TỔ HỢP CƠ BẢN

Đỗ Ngọc Khánh

Chuyên Tin khoá 2012 – 2015

Khi không có hi vọng, bổn phận của chúng ta là sáng tạo ra nó.

Albert Camus

I. GIỚI THIỆU

Toán học tổ hợp là một ngành toán học rời rạc, nghiên cứu về các cấu hình kết hợp các phần tử của một tập hữu hạn phần tử. Các cấu hình đó là các hoán vị, chỉnh hợp, tổ hợp,... các phần tử của một tập hợp. Nó có liên quan đến nhiều lĩnh vực khác của toán học, cũng như đến các ngành ứng dụng như khoa học máy tính.

Trong khuôn khổ của bài viết này, tôi xin được trình bày về những vấn đề cơ bản nhất của tổ hợp, được ứng dụng nhiều trong các bài toán tin học và cách ứng dụng những điều cơ bản đó vào một vài bài toán.

II. MỘT SỐ CẤU HÌNH VÀ CÔNG THỨC CHÍNH

Với một tập hợp có n phần tử, ta có các cấu hình sau:

1. Hoán vị (Permutation)

– Hoán vị của một tập n phần tử là một cách sắp xếp n phần tử liên tiếp trên một đường thẳng.

– Số lượng hoán vị của tập n phần tử là: $P(n) = n!$.

2. Chỉnh hợp (Partial–permutation)

– Chỉnh hợp chập k $(0 \le k \le n)$ của tập n phần tử là một bộ sắp thứ tự k phần tử đôi một phân biệt của tập.

– Số lượng chỉnh hợp chập k của n phần tử là: $A(n,k) = (n)_k = \frac{n!}{(n-k)!}$.

– Hai tính chất của $(n)_k$: $(n)_0 = 1$ và $(n)_n = n!$.

3. Tổ hợp (Combination)

– Tổ hợp chập k $(0 \le k \le n)$ của tập n phần tử là một tập con gồm k phần tử của tập đó.

– Số lượng tổ hợp chập k của n phần tử là: $C(n,k) = \binom{n}{k} = \frac{n!}{n!(n-k)!}$.

– $\binom{n}{k}$ là kí hiệu của hệ số nhị thức (Binomial coefficient) và có các tính chất dưới đây:

+ $\binom{n}{0} = \binom{n}{n} = 1$.

+ Tính đối xứng: $\binom{n}{k} = \binom{n}{n-k}$.

+ Quy tắc Pascal: $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.

III. NHỮNG QUY TẮC, NGUYÊN LÍ CƠ BẢN

1. Quy tắc cộng (Rule of sum)

– Phát biểu quy tắc: Giả sử có n phương án đôi một khác nhau $A_1, A_2, \dots, A_n$ để hoàn thành một công việc. Trong đó, phương án A_i có k_i cách hoàn thành. Khi đó, ta có $k_1 + k_2 + \cdots + k_n$ cách để hoàn thành công việc.

– Biểu thức: Cho n tập đôi một phân biệt $S_1, S_2, \dots, S_n$, ta có: $\sum|S_i| = |\cup S_i|$.

– Ví dụ: Bạn cần mua một chiếc bánh, cửa hàng 1 có bán 3 loại bánh khác nhau, cửa hàng 2 có bán 2 loại bánh khác nhau. Vậy tổng cộng bạn có $3 + 2 = 5$ cách chọn loại bánh để mua.

2. Quy tắc nhân (Rule of product)

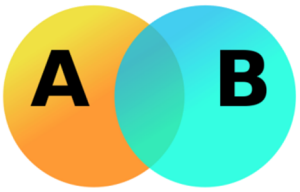
– Phát biểu quy tắc: Giả sử một công việc cần thực hiện lần lượt qua n giai đoạn $A_1, A_2, \dots, A_n$. Trong đó, giai đoạn A_i có k_i cách hoàn thành. Khi đó, ta có $k_1 k_2 \dots k_n$ cách để hoàn thành công việc.

– Biểu thức: Cho n tập $S_1,\ S_2, \dots ,\ S_n$, ta có: $\prod_{i=1}^{n}|S_i| = |S_1 \times \dots \times S_n|$.

+ Ở đây $\times$ là tích Descartes: $S_1 \times \dots \times S_n = \{(x_1, \dots, x_n) \mid x_i \in S_i\}$.

– Ví dụ: Bạn muốn mua một chiếc bánh, cửa hàng có các khuôn dạng bánh là tròn, vuông hoặc tam giác, số tầng bánh là 1, 2 hoặc 3, lớp kem phủ là kem tươi hoặc kem bơ. Vậy thì bạn sẽ có $3 \times 3 \times 2 = 18$ cách cách chọn bánh.

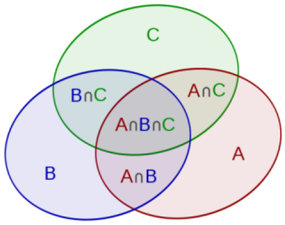
3. Nguyên lí bao hàm loại loại trừ (Inclusion–exclusion principle)



– Nguyên lí bao hàm loại trừ liên quan đến tổng số phần tử của nhiều tập hợp lại với nhau, số phần tử của mỗi tập và số phần tử thuộc phần giao nhau giữa hai hay nhiều tập bất kì.

