong dãy có chẵn không. Nếu không thỏa, dãy không phải graphical sequence.
- **Áp dụng Havel-Hakimi**:
  - Sắp xếp dãy giảm dần để lấy số lớn nhất d_1.
  - Giảm d_1 về 0 và giảm d_1 số tiếp theo đi 1.
  - Nếu không đủ số để giảm hoặc xuất hiện số âm, dãy không phải graphical sequence.
  - Lặp lại quá trình cho dãy mới (sau khi bỏ d_1) cho đến khi dãy toàn 0 (là graphical sequence) hoặc không thể giảm tiếp (trả về dãy hiện tại).
- **Kết quả**:
  - Nếu dãy toàn 0, trả về 1.
  - Nếu có số âm hoặc không đủ số để giảm, trả về 0.
  - Nếu dãy không thể giảm tiếp mà vẫn có số khác 0, trả về dãy đó.

Bài toán 11 (Euler candy problem – Bài toán chia kẹo Euler)

Cho $m, n \in \mathbb{N}$. Xét phương trình nghiệm nguyên

$$\sum_{i=1}^{n} x_i = m,$$

(a) Đếm số nghiệm nguyên dương của phương trình. (b) Đếm số nghiệm nguyên không âm của phương trình. (c) Đếm số nghiệm nguyên của phương trình

$$\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i = s, \quad \{x_i \geq 1, \forall i \in [m]\},$$

$$\{y_i \geq 0, \forall i \in [n]\},$$

(d) Đếm số nghiệm nguyên của phương trình

$$\sum_{i=1}^{n} x_i = m \cdot s,\ x_i \geq m_i,\ \forall i \in [n],$$

(e) Đếm số nghiệm nguyên của phương trình

$$\sum_{i=1}^{n} x_i = m \cdot s,\ x_i \leq M_i,\ \forall i \in [n],$$

Bài giải:

Phần (a): Đếm số nghiệm nguyên dương của $\sum_{i=1}^{n} x_i = m$ Tìm số nghiệm nguyên dương của phương trình:

$$x_1 + x_2 + \cdots + x_n = m, \quad x_i \geq 1, \forall i \in [n].$$

Đây là bài toán tìm số cách phân phối m đơn vị (kẹo) vào n hộp, mỗi hộp nhận ít nhất 1 đơn vị. Đặt $y_i = x_i - 1$, ta có $y_i \geq 0$ và phương trình trở thành:

$$(y_1 + 1) + (y_2 + 1) + \cdots + (y_n + 1) = m \implies y_1 + y_2 + \cdots + y_n = m - n.$$

Số nghiệm nguyên không âm của phương trình $y_1 + y_2 + \cdots + y_n = m - n$ là số tổ hợp chập $m - n$ của n với lặp, được tính bởi:

$$\binom{(m-n)+n-1}{n-1} = \binom{m-1}{n-1}.$$

Vậy, số nghiệm nguyên dương là:

$$\boxed{\binom{m-1}{n-1}}.$$

Điều kiện: $m \geq n$, vì nếu $m < n$, không tồn tại nghiệm nguyên dương (do mỗi $x_i \geq 1$).

Phần (b): Đếm số nghiệm nguyên không âm của $\sum_{i=1}^{n} x_i = m$ Tìm số nghiệm nguyên không âm của phương trình:

$$x_1 + x_2 + \cdots + x_n = m, \quad x_i \geq 0, \forall i \in [n].$$

Đây là bài toán phân phối m đơn vị vào n hộp, mỗi hộp có thể nhận 0 hoặc nhiều hơn. Số nghiệm nguyên không âm được tính bằng số tổ hợp chập m của n với lặp:

$$\binom{m+n-1}{n-1}.$$

Vậy, số nghiệm nguyên không âm là:

$$\boxed{\binom{m+n-1}{n-1}}.$$

Phần (c): Đếm số nghiệm nguyên của $\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i = s$ Tìm số nghiệm nguyên của phương trình:

$$x_1 + x_2 + \cdots + x_m + y_1 + y_2 + \cdots + y_n = s, \quad x_i \geq 1, \forall i \in [m], \quad y_i \geq 0, \forall i \in [n].$$

Đặt $x_i' = x_i - 1$, ta có $x_i' \geq 0$ và phương trình trở thành:

$$(x_1' + 1) + \cdots + (x_m' + 1) + y_1 + \cdots + y_n = s \implies x_1' + \cdots + x_m' + y_1 + \cdots + y_n = s - m.$$

Đây là bài toán tìm số nghiệm nguyên không âm của phương trình với $m + n$ biến $(x_1', \ldots, x_m', y_1, \ldots, y_n)$ có tổng bằng $s - m$. Số nghiệm là:

$$\binom{(s-m)+(m+n)-1}{(m+n)-1} = \binom{s+n-1}{m+n-1}.$$

Vậy, số nghiệm là:

$$\boxed{\binom{s+n-1}{m+n-1}}.$$

Điều kiện: $s \geq m$, vì mỗi $x_i \geq 1$.

Phần (d): Đếm số nghiệm nguyên của $\sum_{i=1}^{n} x_i = m \cdot s$, $x_i \geq m_i$ Tìm số nghiệm nguyên của phương trình:

$$x_1 + x_2 + \cdots + x_n = m \cdot s, \quad x_i \geq m_i, \forall i \in [n].$$

Đặt $x_i' = x_i - m_i$, ta có $x_i' \geq 0$ và phương trình trở thành:

$$(x_1' + m_1) + (x_2' + m_2) + \cdots + (x_n' + m_n) = m \cdot s \implies x_1' + x_2' + \cdots + x_n' = m \cdot s - \sum_{i=1}^{n} m_i.$$

Số nghiệm nguyên không âm của phương trình $x_1' + x_2' + \cdots + x_n' = m \cdot s - \sum_{i=1}^{n} m_i$ là:

$$\binom{(m \cdot s - \sum_{i=1}^{n} m_i) + n - 1}{n-1}.$$

Vậy, số nghiệm là:

$$\boxed{\binom{m \cdot s - \sum_{i=1}^{n} m_i + n - 1}{n-1}}.$$

Điều kiện: $m \cdot s \geq \sum_{i=1}^{n} m_i$, vì tổng các $x_i' \geq 0$.

