

CHUYÊN ĐỀ NÂNG CAO 6

CHUYÊN ĐỀ 6: QUY HOẠCH ĐỘNG**E. QUY HOẠCH ĐỘNG VỊ TRÍ CẤU HÌNH****I. MỘT SỐ BÀI TOÁN ĐIỂN HÌNH:****1. Nhiphan1.pas**

Cho tập A gồm tất cả xâu nhị phân có độ dài N được sắp xếp theo thứ tự từ điển. Ví dụ với $N=3$ ta có tập A như sau:

Vị trí	Dãy nhị phân
1	000
2	001
3	010
4	011
5	100
6	101
7	110
8	111

Trong tập trên ta thấy

- Dây nhị phân 110: ở vị trí số 7
- vị trí thứ 4 trong tập là dãy nhị phân: 011

Yêu cầu: Với một số N bất kỳ ($n \leq 100$), bạn hãy cho biết:

- Dây nhị phân S nằm ở vị trí nào?
- Vị trí thứ K ($k < 10^{18}$): là dãy nhị phân bao nhiêu?

Dữ liệu vào ra:

Nhiphan1.inp	Nhiphan1.out
4	14
1101	0111
8	

Với bài toán này ta có cách đơn giản: Vị trí của dãy nhị phân chính là giá trị thập phân tương ứng. Nhưng để làm quen với Quy hoạch động vị trí cấu hình ta sẽ làm theo cách sau:

Bây giờ ta xét theo từng yêu cầu của bài toán:

*** Đầu tiên: Phần này chung cho cả hai yêu cầu:**

Ta xây dựng mảng một chiều F như sau: $F[i]$ là số lượng dãy nhị phân có độ dài i bit.

Khi đó ta có mảng F sẽ được tính bằng quy hoạch động:

- $F[0] = 1$
- $F[i] = 2 * F[i-1]$ Với $i=1..n$

Mã chương trình tính F:

```
F[0] := 1;
For i:=1 to n do F[i] := 2*F[i-1];
```

Thật vậy, ví dụ: Với $N = 4$ ta có:

- $F[0] = 1$
- $F[1] = 2 * F[0] = 2$ Gồm hai dãy nhị phân có độ dài 1: 0, 1
- $F[2] = 2 * F[1] = 4$ Gồm 4 dãy: 00, 01, 10, 11
- $F[3] = 2 * F[2] = 8$ Gồm 8 dãy: 000, 001, 010, 011, 100, 101, 110, 111
- $F[4] = 2 * F[3] = 16$ Gồm 16 dãy: 0000, 0001,..., 1110, 1111.

*** Yêu cầu 1: Hỏi dãy nhị phân S nằm vị trí nào?**

Ta xem dãy S như sau:

- 1 Ta nhận thấy nếu $S[i] = 1$ thì khi đó ta thấy có $F[n-i]$ dãy nhị phân có vị trí bé hơn vị trí của S. Ví

dụ: $S = '0100'$ ta có $S_2 = 1$ vậy thì ta có $F[4-2] = F[2] = 4$ dãy nhị phân có vị trí bé hơn S . Đó là $000, 001, 010, 011$.

2 Vậy từ đây ta có:

- Vt:=0;
- Xét tất từ 1 đến độ dài của length(s): Nếu vị trí nào bằng 1 thì vt:= vt + F[n-i]
- Kết quả vị trí của xâu S trong tập đã xếp theo từ điển chính là vt+1.

Ví dụ: Với $n=4$ và $S = '0110'$ ta có

- Vt:=0;
- Xét từ 1 đến length(s): ta có
 - Vt = vt + f[n-2] = 4 Vị trí $S_2 = '1'$
 - Vt = vt + f[n-3] = 4 + f[1] = 6 Vị trí $S_3 = '1'$
- Kết quả S nằm ở vị trí vt+1 = 7.

Vậy ta có đoạn chương trình giải quyết yêu cầu số 1 như sau:

```
Vt:=0;
For i:=1 to length(s) do if s[i] = '1' then vt:=vt+f[n-i] ;
```

*** Yêu cầu 2: Hỏi dãy nhị phân nằm ở vị trí K?**

Đầu tiên ta cho dãy nhị phân nằm ở vị trí k là xâu C:

Ta nhận thấy:

- Nếu $K > F[n-i]$ thì {bít thứ i của C là $C_i = 1$ và $k:=k-F[n-i]$ }

Ví dụ: Với $k=3$ ta có: $k > F[1]$ nên $C_3 = 1$ (đó là 0010)

- Ta sẽ lần lượt xét i từ bít 1 đến bít n của C:

Nếu $k > F[n-i]$ thì $C_i = 1$ và K còn lại là $K = K - F[n-i]$

Để dễ hiểu cho đoạn trên ta thử với $n=4$: Hỏi dãy nhị phân ở vị trí $K=3$

- Ban đầu cho $C='0000'$
- Xét i từ 1 đến 4:
 - $i=1$: $k > F[4-1]$ là sai
 - $i=2$: $K > F[4-2]$ là sai
 - $i=3$: $K > F[4-3]$ tức $K=3 > F[1]=2$ nên ta có: $C_3 = 1$ và $K = K - F_1 = 3 - 2 = 1$
 - $i=4$: $K=1 > F[4-4]=1$ là sai
- Kết quả $C = '0010'$ chính là đáp án.