– Phiên bản nhỏ nhất của nguyên lí này áp dụng cho hai tập A và B hữu hạn. Tổng số phần tử của hai tập hợp lại với nhau sẽ bằng tổng số phần tử của hai tập trừ đi số phần tử thuộc cả hai tập.

+ Biểu thức: $|A \cup B| = |A| + |B| - |A \cap B|$.



+ Trong các bài toán phiên bản này được sử dụng khá nhiều.

– Với ba tập hữu hạn chúng ta có: $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$.

– Tổng quát cho n tập hữu hạn $S_1, S_2, \dots, S_n$ ta có: Để đếm số phần tử của tập tạo bởi n tập hữu hạn hợp lại với nhau, đầu tiên ta tính tổng số phần tử của mỗi tập, rồi trừ đi số phần tử thuộc nhiều hơn một tập, rồi cộng với số phần tử thuộc nhiều hơn hai tập, rồi trừ đi số phần tử thuộc nhiều hơn ba tập, v.v...

$$\left|\bigcup_{i=1}^{n} S_i\right| = \sum_{k=1}^{n}(-1)^{k-1} \sum_{\substack{I \subseteq 1,n \\ |I|=k}} \left|\bigcap_{j \in I} S_j\right|.$$

$$|S_1 \cup \dots \cup S_n| = \sum_{i=1}^{n}|S_i| - \sum_{1 \le i < j \le n}|S_i \cap S_j| + \sum_{1 \le i < j < k \le n}|S_i \cap S_j \cap S_k| - \cdots + (-1)^{n-1}|S_1 \cap \dots \cap S_n|.$$

IV. MỘT SỐ BÀI TOÁN

Một số phần tóm tắt đề bài của các bài đã được thay đổi, giản lược, bỏ bớt một số ý không quan trọng so với đề bài gốc để tập trung vào nội dung chính của bài.

Trong các bài toán, để đơn giản, ta coi $\binom{n}{k} = 0$ và $(n)_k = 0$ khi $k < 0$ hoặc $k > n$.

BÀI 1: Tính hệ số nhị thức $\binom{n}{k} \bmod p$.

Với bài toán này ta có một số cách giải thông dụng sau:

– *Cách 1:* Áp dụng công thức $\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k!}$ để tính $\binom{n}{k}$ rồi chia lấy dư cho p.

+ Độ phức tạp tính toán: $O(k)$, bộ nhớ phụ: $O(1)$.

+ Ưu điểm: Thời gian tuyến tính, chi phí bộ nhớ phụ nhỏ.

+ Nhược điểm: Do có cả phép chia nên không thể tính tới đâu chia lấy phần dư tới đó, vì thế có thể phải xử lí số nguyên lớn làm tăng thời gian tính toán.

– *Cách 2:* Áp dụng quy tắc Pascal: $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.

+ Độ phức tạp tính toán: $O(nk)$, bộ nhớ phụ: $O(k)$ hoặc $O(nk)$.

+ Ưu điểm: Do chỉ sử dụng phép cộng nên có thể vừa tính vừa chia lấy phần dư, không cần xử lí số nguyên lớn.

+ Nhược điểm: Thời gian tính toán khá lớn, mất thêm bộ nhớ phụ.

+ Cách làm này thường được áp dụng cho những bài phải dùng tới rất nhiều hệ số nhị thức.

– *Cách 3:* (Áp dụng cho trường hợp p là số nguyên tố và $k < p$)

+ Biến đổi công thức ở cách 1 một chút, ta có: $\binom{n}{k} = (\prod_{i=n-k+1}^{n} i)(k!)^{-1}$. Như vậy, vấn đề chính của cách làm này sẽ là tính $a^{-1} \bmod p$ với $a = k!$.

> Vì p là số nguyên tố và $k < p$ nên $k!$ không chia hết cho p, theo dụng định lí Fermat nhỏ, ta có: $a^{p-1} \equiv 1 \ (mod\ p)$. Suy ra $a^{-1} \equiv a^{-1}a^{p-1} \equiv a^{p-2} \ (mod\ p)$. Việc tính $a^{p-2} \bmod p$ có thể tính trong thời gian $O(\log p)$. Việc tính $\binom{n}{k}$ theo cách này không còn gì khó khăn nữa.

+ Độ phức tạp tính toán: $O(k + \log p)$, bộ nhớ phụ: $O(1)$.

+ Ưu điểm: độ phức tạp tính toán gần như tuyến tính, bộ nhớ phụ nhỏ, do chỉ dùng phép nhân nên có thể vừa nhân vừa chia lấy phần dư cho p.

+ Nhược điểm: Không áp dụng được với n, p, k bất kì.