Phần (e): Đếm số nghiệm nguyên của $\sum_{i=1}^{n} x_i = m \cdot s$, $x_i \leq M_i$ Tìm số nghiệm nguyên của phương trình:

$$x_1 + x_2 + \cdots + x_n = m \cdot s, \quad x_i \leq M_i, \forall i \in [n].$$

Sử dụng nguyên lý bổ sung (inclusion-exclusion principle) để đếm số nghiệm nguyên không âm thỏa mãn $x_i \leq M_i$. Đầu tiên, số nghiệm nguyên không âm của phương trình $x_1 + x_2 + \cdots + x_n = m \cdot s$ là:

$$\binom{m \cdot s + n - 1}{n-1}.$$

Gọi A_i là tập hợp các nghiệm có $x_i \geq M_i + 1$. Số nghiệm thỏa mãn ít nhất một điều kiện $x_i \geq M_i + 1$ được tính bằng:

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

Với mỗi tập $S \subseteq \{1, 2, \ldots, n\}$, số nghiệm có $x_i \geq M_i + 1$ cho mọi $i \in S$ được tính bằng cách đặt $x_i' = x_i - (M_i + 1)$, dẫn đến:

$$\sum_{i \in S} x_i' + \sum_{i \notin S} x_i = m \cdot s - \sum_{i \in S} (M_i + 1).$$

Số nghiệm không âm của phương trình này là:

$$\binom{m \cdot s - \sum_{i \in S} (M_i + 1) + n - 1}{n-1}.$$

Do đó, số nghiệm thỏa mãn tất cả $x_i \leq M_i$ là:

$$\binom{m \cdot s + n - 1}{n - 1} - \sum_{S \subseteq \{1,\ldots,n\}, S \neq \emptyset} (-1)^{|S|+1} \binom{m \cdot s - \sum_{i \in S}(M_i + 1) + n - 1}{n - 1}.$$

Vậy, số nghiệm là:

$$\boxed{\sum_{S \subseteq \{1,\ldots,n\}} (-1)^{|S|} \binom{m \cdot s - \sum_{i \in S}(M_i + 1) + n - 1}{n - 1}}.$$

Điều kiện: $m \cdot s \geq 0$, và số tổ hợp được lấy bằng 0 nếu số âm hoặc không xác định.

Week 03

Chứng minh: Mệnh đề 1

Cho hàm sinh $G(x) = (1 + x + x^2 + \cdots)^n$.

(a) Đặt a_r là hệ số của x^r trong khai triển của $G(x)$ thì $a_r = C^r_{r+n-1}$.

(b) $(1 - x^m)^n = 1 - C^1_n x^m + C^2_n x^{2m} - \cdots + (-1)^n x^{nm}$.

Chứng minh:

Mệnh đề 2 (Công thức xác định hệ số tích của hai hàm sinh). Cho hai hàm sinh $A(x)$ và $B(x)$ lần lượt là

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots,$$

$$B(x) = b_0 + b_1 x + b_2 x^2 + \cdots.$$

Đặt

$$G(x) = A(x) \cdot B(x) = (a_0 + a_1 x + a_2 x^2 + \cdots)(b_0 + b_1 x + b_2 x^2 + \cdots)$$

$$= a_0 b_0 + (a_0 b_1 + a_1 b_0)x + (a_0 b_2 + a_1 b_1 + a_2 b_0)x^2 + (a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0)x^3 + \cdots.$$

Khỏi đầu số của x^r trong khai triển của $G(x)$ là

$$a_0 b_r + a_1 b_{r-1} + a_2 b_{r-2} + \cdots + a_{r-1} b_1 + a_r b_0 \quad (*)$$

Bài giải:

Chứng minh Mệnh đề 1

Cho hàm sinh $G(x) = (1 + x + x^2 + \cdots)^n$.

(a) Đặt a_r là hệ số của x^r trong khai triển của $G(x)$, chứng minh rằng $a_r = C^r_{r+n-1}$.

Lời giải:

Ta có $1 + x + x^2 + \cdots = \frac{1}{1-x}$ (với $|x| < 1$), do đó:

$$G(x) = \left(\frac{1}{1-x}\right)^n = (1-x)^{-n}.$$

Theo công thức khai triển nhị thức Newton cho lũy thừa âm:

$$(1-x)^{-n} = \sum_{r=0}^{\infty} \binom{-n}{r} (-x)^r = \sum_{r=0}^{\infty} \binom{n+r-1}{r} x^r,$$

trong đó hệ số của x^r là:

$$\binom{n+r-1}{r} = C^r_{n+r-1}.$$

Vậy, hệ số a_r của x^r trong khai triển của $G(x)$ là:

$$a_r = C^r_{n+r-1}.$$

Kết luận: $a_r = C^r_{r+n-1}$.

(b) Chứng minh rằng:

$$(1-x^m)^n = 1 - C^1_n x^m + C^2_n x^{2m} - \cdots + (-1)^n x^{nm}.$$

Lời giải:

Áp dụng công thức nhị thức Newton cho $(1-x^m)^n$:

$$(1-x^m)^n = \sum_{k=0}^{n} \binom{n}{k} (1)^{n-k} (-x^m)^k = \sum_{k=0}^{n} \binom{n}{k} (-1)^k x^{mk}.$$

Do đó, ta có:

$$(1-x^m)^n = \sum_{k=0}^{n} C^k_n (-1)^k x^{mk} = 1 - C^1_n x^m + C^2_n x^{2m} - \cdots + (-1)^n C^n_n x^{nm}.$$

Kết luận: Công thức được chứng minh.

