1. **CÁC BÀI TOÁN CƠ BẢN**
2. **Bài toán 1**

Cho tập hợp tất cả các xâu nhị phân có độ dài N được sắp xếp theo thứ tự từ điển. Ví dụ N = 3 ta có tập hợp: 000, 001, 010, 011, 100, 101, 110, 111.

Trong tập hợp trên, ta có:

* Xâu nhị phân 100: ở vị trí số 5.
* Vị trí số 7 trong tập là xâu nhị phân: 110.

***Yêu cầu:*** Với một số N (N <= 63) cho trước, hãy cho biết:

* Xâu nhị phân S (có độ dài N) nằm ở vị trí nào của tập.
* Vị trí thứ X (1 <= X <= 1018) là xâu nhị phân nào?

***Dữ liệu vào:*** Cho trong file **NHIPHAN.INP**:

* Dòng đầu chứa một số nguyên N, là độ dài của các xâu nhị phân.
* Dòng thứ hai chứa một xâu nhị phân S có độ dài bằng N.
* Dòng thứ ba chứa một số nguyên X.

***Dữ liệu ra:*** In ra file **NHIPHAN.OUT**:

* Dòng đầu tiên chứa một số nguyên là vị trí của S trong tập hợp.
* Dòng thứ hai ghi xâu nhị phân có N phần tử nằm ở vị trí thứ X.

|  |  |
| --- | --- |
| **NHIPHAN.INP** | **NHIPHAN.OUT** |
| 3  100  7 | 5  110 |

***Phân tích****:*

Thực tế vị trí của xâu nhị phân S trong tập chính là dạng thập phân của S cộng thêm 1. Tuy nhiên để làm quen với Quy hoạch động vị trí cấu hình, ta sẽ làm theo cách sau:

*a. Xác định số lượng cấu hình thuộc tập trên khi biết một đoạn các tiền tố của các cấu hình đó.*

Giả sử ta biết được một đoạn tiền tố a1­a2…aK (K <= N) của một xâu nhị phân n phần tử. Vấn đề đặt ra là có bao nhiêu xâu thuộc tập hợp có đoạn tiền tố như trên.

Với bài toán trên ta dễ dàng nhận thấy rằng: Số bit nhị phân còn lại chưa biết là (N – K) bit mà mỗi bit chỉ nhận giá trị 0 hoặc 1 nên số phần tử thỏa mãn là 2N - K .

Vậy gọi F[i] là số lượng phần tử thỏa mãn khi biết (N – i) phần tử tiền tố.

       F[0] = 1

       F[i] = F[i – 1] \* 2

*b. Xác định xâu nhị phân S nằm vị trí nào?*

Ta xét xâu nhị phân S: a1a2a3…aN-1­aN.

Ta chỉ quan tâm đến các bit 1 của S, giả sử bit 1 xuất hiện ở vị trí t thì S: a1a2…at-11 …aN-1aN

Khi đó S sẽ có vị trí lớn hơn vị trí những cấu hình có dạng là a1a2…at-1 0 xt+1…xn-1xn(trong đó xi nhận các giá trị 0 hoặc 1). Như đã tính ở trên thì sẽ có F[N - t] cấu hình có dạng a1a2…at-1 0 xt+1…xN-1xN có vị trí bé hơn S.

Gọi vt là số lượng các cấu hình có vị trí bé hơn S.

        vt = ∑ F[N - ti], với ti­ là vị trí xuất hiện các bit 1 trong S.

Suy ra, vị trí của S trong tập hợp là vt + 1.

*c. Xác định xâu nhị phân nào nằm ở vị trí K?*

Để tìm cấu hình S ở vị trí K, ta duyệt qua tất các bit của S từ 1 đến N. Nếu bit thứ t của S nhận giá trị 1 : a1a2…at-11 …aN-1aN thì rõ ràng S lớn hơn F[t – 1] cấu hình có dạng a1a2…at-1 0 xt+1…xN-1xN. Thế nên:

- Nếu k <= F[N – t] thì bit thứ t của S nhận giá trị 0.

- Nếu k > F[N – t] thì bit thứ t của S nhận giá trị 1 và gán lại k = k – F[N – t].

1. **Bài toán 2**

Một tập hợp S gồm các dãy N bit 0, 1 trong đó không có hai bit 1 nào kề nhau. Ví dụ N = 5 thì S gồm các dãy 00000, 00001, 00101,… Tập S được sắp xếp theo thứ tự từ điển.

***Yêu cầu:*** Cho một số nguyên N (N < 63) cho biết:

* Xâu nhị phân S (có độ dài N) nằm ở vị trí nào của tập.
* Vị trí thứ K (K <= 1018) là xâu nhị phân nào?

***Dữ liệu vào:*** Cho trong file **NHIPHAN2.INP**:

* Dòng đầu chứa một số nguyên N, là độ dài của các xâu nhị phân.
* Dòng thứ hai chứ một xâu nhị phân S có độ dài bằng N.
* Dòng thứ ba chứa một số nguyên K.