– *Cách 4:* Sử dụng định lí Lucas (Áp dụng cho trường hợp p là số nguyên tố)

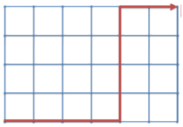
+ Ta đã thấy nhược điểm của cách làm trên là không áp dụng được với $k \geq p$ do ràng buộc của định lí Fermat bé. Với $k \geq p$ ta cần sử dụng định lí Lucas:

> *Định lí Lucas:* Với hai số m và n có biểu diễn trong hệ cơ số p tương ứng có dạng $\overline{m_{k-1} \dots m_0}_{(p)}$ và $\overline{n_{k-1} \dots n_0}_{(p)}$, ta có: $\binom{m}{n} \equiv \prod_{i=0}^{k-1} \binom{m_i}{n_i} \ (mod\ p)$. Quy ước rằng $\binom{n}{k} = 0$ nếu $n < k$.

+ Như vậy ta có thể phân tích hai số n và k theo hệ cơ số p sau đó tính mỗi hệ số nhị thức $\binom{n_i}{k_i}$ bằng cách 3.

Kết luận: Qua 4 cách giải vừa nêu, chúng ta thấy khi tính hệ số nhị thức cần phải linh hoạt, tuỳ vào từng bài toán mà áp dụng cho phù hợp. Vì hệ số nhị thức có tính đối xứng nên đôi lúc chúng ta cần cân nhắc xem có nên thay k bởi $n-k$ để giảm chi phí tính toán hay không.

BÀI 2: Đường đi trên lưới

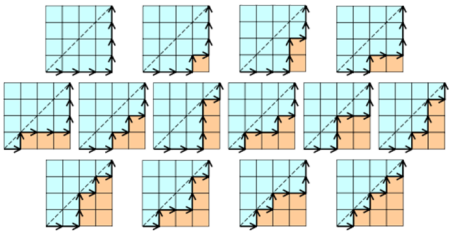


Cho một lưới hình chữ nhật kích thước $m \times n$. Hãy đếm số cách đi từ góc trái dưới – toạ độ $(0,0)$ đến góc phải trên – toạ độ (m,n) của lưới. Tại mỗi bước đi, ta được phép đi lên trên hoặc sang phải theo cạnh của một ô vuông trên lưới. Hình bên cho thấy một cách đi thoả mãn yêu cầu trên lưới 4×6.

Đây là một bài toán quy hoạch động khá kinh điển nhưng chúng ta sẽ xét nó dưới góc độ một bài toán tổ hợp.

Dễ thấy mọi đường đi thoả mãn yêu cầu đề bài đều phải đi đúng $m+n$ bước trong đó có m bước đi lên trên và n bước đi sang phải. Như vậy để xây dựng một đường đi ta chỉ cần chọn ra trong $m+n$ vị trí m vị trí để xếp bước đi lên trên, n vị trí còn lại dùng để xếp bước đi sang phải. Do vậy kết quả bài toán là: $\binom{n+m}{n} = \binom{n+m}{m}$.

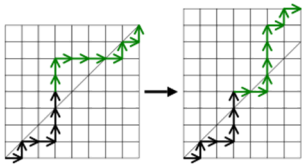
BÀI 3: Đường đi đơn điệu trên lưới vuông



Cho một lưới hình vuông kích thước $n \times n$. Hãy đếm số cách đi từ điểm $(0,0)$ đến điểm (n,n) với quy tắc di chuyển như [IV.2] và bổ sung thêm điều kiện là đường đi trên lưới không được cắt qua đường chéo phụ của lưới – đường thẳng nối điểm $(0,0)$ và điểm (n,n). Một đường đi như vậy gọi là đường đi đơn điệu (monotonic path). Hình bên cho thấy tất cả 14 *đường đi đơn điệu trên lưới* 4×4.

Theo [IV.2] ta có tổng số đường đi trên lưới $n \times n$ là $\binom{2n}{n}$.

Xét một đường đi không thoả mãn quy tắc trên (tạm gọi là đường đi không đơn điệu), dễ thấy điểm đầu tiên nằm bên trên đường chéo phụ sẽ có dạng $(a,\ a+1)$ với $0 \le a < n$. Trên đường đi từ $(0,0)$ đến $(a,\ a+1)$ ta đã đi sang phải a lần và lên trên $(a+1)$ lần, nói cách khác, từ $(a,\ a+1)$ tới $(n,\ n)$ ta còn phải đi sang phải $(n-a)$ lần và đi lên trên $(n-a-1)$ lần.



Bây giờ ta đổi hướng đường đi từ $(a,\ a+1)$ tới $(n,\ n)$, tức là đang sang phải chuyển thành đi lên và ngược lại. Khi đó, đường đi mới từ $(a,\ a+1)$ sẽ đi sang phải $(n-a-1)$ lần, đi lên trên $(n-a)$ lần. Cộng thêm số lần sang phải, đi lên của đường đi từ $(0,0)$ tới $(a,\ a+1)$ thì tổng cộng chúng ta đã đi sang phải $(n-1)$ lần và đi lên trên $(n+1)$ lần. Đây là một đường đi trên lưới $(n+1) \times (n-1)$. Hình bên cho thấy một đường đi không đơn điệu trên lưới 8×8 được đổi hướng để trở thành một đường đi trên lưới 9×7.