Chứng minh Mệnh đề 2

Cho hai hàm sinh:

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots,$$
$$B(x) = b_0 + b_1 x + b_2 x^2 + \cdots.$$

Đặt $G(x) = A(x) \cdot B(x)$. Chứng minh rằng hệ số của x^r trong khai triển của $G(x)$ là:

$$a_0b_r + a_1b_{r-1} + a_2b_{r-2} + \cdots + a_{r-1}b_1 + a_rb_0.$$

Lời giải:
Ta có:

$$G(x) = A(x) \cdot B(x) = \left(\sum_{i=0}^{\infty} a_i x^i\right)\left(\sum_{j=0}^{\infty} b_j x^j\right).$$

Hệ số của x^r trong tích $G(x)$ là tổng các tích a_ib_j sao cho $i + j = r$. Do đó:

$$\text{Hệ số của } x^r = \sum_{i=0}^{r} a_i b_{r-i}.$$

Viết rõ hơn:

$$\sum_{i=0}^{r} a_i b_{r-i} = a_0b_r + a_1b_{r-1} + a_2b_{r-2} + \cdots + a_{r-1}b_1 + a_rb_0.$$

Điều này khớp với biểu thức đã cho. Vậy, hệ số của x^r trong khai triển của $G(x)$ là:

$$a_0b_r + a_1b_{r-1} + a_2b_{r-2} + \cdots + a_{r-1}b_1 + a_rb_0.$$

Kết luận: Công thức được chứng minh.

Week 04

Chứng minh Dijkstra algorithm

Bài giải:

Mô tả thuật toán Dijkstra

Cho đồ thị $G = (V, E)$ với tập đỉnh V và tập cạnh E, mỗi cạnh $(u, v) \in E$ có trọng số không âm $w(u, v) \geq 0$. Thuật toán Dijkstra bắt đầu từ đỉnh nguồn $s \in V$ và thực hiện các bước sau:

1. Khởi tạo mảng khoảng cách $d[v] = \infty$ cho mọi $v \in V$, trừ $d[s] = 0$.
2. Khởi tạo tập các đỉnh đã thăm $S = \emptyset$ và hàng đợi ưu tiên Q chứa tất cả các đỉnh, với ưu tiên dựa trên giá trị $d[v]$.
3. Trong khi Q không rỗng:
   - Lấy đỉnh $u \in Q$ có $d[u]$ nhỏ nhất và xóa u khỏi Q.
   - Thêm u vào S.
   - Với mỗi đỉnh kề v của u mà $v \notin S$, nếu $d[u] + w(u, v) < d[v]$, cập nhật $d[v] = d[u] + w(u, v)$ và cập nhật hàng đợi ưu tiên.
4. Trả về mảng khoảng cách d.

Để chứng minh thuật toán Dijkstra đúng, ta cần chứng minh rằng khi một đỉnh u được lấy ra từ hàng đợi ưu tiên và thêm vào tập S, giá trị $d[u]$ là khoảng cách ngắn nhất từ đỉnh nguồn s đến u. Ta sử dụng quy nạp trên số đỉnh trong tập S.

Bước cơ sở

Khi $S = \emptyset$, đỉnh đầu tiên được chọn là đỉnh nguồn s. Theo thuật toán, $d[s] = 0$, và không có đường đi nào từ s đến s có độ dài nhỏ hơn 0. Do đó, $d[s]$ là khoảng cách ngắn nhất, và bước cơ sở được thỏa mãn.

Bước quy nạp

Giả sử rằng với mọi đỉnh v đã được thêm vào S, ta có $d[v]$ là khoảng cách ngắn nhất từ s đến v. Gọi u là đỉnh tiếp theo được chọn từ hàng đợi ưu tiên, tức là u có $d[u]$ nhỏ nhất trong số các đỉnh chưa thuộc S.

Ta cần chứng minh rằng $d[u]$ là khoảng cách ngắn nhất từ s đến u. Giả sử ngược lại rằng tồn tại một đường đi ngắn hơn từ s đến u với độ dài $\delta(s, u) < d[u]$. Gọi đường đi ngắn nhất này là $P = s \to v_1 \to v_2 \to \cdots \to v_k \to u$, trong đó v_k là đỉnh cuối cùng trên đường đi P trước u mà chưa thuộc S (có thể $v_k = u$).

Vì $v_k \notin S$, ta có $d[v_k] \leq \delta(s, v_k)$, bởi $d[v_k]$ là khoảng cách nhỏ nhất từ s đến v_k tại thời điểm hiện tại (theo giả thiết quy nạp hoặc do v_k vẫn còn trong hàng đợi). Vì $\delta(s, u)$ là độ dài đường đi ngắn nhất từ s đến u, ta có:

$$\delta(s, u) = \delta(s, v_k) + w(v_k, u).$$

Do trọng số $w(v_k, u) \geq 0$, suy ra:

$$\delta(s, v_k) \leq \delta(s, u) < d[u].$$

Vì vậy, $d[v_k] \leq \delta(s, v_k) < d[u]$. Điều này mâu thuẫn với việc u được chọn từ hàng đợi ưu tiên, bởi $v_k \notin S$ và $d[v_k] < d[u]$, nghĩa là v_k đáng lẽ phải được chọn trước u.

Do đó, không thể tồn tại đường đi nào từ s đến u ngắn hơn $d[u]$. Vậy, tại thời điểm u được thêm vào S, $d[u]$ là khoảng cách ngắn nhất từ s đến u.

Kết luận

Qua quy nạp, khi thuật toán kết thúc, với mỗi đỉnh $v \in V$, giá trị $d[v]$ là khoảng cách ngắn nhất từ s đến v. Thuật toán Dijkstra đảm bảo tính đúng đắn khi tất cả trọng số cạnh là không âm.

Week 05

P 10.2.1

Let x and y be vertices of a general graph.

(a) Suppose that there is a closed walk containing both x and y. Must there be a closed trail containing both x and y?

(b) Suppose that there is a closed trail containing both x and y. Must there be a cycle containing both x and y?