***Dữ liệu ra:*** In ra file **NHIPHAN2.OUT**:

* In trên từng dòng là kết quả từng yêu cầu của bài toán.

|  |  |
| --- | --- |
| **NHIPHAN2.INP** | **NHIPHAN2.OUT** |
| 5  00001  3 | 2  00010 |

***2.1 Phân tích:***

*a. Xác định số lượng cấu hình thuộc tập trên khi biết một đoạn các tiền tố của các cấu hình đó.*

Gọi F[i] là số lượng dãy nhị phân có i bit mà không có hai bit 1 liền kề nhau.

F[0] = 1, F[1] = 2.

F[i] = F[i – 1] (nếu chọn bit i = 0) + F[i – 2] (nếu chọn bit i = 1)

Giả sử ta biết được một đoạn tiền tố K phần tử a1­a2…aK xK+1…xN-1xN (K <= N) của một xâu nhị phân n phần tử thì số lượng cấu hình phù hợp là:

- Nếu ***bit thứ*** K = 0 thì sẽ có F[N – K] cấu hình phù hợp.

- Nếu ***bit thứ*** K = 1 thì sẽ có F[N – K – 1] cấu hình phù hợp do bit thứ K + 1 phải bằng 0.

*b. Xác định xâu S nằm ở vị trí nào và xâu nào nằm ở vị trí K ?*

Ta nhận thấy hai yêu cầu trên làm tương tự như bài toán 1.

1. **Bài toán 3:**

Xét tất cả các hoán vị của dãy số tự nhiên (1, 2, …, N) (1 <= N <= 20), các hoán vị được sắp xếp theo thứ tự từ điển.

***Yêu cầu:***

* Cho trước 1 hoán vị: Tìm số hiệu hoán vị đó trong dãy đã sắp xếp.
* Cho trước số hiệu của 1 hoán vị trong dãy đã sắp xếp, tìm hoán vị đó.

***Dữ liệu vào:*** Cho trong file **SHHV.inp**:

* Dòng đầu tiên chứa số nguyên dương N.
* Dòng thứ hai chứa N số a1, a2, …, aN (dãy hoán vị N phần tử).
* Dòng thứ ba chứa số K (số hiệu hoán vị trong dãy N phần tử).

***Dữ liệu ra:*** In ra file **SHHV.out**:

* Dòng đầu tiên ghi số K (số hiệu hoán vị của dãy ai).
* Dòng thứ hai ghi N số b1, b2, …, bN (dãy hoán vị có số hiệu K).

|  |  |
| --- | --- |
| **SHHV.INP** | **SHHV.OUT** |
| 3  2 1 3  4 | 3  2 3 1 |

***3.1 Phân tích***

*a. Xác định số lượng cấu hình thuộc tập trên khi biết một đoạn các tiền tố của các cấu hình đó.*

Giả sử ta biết được một đoạn tiền tố K phần tử a1­a2…aK xK+1…xN-1xN (K <= N) của một hoán vị N phần tử thì số lượng cấu hình phù hợp là: (N – K)! do ta có N – K số chưa dùng có thể điền vào bất kì vị trí nào còn trống.

Gọi F[i] là số lượng hoán vị có i chữ số.

F[i] = i! hay F[i] = F[i – 1] \* i

*b. Xác định vị trí của dãy hoán vị cho trước:*

Ta xét hoán vị S: a1­a2…a­­N-1aN.

Để biết vị trí K của hoán vị S, ta duyệt qua tất cả các phần tử của S từ 1 đến N. Giả sử ta đang xét đến phần tử t của S thì ta được một đoạn tiền tố S cố định: a1a2…at xt+1…xN với (xt+1…xN là một hoán vị của các phần tử chưa xuất hiện trong t phần tử đầu).

Nếu tồn tại 1 phần tử M chưa xuất hiện trong t phần tử đầu và M < at (1) thì ta thấy a1a2…at xt+1…xN> a1a2…M xt+1…xN. Số lượng cấu hình thõa mãn a1a2…M xt+1…xN là F[N – T]. Vậy cứ tương ứng với mỗi M thõa mãn (1) thì S có số hiệu hoán vị lớn hơn F[N – t] cấu hình khác.

Gọi vt là số lượng cấu hình bé hơn S:

vt = ∑ find(t) \* F[N – t], với find(t) là số lượng các số chưa xuất hiện trong t phần tử đầu và có giá trị bé hơn at.

Suy ra, vị trí của S trong tập hợp là vt + 1.

*c. Xác định hoán vị có vị trí cho trước:*

Để tìm cấu hình S ở vị trí K, ta duyệt qua tất các phần tử của S từ 1 đến N. Nếu phần tử thứ t của S nhận giá trị M: a1a2…at-1M …aN-1aN thì rõ ràng S lớn hơn find(t) \* F[N - t] cấu hình có dạng a1a2…at-1 0 xt+1…xN-1xN. Thế nên, ta duyệt qua tất cả các phần tử M chưa xuất hiện trong t – 1 phần tử đầu theo thứ tự từ bé đến lớn:

* Nếu K <= F[N – t] thì phần tử thứ t của S nhận giá trị M.
* Nếu k > F[N – t] thì gán lại k = k – F[N – t].