Như vậy, với mỗi đường đi không đơn điệu trên lưới $n \times n$, có duy nhất một đường đi trên lưới $(n+1) \times (n-1)$ tương ứng với đường đi này, và ngược lại (bạn đọc tự chứng minh điều ngược lại). Theo [IV.2] số đường đi trên lưới $(n+1) \times (n-1)$ là $\binom{2n}{n+1}$, chính là số đường đi không đơn điệu trên lưới $n \times n$.

Theo quy tắc cộng ta có số lượng đường đi đơn điệu là: $\binom{2n}{n} - \binom{2n}{n+1}$. Và đây cũng chính là giá trị của số Catalan thứ n.

BÀI 4: SuperSum (TopCoder SRM 467 – Division 1 – Level 2) [1]

Người ta định nghĩa hàm $f(k,n)$ như sau:

- $f(0,n) = n$ *với mọi* n *nguyên dương.*
- $f(k,n) = \sum_{i=1}^{n} f(k-1,i)$ *với mọi* k, n *nguyên dương.*

Hãy tính $f(k,n) \bmod p$ *với* $1 \le k \le 50$, $1 \le n \le 10^9$ *và* $p = 10^9 + 7$.

Lời giải:

Khá dễ để nhận ra rằng $f(k,n) = f(k-1,n) + f(k,n-1)$. Công thức này trông có vẻ khá "giống" với quy tắc Pascal của hệ số nhị thức, điều này làm chúng ta liên tưởng tới cách tính $f(k,n)$ dựa vào tam giác Pascal. Các bạn có thể tham khảo thêm về tam giác Pascal trong các tài liệu khác nhưng trong bài toán này chúng ta chỉ quan tâm tới tính chất: Giá trị ở hàng thứ n, vị trí thứ k (đếm từ trái hoặc từ phải sang) của tam giác Pascal là $\binom{n}{k}$.

Tính trước một vài giá trị của f ta có bảng sau:

k\n	1	2	3	4	5
0	1	2	3	4	5
1	1	3	6	10	15
2	1	4	10	20	35
3	1	5	15	35	70

Bảng này có vẻ như vẫn chưa cho chúng ta liên tưởng nào về tam giác Pascal. Bây giờ, không làm ảnh hưởng tới tính chất của hàm số, ta định nghĩa thêm giá trị $f(-1,n) = 1$ với mọi n nguyên dương, $f(0,n)$ vẫn bằng n, giá trị của hàm số không bị thay đổi. Thêm vào dòng $k = -1$, ta có bảng giá trị mới như sau:

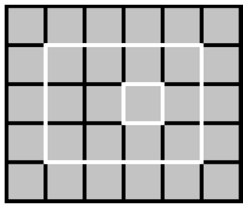
k\n	1	2	3	4	5
–1	1	1	1	1	1
0	1	2	3	4	5
1	1	3	6	10	15
2	1	4	10	20	35
3	1	5	15	35	70

Nhìn vào bảng trên ta thấy được rằng các đường chéo của bảng chạy theo hướng Tây Nam – Đông Bắc cho ta hình ảnh các hàng của tam giác Pascal. Nếu như cầm bảng và xoay sang phải 45° ta sẽ dễ dàng nhận ra giá trị $f(k,n)$ nằm ở hàng $(n+k)$, vị trí $(k+1)$ đếm từ phải sang của tam giác Pascal.

Vậy ta có kết luận: $f(n,k) = \binom{n+k}{k+1}$.

<u>BÀI 5:</u> Trò chơi với những hình chữ nhật (Codeforces Beta Round #94 Div. 1 Only – Problem C) [2]

Cho một lưới hình chữ nhật kích thước $m \times n$*, đếm số cách vẽ* k *hình chữ nhật lên lưới sao cho các cạnh của hình chữ nhật được vẽ dọc theo đường lưới, không có hai cạnh của hai hình chữ nhật nào có điểm chung, các cạnh phải nằm hoàn toàn bên trong lưới, tức là không được nằm trên đường biên*



lưới, hình chữ nhật vẽ sau phải nằm hoàn toàn bên trong hình vẽ trước. Hai cách vẽ được gọi là khác nhau nếu như hình ảnh sau cùng thu được là khác nhau. Kết quả đưa ra được chia lấy dư cho $(10^9 + 7)$.

Giới hạn: $1 \leq n, m, k \leq 1000$.

Lời giải:

Hình bên cho thấy một cách vẽ đúng quy tắc 2 hình chữ nhật lên lưới 5×6.

Một hình chữ nhật trên lưới chắc chắn được tạo bởi việc tô màu một phần 2 đường lưới dọc và 2 đường lưới ngang. Như vậy thì việc vẽ k hình chữ lên lưới chắc chắn đã tô màu một phần $2k$ đường lưới dọc và $2k$ đường lưới ngang. Hơn nữa, để thoả mãn được các quy tắc vẽ thì $2k$ đường lưới dọc này phải đôi một phân biệt, $2k$ đường lưới ngang cũng đôi một phân biệt.