Bài giải

(a) Không, không nhất thiết tồn tại một đường dẫn khép kín chứa cả x và y. Một đường đi khép kín cho phép lặp lại cả đỉnh và cạnh, trong khi một đường dẫn khép

kín cho phép lặp lại đỉnh nhưng không lặp lại cạnh. Xét một đồ thị với các đỉnh $\{x, y, z\}$ và các cạnh $\{xy, yz, yz\}$ (một đa đồ thị với hai cạnh giữa y và z). Đường đi x, y, z, y, x là một đường đi khép kín chứa cả x và y, sử dụng cạnh yz hai lần. Tuy nhiên, bất kỳ đường dẫn khép kín nào cũng phải sử dụng mỗi cạnh không quá một lần. Vì đồ thị chỉ có ba cạnh, bất kỳ đường dẫn khép kín nào (như một mạch Euler) phải có bậc chẵn tại mỗi đỉnh. Ở đây, $\deg(y) = 3$, là số lẻ, nên không tồn tại mạch Euler, và do đó không có đường dẫn khép kín nào chứa cả x và y mà không lặp lại cạnh.

(b) Không, không nhất thiết tồn tại một chu trình chứa cả x và y. Một đường dẫn khép kín là một đường đi không lặp lại cạnh, nhưng có thể lặp lại đỉnh. Một chu trình là một đường dẫn khép kín không lặp lại đỉnh (trừ đỉnh đầu và cuối). Xét một đồ thị với các đỉnh $\{x, y, z\}$ và các cạnh $\{xy, yz, zx\}$, tạo thành một tam giác. Đường dẫn khép kín x, y, z, x chứa cả x và y, nhưng nó lặp lại đỉnh z, nên không phải là một chu trình. Để có một chu trình chứa cả x và y, cần một đường đi đơn giản khép kín đi qua cả hai đỉnh, nhưng trong đồ thị này, chu trình x, y, z, x không phải là chu trình chỉ chứa x và y mà không có đỉnh khác. Do đó, không nhất thiết tồn tại chu trình chỉ chứa x và y.

P 10.2.2

If a vertex x of a graph is on a circuit, then is x also on a cycle?

Bài giải:

Có, nếu một đỉnh x nằm trên một mạch, thì x cũng nằm trên một chu trình. Một mạch là một đường dẫn khép kín (không lặp lại cạnh), và một chu trình là một đường dẫn khép kín không lặp lại đỉnh (trừ đỉnh đầu và cuối). Giả sử x nằm trên một mạch $v_1, v_2, \ldots, v_k, v_1$, với $v_i = x$ tại một vị trí nào đó. Nếu mạch này không lặp lại đỉnh (trừ v_1), thì nó đã là một chu trình chứa x. Nếu có đỉnh lặp lại, ta có thể rút ngắn mạch bằng cách loại bỏ các đoạn giữa hai lần xuất hiện của cùng một đỉnh, lặp lại quá trình này cho đến khi thu được một chu trình (không lặp lại đỉnh trừ đầu và cuối). Chu trình này vẫn chứa x, vì x có trong mạch ban đầu. Do đó, x luôn nằm trên một chu trình.

P 10.2.3

Let $G = (V, E)$ be a general graph and let $x, y \in V$ with $x \neq y$. Assume that x and y are joined by a walk. In Lemma 10.16, we proved that x and y must be joined by a path. A friend objects to the proof given in the text. In that proof it was decreed: "Eliminate any portion between two occurrences of the same vertex (and one of the two offending vertices)." Your friend claims that this is an ambiguous proclamation since there may be multiple vertices occurring multiple times. It is not clear where to start and how to stop. Put your friend at ease by rewriting the proof using induction on the number of edges on the walk.

Bài giải:

Để chứng minh rằng nếu có đường đi từ x đến y, thì tồn tại một đường dẫn từ x đến y, ta sử dụng quy nạp trên số cạnh m của đường đi.

Cơ sở quy nạp: Nếu $m = 1$, đường đi chỉ có một cạnh xy. Đây là một đường dẫn đơn giản (không lặp đỉnh hay cạnh), nên kết luận đúng.

Bước quy nạp: Giả sử định lý đúng với mọi đường đi có ít hơn m cạnh ($m \geq 2$). Xét một đường đi $W = v_1, v_2, \ldots, v_{m+1}$ từ $x = v_1$ đến $y = v_{m+1}$ với m cạnh. Nếu W không lặp đỉnh (trừ có thể $v_1 = v_{m+1}$), thì W đã là một đường dẫn. Nếu W lặp đỉnh, giả sử $v_i = v_j$ với $1 \leq i < j \leq m + 1$. Ta xét hai trường hợp:

- Nếu $i = 1$ và $j = m + 1$, thì $v_1 = v_{m+1}$, và đường đi $v_1, v_2, \ldots, v_m, v_1$ là một chu trình. Loại bỏ v_{m+1}, ta được đường dẫn $v_1, v_2, \ldots, v_m$ từ x đến v_m.

- Nếu $i \geq 1$ và $j \leq m$, hoặc $j = m + 1$ nhưng $v_1 \neq v_{m+1}$, ta chia đường đi thành hai phần: $W_1 = v_1, \ldots, v_i$ từ x đến v_i, và $W_2 = v_j, \ldots, v_{m+1}$ từ v_i đến y (vì $v_i = v_j$). Cả W_1 và W_2 có ít hơn m cạnh. Theo giả thiết quy nạp, tồn tại đường dẫn P_1 từ x đến v_i và P_2 từ v_i đến y. Nối P_1 và P_2, ta được một đường đi từ x đến y. Nếu đường đi này lặp đỉnh, ta lặp lại quá trình. Vì số cạnh giảm sau mỗi bước, quá trình này kết thúc với một đường dẫn từ x đến y.

Do đó, theo quy nạp, luôn tồn tại một đường dẫn từ x đến y

P 10.2.6

We defined a tree as a connected graph with no cycles (Definition 10.22). Could we have just as well defined a tree as a connected graph with no circuits? What about a connected graph with no closed walks?