Mà bên trong một lưới kích thước $m \times n$ có $m - 1$ đường lưới ngang và $n - 1$ đường lưới dọc nên việc tạo ra một hình vẽ thoả mãn là việc chọn ra $2k$ đường lưới ngang và $2k$ đường lưới dọc sau đó tô màu sao cho hình chữ nhật thứ i được tạo bởi hai đường lưới ngang i và $(2k - i + 1)$ kết hợp với hai đường lưới dọc i và $(2k - i + 1)$.

Theo quy tắc nhân, kết quả bài toán sẽ là $\binom{m-1}{2k}\binom{n-1}{2k}$.

<u>BÀI 6:</u> Greg và những người bạn (Codeforces Round #179 Div. 1 – Problem C) [3]

Có n *người cần đi qua sông, mỗi người nặng* 50 kg *hoặc* 100 kg. *Chỉ có một con thuyền để đi qua sông, trọng tải tối đa của thuyền là* k kg *và thuyền phải có người chèo mới di chuyển được. Xác định số lần chèo thuyền qua sông ít nhất có thể để cả* n *người sang được bờ bên kia và đếm số cách như vậy chia lấy dư cho* $(10^9 + 7)$. ***Hai cách qua sông được gọi là khác nhau nếu như có một lần chèo thuyền qua sông nào đó, tập người trên thuyền ở cách này khác với ở cách kia (*).***

Giới hạn: $1 \leq n \leq 50, 1 \leq k \leq 5000$.

Ví dụ:

Nếu có 3 người, 2 người nặng 50 kg và 1 người nặng 100 kg, trọng tải tối đa của thuyền là 100 kg. Thì số lần chèo thuyền qua sông ít nhất là 5 và có 2 cách như vậy, các lần chèo thuyên qua sông như sau:

(1) Hai người nặng 50 kg đi qua sông.
(2) Một người nặng 50 kg quay về.
(3) Người nặng 100 kg qua sông.
(4) Người nặng 50 kg quay về.
(5) Hai người nặng 50 kg qua sông.

Do có 2 cách chọn người nặng 50 kg nào sẽ quay về ở bước (2) nên có tất cả 2 cách qua sông cho 3 người này.

Lời giải:

Khi giải bài toán này phải rất chú ý tới điều kiện (*).

Gọi bờ lúc đầu cả n người đang đứng là bờ 0, bờ họ cần đi sang là bờ 1. Gọi tổng số người nặng 50 kg là X, tổng số người nặng 100 kg là Y. Ta có thể giải bài này sử dụng thuật toán loang theo lớp như sau. Mỗi trạng thái của bài toán là một bộ (x, y, k) trong đó x, y tương ứng là số người nặng $50, 100$ kg đứng ở bờ 0 và k là bờ mà thuyền đang đỗ. Từ trạng thái (x, y, k) có thể dẫn tới các trạng thái $(x', y', 1 - k)$ như sau:

– Nếu $k = 0$, $x' \leq x$, $y' \leq y$ và $0 < 50(x - x') + 100(y - y') \leq k$ thì có $\binom{x}{x-x'}\binom{y}{y-y'}$ cách chuyển từ trạng thái (x, y, k) sang trạng thái $(x', y', 1 - k)$. Đây là trường hợp thuyền đi xuôi.

– Nếu $k = 1$, $x' \geq x$, $y' \geq y$ và $0 < 50(x' - x) + 100(y' - y) \leq k$ thì có $\binom{X-x}{x'-x}\binom{Y-y}{y'-y}$ cách chuyển từ trạng thái (x, y, k) sang trạng thái $(x', y', 1 - k)$. Đây là trường hợp thuyền đi ngược.

– Nếu không có điều kiện nào trong hai điều kiện trên được thoả mãn thì không có cách nào chuyển trực tiếp giữa hai trạng thái (x, y, k) và $(x', y', 1 - k)$.

Vậy ta có thể tiến hành loang theo lớp từ trạng thái xuất phát $(X, Y, 0)$ đến trạng thái đích $(0,0,1)$ kết hợp với đếm số đường đi như vậy.

Lời giải trên có độ phức tạp $O(n^4)$.

BÀI 7: Phép chia hết (VOJ – PBCDIV) [4]

Có bao nhiêu số nguyên dương không vượt quá N chia hết cho đúng hai trong ba số 4, 6 *và* 15*?*

Giới hạn: $N \leq 10^{18}$.

Lời giải:

Nếu gọi $f(N, S)$ là số lượng số nguyên dương không vượt quá N chia hết cho tất cả các số trong tập S thì ta có kết quả bài toán theo nguyên lí bao hàm loại trừ là: $f(N, \{4,6\}) + f(N, \{4,15\}) + f(N, \{6,15\}) - 3f(N, \{4,6,15\}) = \left\lfloor \frac{N}{12} \right\rfloor + \left\lfloor \frac{N}{30} \right\rfloor - 2\left\lfloor \frac{N}{60} \right\rfloor$.

BÀI 8: Sâu ăn lá (INOI Online – September 2004 – Advanced Division) [5]

Cho một dãy A gồm K số nguyên $a_1, a_2, \dots, a_n$ *và một số nguyên N, hãy đếm xem có bao nhiêu số nguyên dương không vượt quá N chia hết cho ít nhất một trong K số của dãy A.*

Giới hạn: $1 \leq N \leq 10^9, 1 \leq K \leq 20, 1 \leq a_i \leq N$.

Lời giải:

Theo nguyên lí bao hàm loại trừ thì số lượng số cần tìm là số lượng các số nguyên dương không vượt quá N chia hết cho ít nhất 1 trong K số, trừ đi số lượng các số chia hết cho ít nhất 2 trong K số, cộng với số lượng các số chia hết cho ít nhất 3 trong K số, v.v... Mà số lượng số nguyên dương không vượt quá N chia hết cho x là $\left\lfloor \frac{N}{x} \right\rfloor$, một số chia hết cho cả q số thì sẽ chia hết cho bội chung nhỏ nhất của q số đó nên kết quả bài toán là:

$$S = \sum_{i=1}^{K} \left\lfloor \frac{N}{a_i} \right\rfloor - \sum_{1 \leq i < j \leq K} \left\lfloor \frac{N}{\operatorname{lcm}(a_i, a_j)} \right\rfloor + \cdots + (-1)^{K-1} \left\lfloor \frac{N}{\operatorname{lcm}(a_1, \dots, a_K)} \right\rfloor.$$

Để tính được giá trị S này thì chúng ta phải xét tất cả các tập con khác rỗng của tập K phần tử, vậy thì cách cài đặt đơn giản nhất là quay lui.

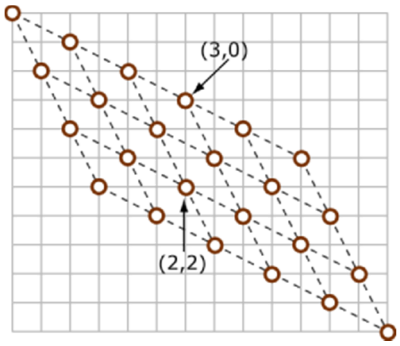
Lời giải trên có độ phức tạp vào khoảng $O(2^K \log_{10}^2 N)$, phù hợp với giới hạn của bài toán.

BÀI 9: Quân mã (Google Code Jam 2008 – Round 3) [6]

Trên một bàn cờ kích thước $H \times W$ *có một quân mã đang đứng ở góc trái trên – ô* $(1, 1)$. *Quân mã cần di chuyển tới góc phải dưới – ô* (H, W). *Quân mã di chuyển theo hình chữ "L", chỉ đi từ ô có chỉ số hàng cột nhỏ hơn tới ô có chỉ số hàng cột lớn hơn và không đi ra khỏi bàn cờ. Trên bàn cờ có R ô cấm đôi một phân biệt, quân mã sẽ không bao giờ đi vào những ô này. Các ô xuất phát và ô đích không bao giờ bị cấm. Hỏi quân mã có bao nhiêu cách di chuyển từ ô xuất phát tới ô đích? Kết quả đưa ra được chia lấy dư cho số nguyên tố* $(10^4 + 7)$.

Giới hạn: $1 \leq H, W \leq 10^8, 0 \leq R \leq 10$.

Lời giải:



Nếu vẽ vùng di chuyển của quân mã ra ta sẽ thấy nó có dạng hình bình hành với góc trái trên $(1,1)$ cố định, hơn nữa quân mã chỉ di chuyển theo hai vector $(2,1)$ và $(1,2)$ nên ta có thể quy đổi từ ô (x,y) thuộc lưới bình hành thành ô (x',y') thuộc lưới ô vuông theo đẳng thức sau: $x'(2,1)+y'(1,2)+(1,1)=(x,y)$, giải ra ta được $x'=x-1-\frac{x+y-2}{3}$ và $y'=y-1-\frac{x+y-2}{3}$. (Tham khảo hình bên).

Như vậy ta đã có thể quy đổi tất cả toạ độ của các ô cấm, điểm xuất phát, điểm đích sang hệ toạ độ mới, khi đó bài toán của chúng ta có thể được giải quyết đơn giản hơn rất nhiều do nó đã được quy về đường đi trên lưới ô vuông. Trong quá trình quy đổi ta bỏ qua tất cả các điểm có toạ độ mới không nguyên. Nếu điểm đích có toạ độ không nguyên thì hiển nhiên không có cách đi từ điểm xuất phát tới điểm đích.

Bài toán bây giờ có thể phát biểu lại như sau: Trên một lưới ô vuông kích thước $M \times N$ có P ô cấm, đếm số cách di chuyển từ ô $(1,1)$ tới ô (M,N) của lưới mà không đi vào ô cấm, mỗi bước chỉ đi từ ô (i,j) tới ô $(i+1,j)$ hoặc $(i,j+1)$.

Để giải bài toán trên ta sử dụng nguyên lí bao hàm loại trừ. Kết quả bài toán là tổng số đường đi đi qua chẵn ô cấm trừ đi tổng số đường đi đi qua lẻ ô cấm. Mà ta có số cách đi lần lượt qua n ô $(x_1,y_1),(x_2,y_2),\dots,(x_n,y_n)$ với $x_1 \le x_2 \le \cdots \le x_n$ và $y_1 \le y_2 \le \cdots \le y_n$ là $\prod_{i=1}^{n-1} \binom{x_{i+1}-x_i+y_{i+1}-y_i}{x_{i+1}-x_i}$ (Bạn đọc tự chứng minh). Vậy bài toán tới đây có thể giải quyết dễ dàng.