Bài giải:

Một cây có thể được định nghĩa là một đồ thị liên thông không có mạch. Một mạch là một đường dẫn khép kín không lặp lại cạnh, và một chu trình là một mạch không lặp lại đỉnh (trừ đầu và cuối). Trong một cây, không có chu trình, và vì mọi mạch đều chứa ít nhất một chu trình (bằng cách rút ngắn các đỉnh lặp lại), một cây cũng không có mạch. Do đó, định nghĩa "đồ thị liên thông không có mạch" tương đương với định nghĩa cây.

Tuy nhiên, một đồ thị liên thông không có đường đi khép kín không tương đương với một cây. Một đường đi khép kín có thể lặp lại cả đỉnh và cạnh. Một đồ thị chỉ có một đỉnh và không có cạnh là liên thông và không có đường đi khép kín, nhưng nó không phải là cây theo định nghĩa (vì cây thường yêu cầu có ít nhất một cạnh nếu có nhiều hơn một đỉnh). Hơn nữa, một đồ thị liên thông với các cạnh lặp (như một đa đồ thị) có thể không có chu trình nhưng vẫn có đường đi khép kín (ví dụ, đi qua một cạnh nhiều lần). Do đó, định nghĩa này không tương đương với cây.

P 10.2.7

I have a simple graph with five vertices and four edges. Could it be a tree? Does it have to be a tree?

Bài giải:

Một đồ thị đơn với 5 đỉnh và 4 cạnh có thể là một cây, nhưng không nhất thiết phải là một cây.

- Một cây với n đỉnh có đúng $n-1$ cạnh và là liên thông. Với $n=5$, một cây có 4 cạnh. Một đồ thị đơn với 5 đỉnh và 4 cạnh có thể liên thông, ví dụ như một đồ thị đường dẫn P_5 (đỉnh v_1, v_2, v_3, v_4, v_5 với các cạnh $v_1v_2, v_2v_3, v_3v_4, v_4v_5$), là một cây.

- Tuy nhiên, đồ thị có thể không liên thông, ví dụ, một đồ thị gồm một chu trình C_3 (3 đỉnh, 3 cạnh) và hai đỉnh cô lập (0 cạnh), tổng cộng 5 đỉnh và $3+0=3$ cạnh, nhưng nếu thêm một cạnh giữa một đỉnh cô lập và một đỉnh trong C_3, ta được 4 cạnh, và đồ thị này không phải là cây vì nó không liên thông.

Do đó, đồ thị có thể là một cây, nhưng không nhất thiết phải là một cây.

P 10.2.8

Let G be a forest consisting of t trees. Assume G has n vertices. How many edges does G have?

Bài giải:

Một rừng là một đồ thị không có chu trình, gồm t thành phần liên thông, mỗi thành phần là một cây. Một cây với k_i đỉnh có $k_i - 1$ cạnh. Giả sử rừng G có t cây, với các cây có số đỉnh lần lượt là $k_1, k_2, \ldots, k_t$, và tổng số đỉnh là $n = k_1 + k_2 + \cdots + k_t$. Số cạnh của rừng là tổng số cạnh của các cây:

$$\text{Số cạnh} = (k_1 - 1) + (k_2 - 1) + \cdots + (k_t - 1) = (k_1 + k_2 + \cdots + k_t) - t = n - t.$$

Vậy, G có $n - t$ cạnh.

P 10.2.9

Let G be a tree. Recall that a vertex v of G of degree 1 is called a leaf. Show that every finite tree with at least two vertices has at least two leaves.

Bài giải:

Xét một cây G với $n \geq 2$ đỉnh. Vì G là một cây, nó liên thông và không có chu trình, nên có đúng $n-1$ cạnh. Tổng các bậc của các đỉnh là $\sum \deg(v) = 2 \cdot (n-1) = 2n-2$. Mỗi lá có bậc 1, và các đỉnh không phải lá có bậc ít nhất 2. Giả sử G có k lá. Tổng bậc của các lá là $k \cdot 1 = k$, và tổng bậc của $n-k$ đỉnh không phải lá là $2n-2-k$. Vì mỗi đỉnh không phải lá có bậc ít nhất 2, ta có:

$$2(n-k) \leq 2n-2-k.$$

Rút gọn: $2n - 2k \leq 2n - 2 - k$, tức là $-2k \leq -2 - k$, hay $k \geq 2$. Do đó, G có ít nhất hai lá.

P 10.2.10

Let G be a tree, and assume that G has one vertex of degree 4. What is the minimum number of leaves possible for G?

Bài giải:

Một cây với n đỉnh có $n-1$ cạnh, và tổng các bậc là $2(n-1) = 2n-2$. Giả sử cây có một đỉnh v bậc 4, k lá (bậc 1), và $m = n-k-1$ đỉnh không phải lá khác v (mỗi đỉnh có bậc ít nhất 2). Tổng các bậc là:

$$\deg(v) + \sum_{\text{lá}} \deg(u) + \sum_{\text{không phải lá}} \deg(w) = 4 + k \cdot 1 + \sum_{\text{không phải lá}} \deg(w) = 2n-2.$$

Tổng bậc của m đỉnh không phải lá là $\sum \deg(w) \geq 2m = 2(n-k-1) = 2n-2k-2$. Do đó:

$$4 + k + (2n-2k-2) \leq 2n-2.$$

Rút gọn: $2 + k - 2k \leq -2$, tức là $k \geq 4$. Vậy, số lá tối thiểu là 4. Để đạt được số lá tối thiểu, các đỉnh không phải lá (trừ v) phải có bậc 2, và ta có thể xây dựng một cây với một đỉnh bậc 4, bốn lá, và các đỉnh còn lại bậc 2.

P 10.2.11

A mystery simple graph G has the degree sequence $3, 3, 3, 2, 1$. Prove that G has a cycle.