Điều đáng chú ý ở bài này là hệ số nhị thức ở đây cần tính dựa vào định lí Lucas, ta cần tính trước hai hàm $P(n)=n! \bmod p$ và $I(n)=(n!)^{-1} \bmod p$ với $p=10^4+7$ trong thời gian $O(p \log p)$. Sau đó mỗi hệ số nhị thức được tính trong thời gian xấp xỉ $O(1)$.

Bài toán có lời giải với độ phức tạp $O(2^R + p \log p)$.

<u>BÀI 10:</u> Trang trí cây thông (Codeforces Round #100 – Problem E) [7]

Có một cây thông cao n tầng và m loại bóng đèn có màu sắc khác nhau, mỗi loại bóng đèn có vô hạn bóng. Người ta cần trang trí cây thông sao cho tầng thứ i được trang trí bởi một dây đèn màu độ dài l_i, màu sắc của hai bóng đèn liên tiếp trên một dây đèn là khác nhau và tập màu sắc bóng đèn trên hai tầng liên tiếp khác nhau. Hãy đếm số cách trang trí cây thông hợp lệ chia lấy dư cho p. Hai cách trang trí được gọi là khác nhau nếu như có một tầng mà dây đèn ở cây thông này khác dây đèn ở cây thông kia. Giới hạn: $1 \le n,m \le 10^6, 2 \le p \le 10^9, 1 \le l_i \le 5000, L=\sum l_i \le 10^7$.

Lời giải:

Xét một tầng của cây thông, gọi $g(l,c)$ là số cách trang trí hợp lệ một dây đèn độ dài l, sử dụng hết một dãy màu đèn đôi một phân biệt gồm c phần tử. Có các khả năng sau:

– Nếu như chỉ có đèn l có màu c thì có $g(l-1,c-1)$ dây đèn như vậy.

– Nếu như cả c màu đèn đều đã được dùng cho $(l-1)$ đèn đầu tiên thì sẽ có $(c-1)$ cách chọn màu cho đèn l (do đèn thứ l phải có màu khác với đèn $(l-1)$), tức là có $(c-1)g(l-1,c)$ dây đèn như vậy.

Vậy thì: $g(l,c) = g(l-1,c-1) + (c-1)g(l-1,c)$. Quy ước $g(0,0) = 1$.

Chú ý: $g(l,c)$ là số cách trang trí dây đèn sử dụng hết một *dãy* màu c phần tử, nếu như phải sử dụng hết một *tập* màu gồm c phần tử thì sẽ có $g(l,c)c!$ cách trang trí dây đèn sử dụng hết các màu trong tập này.

Gọi $f(h,c)$ là số cách trang trí hợp lệ cây thông cao h tầng, tầng h được trang trí bởi đúng c màu đèn đôi một phân biệt. Như vậy thì cách trang trí hợp lệ cây thông cao h tầng sẽ là: $P(h) = \sum_{k=1}^{l_h} f(h,k)$ (bởi vì số màu đèn không thể vượt quá số đèn). Quy ước $f(0,0) = 1$, các giá trị $f(h,c)$ khác được tính như sau dựa trên hai cách trang trí sau:

– *Trang trí loại* 1*:* Trang trí cây thông cao h tầng, tầng h được trang trí bởi đúng c màu đèn đôi một phân biệt.

+ Số cách chọn một tập màu c phần tử cho tầng h là $\binom{m}{c}$.

+ Số cách trang trí hợp lệ cây thông cao $(h-1)$ tầng là $P(h-1)$.

+ Số cách trang trí hợp lệ dây đèn sử dụng một tập c màu cho tầng h là $g(l_h,c)c!$.

Suy ra số cách trang trí loại 1 là: $D_1 = g(l_h,c)c!\binom{m}{c}P(h-1) = g(l_h,c)(m)_c P(h-1)$.

– *Trang trí loại* 2*:* Trang trí cây thông cao h tầng, tầng h được trang trí bởi đúng c màu đèn đôi một phân biệt, tầng $(h-1)$ cũng được trang trí bởi đúng c màu đèn giống tầng h.

+ Số cách trang trí hợp lệ cây thông cao $(h-1)$ tầng, tầng $(h-1)$ sử dụng đúng c màu là $f(h-1,c)$.

+ Số cách trang trí hợp lệ dây đèn sử dụng chính tập c màu đã trang trí cho tầng $(h-1)$ để trang trí cho tầng h là: $g(l_h,c)c!$.

Suy ra số cách trang trí loại 2 là: $D_2 = g(l_h,c)c!\,f(h-1,c)$.

Vậy ta có: $f(h,c) = D_1 - D_2 = g(l_h,c)[(m)_c P(h-1) - c!\,f(h-1,c)]$.