Bài giải:

Đồ thị đơn G có 5 đỉnh với dãy bậc $3, 3, 3, 2, 1$. Tổng các bậc là $3+3+3+2+1 = 12$, nên số cạnh là $12/2 = 6$. Vì G là đồ thị đơn với 5 đỉnh và 6 cạnh, số cạnh bằng $n = 5$, gợi ý rằng G có thể chứa chu trình. Theo Định lý 10.15, một đồ thị đơn với n đỉnh và ít nhất n cạnh phải có một chu trình. Ở đây, $n = 5$, số cạnh là 6, thỏa mãn điều kiện. Hơn nữa, đồ thị có ba đỉnh bậc 3, cho thấy tính liên thông cao. Một cấu hình có thể là một chu trình C_3 (tam giác) với các đỉnh bậc 3, và hai đỉnh khác được nối sao cho tổng bậc là 2 và 1, tạo thành một đồ thị liên thông với ít nhất một chu trình C_3. Do đó, G có một chu trình.

P 10.2.12

Assume that T is a tree with exactly two leaves. Prove that T is a path.

Bài giải:

Một cây T với n đỉnh có $n-1$ cạnh và là liên thông, không có chu trình. Nếu T có đúng hai lá (đỉnh bậc 1), thì các đỉnh còn lại (nếu có) phải có bậc ít nhất 2. Tổng các bậc là $2(n-1) = 2n-2$. Gọi hai lá là u và v, mỗi lá có bậc 1, nên tổng bậc của $n-2$ đỉnh còn lại là $(2n-2) - 2 = 2n-4$. Vì mỗi đỉnh không phải lá có bậc ít nhất 2, số đỉnh không phải lá tối đa là $(2n-4)/2 = n-2$. Do đó, tất cả $n-2$ đỉnh còn lại có bậc đúng 2. Điều này chỉ xảy ra nếu T là một đường dẫn $u = v_1, v_2, \ldots, v_{n-1}, v_n = v$, với v_1 và v_n là lá (bậc 1), và các đỉnh $v_2, \ldots, v_{n-1}$ có bậc 2. Vậy, T là một đường dẫn.

P 10.2.13

Assume that G is a tree with vertices $\{v_1, \ldots, v_n\}$. We know that there is not an edge between v_i and v_j. The graph G is the same as the graph that is present with one extra edge.

Bài giải:

Ta hiểu rằng G là một cây với n đỉnh, và không có cạnh giữa hai đỉnh cụ thể v_i và v_j. Nếu thêm một cạnh giữa v_i và v_j, gọi đồ thị mới là $G' = G + v_iv_j$, thì G' có $n-1+1 = n$ cạnh. Vì G là một cây (liên thông, không chu trình), việc thêm cạnh v_iv_j tạo ra một chu trình duy nhất trong G', vì đường dẫn duy nhất từ v_i đến v_j trong G kết hợp với cạnh v_iv_j tạo thành một chu trình. Do đó, G' là một đồ thị liên thông với đúng một chu trình.

P 10.2.14

Let G be a connected simple graph with 47 vertices and 47 edges. What can you say about the number of cycles in G?

Bài giải:

Một đồ thị đơn liên thông với $n = 47$ đỉnh và $m = 47$ cạnh có số cạnh bằng số đỉnh. Một cây với 47 đỉnh có 46 cạnh, nên G có đúng một cạnh hơn một cây, nghĩa là G là một đồ thị đơn liên thông với đúng một chu trình (được gọi là đồ thị đơn chu trình). Số chu trình trong G là một, vì bất kỳ cạnh bổ sung nào so với cây tạo ra một chu trình duy nhất. Độ dài của chu trình phụ thuộc vào cấu trúc của G, nhưng số lượng chu trình là đúng 1.

P 10.2.15

Let n be an integer greater than or equal to 2. Show that a simple graph (connected or not) with n vertices and at least n edges must have a cycle. What about general graphs?

Bài giải:

Đồ thị đơn: Xét một đồ thị đơn G với $n \geq 2$ đỉnh và ít nhất n cạnh. Nếu G liên thông, ta so sánh với một cây: một cây với n đỉnh có $n-1$ cạnh. Vì G có ít nhất n cạnh, nó có ít nhất một cạnh hơn cây, nên phải chứa ít nhất một chu trình. Nếu G không liên thông, giả sử có k thành phần liên thông, mỗi thành phần có n_i đỉnh ($n_1 + \cdots + n_k = n$). Số cạnh tối đa trong một rừng (không chu trình) là $n_1 - 1 + \cdots + n_k - 1 = n - k$. Vì số cạnh của G là ít nhất n, và $n - k < n$ (vì $k \geq 1$), nên G có nhiều hơn $n - k$ cạnh, do đó ít nhất một thành phần liên thông có số cạnh lớn hơn $n_i - 1$, và thành phần này phải chứa một chu trình.

Đồ thị tổng quát: Trong đồ thị tổng quát (cho phép đa cạnh), kết luận vẫn đúng. Nếu có ít nhất n cạnh, và đồ thị là một rừng (không chu trình), số cạnh tối đa là $n - k$. Với n cạnh, ít nhất một thành phần phải có chu trình, vì số cạnh vượt quá giới hạn của một rừng

P 10.2.17

Let $n \geq 2$, and let $d_1 \geq d_2 \geq \cdots \geq d_n$ be a sequence of positive integers. Prove that this sequence is the degree sequence of a tree if and only if $\sum_{i=1}^{n} d_i = 2n - 2$.

Bài giải:

Chiều thuận: Nếu G là một cây với n đỉnh, thì nó có $n - 1$ cạnh. Tổng các bậc là $\sum \deg(v_i) = 2 \cdot (n - 1) = 2n - 2$. Do đó, dãy bậc $d_1, d_2, \ldots, d_n$ thỏa mãn $\sum d_i = 2n - 2$.

Chiều ngược: Giả sử dãy $d_1 \geq d_2 \geq \cdots \geq d_n$ thỏa mãn $\sum d_i = 2n - 2$, và các d_i là số nguyên dương. Ta chứng minh bằng quy nạp trên n.