Về độ phức tạp thuật toán thì hàm g có thể tính toán trong thời gian $O(\max^2 l_i)$. Ta phải tính tất cả L giá trị của hàm f. Nếu như số lượng chỉnh hợp chập, giai thừa được tính trước trong thời gian $O(L)$ thì thời gian tính toán f cũng là $O(L)$. Vậy thì lời giải trên có độ phức tạp $O(L + \max^2 l_i)$ phù hợp với giới hạn của bài toán.

<u>BÀI 11:</u> Du lịch (Codeforces Round #198 Div. 1 – Problem A) [8]

Có N địa điểm thú vị đôi một phân biệt nằm tại các vị trí $A_1, A_2, \ldots, A_N$ trên trục toạ độ. Iahub muốn xuất phát từ gốc toạ độ đi thăm N địa điểm trên, mỗi địa điểm đúng một lần (đi lướt qua không tính là đã thăm), thời gian đi từ điểm x tới điểm y là $|x-y|$. Tính trung bình cộng độ dài của tất cả các đường đi mà Iahub có thể đi. Kết quả đưa ra ở dạng phân số tối giản.

Giới hạn: $2 \le N \le 10^5, 1 \le A_i \le 10^7$.

Lời giải:

Đầu tiên ta tính tổng độ dài các đường đi. Để có thể bỏ dấu giá trị tuyệt đối ta sẽ đếm xem mỗi một toạ độ của điểm thú vị nhận dấu cộng bao nhiêu lần (số lần nhận dấu cộng có thể âm).

Sắp xếp lại dãy A theo thứ tự tăng dần, xét phần tử A_i:

– Nếu đi thăm A_i đầu tiên, thì A_i sẽ nhận dấu cộng $(N-1)!$ lần do đứng đầu dãy, nhưng ngoài ra còn nhận thêm $(i-1)$ dấu cộng và $(N-i)$ dấu trừ tuỳ theo điểm thăm tiếp theo. Tức là A_i nhận $(N-1)! + (2i-N+1)$ dấu cộng nếu được đi thăm đầu.

– Tương tự nếu A_i được đi thăm cuối cùng, nó sẽ nhận thêm $(2i-N+1)$ dấu cộng nữa.

– Nếu A_i được đi thăm từ thứ 2 đến thứ $N-1$, nó sẽ nhận thêm $2(N-3)!\,(i-1)_2$ dấu cộng và $2(N-3)!\,(N-i)_2$ dấu trừ tuỳ theo vị trí của hai điểm thăm liền trước, liền sau, và thứ tự thăm của $(N-3)$ điểm còn lại. Vậy tổng cộng khi đi thăm A_i từ thứ 2 đến thứ $N-1$, A_i nhận thêm $2(N-2)!\,[(i-1)_2 - (N-i)_2]$ dấu cộng.

Như vậy với tất cả các vị trí có thể của A_i, A_i đã góp vào tổng độ dài tất cả các đường đi một lượng $(N-1)!\,A_i(4i-2N-1)$.

Vậy ta có kết quả bài toán là: $S = \frac{\sum (N-1)! A_i (4i-2N-1)}{N!} = \frac{\sum A_i (4i-2N-1)}{N}$.

Bài toán có lời giải với độ phức tạp $O(N \log N)$.

V. MỘT SỐ BÀI LUYỆN TẬP

Bạn đọc có thể tham khảo các bài [9 – 17].

Như vậy, qua một vài bài tập nói trên, các bạn đã có thể thấy được mức độ đa dạng của những bài toán có liên quan tới tổ hợp. Tôi hi vọng bài viết của mình sẽ giúp ích cho các bạn trong quá trình học tập và nghiên cứu.

Chúc các bạn thành công!

TÀI LIỆU THAM KHẢO

[1] http://community.topcoder.com/stat?c=problem_statement&pm=10239&rd=14151

[2] http://codeforces.com/contest/128/problem/C

[3] http://codeforces.com/contest/295/problem/C

[4] http://vn.spoj.com/problems/PBCDIV/

[5] http://www.iarcs.org.in/inoi/contests/sep2004/Advanced–2.php

[6] https://code.google.com/codejam/contest/32002/dashboard#s=p3

[7] http://codeforces.com/contest/140/problem/E

[8] http://codeforces.com/contest/341/problem/A

[9] http://codeforces.com/contest/341/problem/C

[10] http://codeforces.com/contest/145/problem/C

[11] http://codeforces.com/contest/327/problem/C

[12] http://codeforces.com/contest/252/problem/C

[13] http://codeforces.com/contest/296/problem/B

[14] http://codeforces.com/contest/52/problem/B

[15] http://community.topcoder.com/stat?c=problem_statement&pm=7601&rd=10673

[16] http://community.topcoder.com/stat?c=problem_statement&pm=1659&rd=4515

[17] NEERC 2011, Northern Subregional Contest, Problem H: http://neerc.ifmo.ru/past/2011/northern/problems.pdf

[18] http://en.wikipedia.org/wiki/Catalan_number

[19] http://en.wikipedia.org/wiki/Fermat's_little_theorem

[20] http://en.wikipedia.org/wiki/Lucas'_theorem

[21] http://betterexplained.com/articles/navigate–a–grid–using–combinations–and–permutations/

[22] http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=combinatorics