Cơ sở: Với $n = 2$, $\sum d_i = 2 \cdot 2 - 2 = 2$. Dãy bậc là $(1, 1)$, tương ứng với đồ thị K_2, là một cây.

Bước quy nạp: Giả sử kết quả đúng với $n - 1$. Xét dãy $d_1 \geq \cdots \geq d_n$ với $\sum d_i = 2n - 2$. Vì $d_n \geq 1$, giả sử $d_n = 1$ (đỉnh có bậc nhỏ nhất thường là lá trong cây). Xóa đỉnh n và cạnh nối với nó (nối với một đỉnh k có bậc $d_k \geq 1$). Đồ thị còn lại có $n - 1$ đỉnh, với dãy bậc mới là $d_1, \ldots, d_k - 1, \ldots, d_{n-1}$, và tổng bậc là:

$$(2n - 2) - d_n - 1 = (2n - 2) - 1 - 1 = 2(n - 1) - 2.$$

Theo giả thiết quy nạp, tồn tại một cây với $n-1$ đỉnh và dãy bậc trên. Thêm lại đỉnh n và cạnh nối với đỉnh k, ta được một đồ thị với dãy bậc gốc, liên thông và không chu trình (vì thêm một lá không tạo chu trình). Do đó, đồ thị này là một cây.

P 10.2.18

Let T be a tree. How many spanning trees can T have?

Bài giải:

Một cây khung của một đồ thị G là một cây con liên thông chứa tất cả các đỉnh của G. Nếu G là một cây T, thì chính T là cây con liên thông duy nhất chứa tất cả các đỉnh, vì bất kỳ đồ thị con nào khác hoặc không liên thông, hoặc có thêm cạnh (tạo chu trình, không phải cây). Do đó, T có đúng một cây khung, chính là T.

P 10.2.19

If T_1 and T_2 are both spanning trees of a graph G, then can T_1 and T_2 have different numbers of edges?

Bài giải:

Không, T_1 và T_2 không thể có số cạnh khác nhau. Một cây khung của G là một cây con liên thông chứa tất cả n đỉnh của G. Một cây với n đỉnh luôn có đúng $n - 1$ cạnh. Do đó, cả T_1 và T_2 đều có $n - 1$ cạnh, và số cạnh của chúng luôn bằng nhau.

P 10.2.21

Let G be a connected general graph. Prove that G has a spanning tree.

Bài giải:

Một cây khung của G là một cây con liên thông chứa tất cả các đỉnh của G. Vì G là liên thông, ta có thể xây dựng một cây khung bằng cách sử dụng thuật toán tìm kiếm, như tìm kiếm theo chiều sâu (DFS) hoặc tìm kiếm theo chiều rộng (BFS), bắt đầu từ một đỉnh bất kỳ. Thuật toán này tạo ra một đồ thị con liên thông không có chu trình (vì chỉ thêm các cạnh cần thiết để kết nối các đỉnh). Kết quả là một cây chứa tất cả các đỉnh của G, tức là một cây khung. Do đó, G luôn có một cây khung.

P 10.2.22

Consider K_4, the complete graph of order 4. (a) How many non-isomorphic spanning trees does K_4 have? (b) [Consider K_4, the complete graph of order 4.]

Bài giải:

(a) K_4 có bao nhiêu cây khung không đồng dạng?

Lời giải: K_4 là đồ thị đơn với 4 đỉnh, mỗi đỉnh có bậc 3, và $\binom{4}{2} = 6$ cạnh. Một cây khung của K_4 là một cây với 4 đỉnh và $4 - 1 = 3$ cạnh. Theo định lý Cayley, số cây khung của K_4 là $4^{4-2} = 4^2 = 16$. Để tìm số cây khung không đồng dạng, ta xét các dạng cây có 4 đỉnh:

- **Cây đường dẫn** P_4: Một đường dẫn v_1, v_2, v_3, v_4. Trong K_4, chọn một đường dẫn 4 đỉnh có $4! = 24$ cách, nhưng vì P_4 đối xứng (đảo ngược thứ tự cho cùng một cây), ta chia đôi, được $24/2 = 12$ cây.

- **Cây sao**: Một đỉnh trung tâm nối với 3 đỉnh khác. Chọn đỉnh trung tâm có 4 cách, và các đỉnh còn lại tự động nối với nó. Có 4 cây sao.

Tổng số cây khung là $12 + 4 = 16$, và hai loại này không đồng dạng (đường dẫn có đường kính 3, sao có đường kính 2). Do đó, có 2 cây khung không đồng dạng.

Week 06

Knapsack Problem

Cho n vật với W_i (weight) và V_i (value), $i \in [n]$.

Câu hỏi

1. Cần xếp các vật vào balo dung lượng W sao cho $\sum v_i$ max
2. Gỉa sử $W \in \mathbb{N}^*, W_i = 1, \forall_i \in [n].V_i$ là một giải tăng ngặt. Giải Knapsack.
3. Gỉa sử $W \in \mathbb{N}^*, W_i = w, \forall_i \subset [n].V_i \in [n], v_i$ tăng. Giải Knapsack.
4. Thiết lập công thức DP $\rightarrow$ code: $V_i \in (0, \infty), \forall v_i \in [n], v_i, W \in \mathbb{N}^*, \forall i \in [n]$.
5. Giả sử $\{w_i\}_{i=1}^n = \{(n - m + 1)1, 2, 3, 4, ...m\} \rightarrow$ Đếm số cách sắp balo với W = $\mathbb{N}^*$

Bài giải:

Câu a: Tối ưu hóa tổng giá trị với giới hạn trọng lượng

Mục tiêu là chọn một tập hợp các vật sao cho tổng trọng lượng $\sum w_i \leq W$ và tổng giá trị $\sum v_i$ đạt tối đa.

Lời giải: Đây là bài toán Knapsack 0-1 cổ điển. Ta có thể sử dụng quy hoạch động (Dynamic Programming - DP) để giải. Đặt $dp[i][j]$ là giá trị tối đa có thể đạt được khi xét i vật đầu tiên và giới hạn trọng lượng là j. Công thức DP:

$$dp[i][j] = \max \begin{cases} dp[i-1][j] & \text{(không chọn vật } i\text{)} \\ dp[i-1][j-w_i] + v_i & \text{(chọn vật } i\text{, nếu } j \geq w_i\text{)} \end{cases}$$

Khởi tạo: $dp[0][j] = 0, \forall j \in [0, W]$.
Kết quả là $dp[n][W]$.

Câu b: $W_i = 1, \forall i \in [n], \{v_i\}$ **tăng ngặt**

Khi $W_i = 1$ và $\{v_i\}$ tăng ngặt ($v_1 < v_2 < \cdots < v_n$), bài toán trở thành chọn tối đa W vật từ n vật, sao cho tổng giá trị lớn nhất.

Lời giải: Vì $\{v_i\}$ tăng ngặt, để tối ưu tổng giá trị, ta nên chọn W vật có giá trị lớn nhất, tức là các vật từ v_{n-W+1} đến v_n.

$$\text{Tổng giá trị tối đa} = \sum_{i=n-W+1}^{n} v_i$$

Chứng minh:

- Mỗi vật có trọng lượng $w_i = 1$, nên ta chọn tối đa W vật ($\sum w_i \leq W$).
- Vì v_i tăng ngặt, tập hợp $\{v_{n-W+1}, \ldots, v_n\}$ có giá trị lớn nhất so với bất kỳ tập hợp W vật nào khác.

Câu c: $W_i = w, \forall i \in [n], \{v_i\} \in [n], \{v_i\}$ **tăng**

Giả sử $W \in \mathbb{N}^*, W_i = w, \forall i \in [n], \{v_i\}$ tăng (không nhất thiết ngặt, tức là $v_1 \leq v_2 \leq \cdots \leq v_n$).

Bài giải:

- Mỗi vật có trọng lượng w, nên số lượng vật tối đa có thể chọn là $k = \lfloor W/w \rfloor$.
- Vì $\{v_i\}$ tăng, để tối đa hóa tổng giá trị, ta chọn k vật có giá trị lớn nhất, tức là $v_{n-k+1}, v_{n-k+2}, \ldots, v_n$.
- Tổng giá trị tối đa là:

$$\sum_{i=n-k+1}^{n} v_i, \text{với } k = \lfloor W/w \rfloor$$

Giải thích:

- Tổng trọng lượng của k vật là $k \cdot w \leq W$.
- Nếu chọn nhiều hơn k vật, tổng trọng lượng sẽ vượt quá W.
- Vì v_i tăng, chọn k vật cuối cùng đảm bảo tổng giá trị lớn nhất.

Câu d: Thiết lập công thức DP và code

Cho $v_i \in (0, \infty), w_i \in \mathbb{N}^*, W \in \mathbb{N}^*$. Ta cần thiết lập công thức DP và cung cấp mã giả.

Công thức DP: Đặt $dp[i][j]$ là giá trị tối đa khi xét i vật đầu tiên với giới hạn trọng lượng j. Công thức truy hồi:

$$dp[i][j] = \max \begin{cases} dp[i-1][j] & \text{(không chọn vật } i\text{)} \\ dp[i-1][j-w_i] + v_i & \text{(chọn vật } i\text{, nếu } j \geq w_i\text{)} \end{cases}$$

Khởi tạo: $dp[0][j] = 0, \forall j \in [0, W]$.
Kết quả: $dp[n][W]$.
Mã giả:

Algorithm 2 Knapsack 0-1

```
Input: n, W, w[1..n], v[1..n]
Output: Giá trị tối đa
Khởi tạo mảng dp[0..n][0..W] = 0
for i = 1 to n do
    for j = 0 to W do
        dp[i][j] = dp[i − 1][j]
        if j ≥ w[i] then
            dp[i][j] = max(dp[i][j], dp[i − 1][j − w[i]] + v[i])
        end if
    end for
end for
return dp[n][W]
```

Độ phức tạp:

- Thời gian: $O(nW)$
- Không gian: $O(nW)$ (có thể tối ưu không gian xuống $O(W)$ bằng cách dùng mảng 1 chiều).

Week 07

Công thức truy hồi số Stirling loại 2

Bài giải:

Số Stirling loại 2, ký hiệu $S(n, k)$, biểu thị số cách phân chia một tập hợp có n phần tử thành k tập con không rỗng. Công thức truy hồi của số Stirling loại 2 được cho bởi:

$$S(n, k) = k \cdot S(n-1, k) + S(n-1, k-1),$$

với các điều kiện biên:

$$S(n, 0) = 0 \quad \text{với } n \geq 1,$$
$$S(0, 0) = 1,$$
$$S(n, k) = 0 \quad \text{với } k > n.$$

Giải thích công thức truy hồi:

Xét việc phân chia tập hợp $\{1, 2, \ldots, n\}$ thành k tập con không rỗng. Phần tử n có thể được xử lý theo hai cách:

- **Trường hợp 1:** Phần tử n nằm trong một tập con riêng biệt (tập con chỉ chứa n). Khi đó, các $n-1$ phần tử còn lại cần được phân chia thành $k-1$ tập con không rỗng, điều này được biểu thị bởi $S(n-1, k-1)$.
- **Trường hợp 2:** Phần tử n được thêm vào một trong k tập con đã có từ việc phân chia $n-1$ phần tử thành k tập con không rỗng. Có k lựa chọn để thêm n vào một tập con, và số cách phân chia $n-1$ phần tử thành k tập con là $S(n-1, k)$. Do đó, trường hợp này đóng góp $k \cdot S(n-1, k)$.

Kết hợp hai trường hợp trên, ta thu được công thức truy hồi:

$$S(n, k) = k \cdot S(n-1, k) + S(n-1, k-1).$$

Midterm Exam

Week 08