Ví dụ: với $n=4$: Hỏi dãy nhị phân ở vị trí $K=7$

- Ban đầu cho $C='0000'$
- Xét i từ 1 đến 4:
 - $i=1$: $k > F[4-1]$ là sai
 - $i=2$: $K > F[4-2]$ tức $k=7 > F[2]=4$ nên ta có: $C_2=1$ và $k=k-f_2 = 7-4 = 3$
 - $i=3$: $K > F[4-3]$ tức $K=3 > F[1]=2$ nên ta có: $C_3 = 1$ và $K = K - F_1 = 3 - 2 = 1$
 - $i=4$: $K=1 > F[4-4]=1$ là sai
- Kết quả $C = '0110'$ chính là đáp án.

Vậy ta có đoạn chương trình giải quyết yêu cầu số 2:

```
C:=''; For i:=1 to n do C:=C+'0'; //Tạo xâu C gồm N bit 0
For i:=1 to n do if K>F[n-i] then
  Begin
    C[i] :='1';
    K:=K-F[n-i];
  End;
```

Bây giờ sau khi đọc xong phần trên, các bạn sau khi đã hiểu hãy bật Pascal lên, soạn và chạy chương trình hoàn chỉnh cho bài Nhiphan1.pas. Sau khi chạy được bài này các bạn sẽ hiểu thêm về ý tưởng Quy hoạch động vị trí cấu hình để chuẩn bị cho bài số 2 sau đây.

2. Nhphan2.pas

Một tập S gồm các dãy nhị phân N bit 0 và 1 trong đó không có hai bit 1 nào liền nhau. Ví dụ với N=5 thì S gồm các dãy 00000, 00001, 00101, ... Tập S được sắp xếp theo chiều tăng dần của số nguyên tương ứng mà dãy bit biểu diễn. Cho một số nguyên N ($n < 100$) hãy cho biết:

- Dãy bit thứ M trong dãy S là bao nhiêu?
- Dãy bit R là có số thứ tự là bao nhiêu trong tập S.

Nhphan2.inp	Nhphan2.out
5	000101
3	2
00001	

*** Phần chung:**

Ta xây dựng mảng F với ý nghĩa;

$F[i]$ là số lượng dãy nhị phân có độ dài i bit (Ko có 2 bit 1 liền nhau)

Ta có:

$$F[0] = 1$$

$$F[1] = 2 \quad \text{Gồm: } 0, 1$$

$$F[2] = 3 \quad \text{Gồm: } 00, 01, 10$$

$$F[3] = 5 \quad \text{Gồm: } 000, 001, 010, 100, 101$$

$$F[4] = 8 \quad \text{Gồm: } 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010$$

...

$$\text{Vậy } F[i] = F[i-1] + F[i-2] \quad \text{Với } i=2..n$$

Từ đây ta có đoạn chương trình quy hoạch động tính F như sau:

$F[0] := 1; F[1] := 2;$

For $i := 2$ to n do $F[i] := F[i-1] + F[i-2];$

* Ta nhận thấy yêu cầu 1 và yêu cầu 2 tương tự như bài số 1.

3. Số hiệu hoán vị: SHHV.pas

Xét tập A gồm tất cả các hoán vị của dãy số tự nhiên $(1, 2, \dots, n)$ ($n \leq 20$), các hoán vị được sắp xếp theo thứ tự từ điển.

Yêu cầu:

- Cho trước 1 hoán vị: Tìm số hiệu của hoán vị đó trong tập A
- Cho trước 1 số hiệu: Tìm hoán vị tương ứng.

Dữ liệu vào:

- Dòng 1: Dãy hoán vị S
- Dòng 2: Chứa số hiệu K

Kết quả: Lưu vào SHHV.out

- Dòng 1: q (là số hiệu của hoán vị S)
- Dòng 2: Dãy hoán vị tương ứng với số hiệu K

SHHV.inp	SHHV.out
213	2
4	231

*** Phần yêu cầu chung:**

Ta xây dựng mảng F với ý nghĩa:

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

Ta có:

$F[1] = 1$ Gồm: 1
 $F[2] = 2$ Gồm: 12, 21
 $F[3] = 6$ Gồm: 123, 132, 213, 231, 312, 321
 $F[4] = 24$ Gồm: 1234, 1243, 1324, 1342, 1423, 1432... 4321
 ...
 $F[i] = i!$ hay $F[i-1] * i$

*** Yêu cầu 1: Số hiệu của hoán vị**

xét ví dụ

$N = 4$ a: 3 4 1 2

Ta đi tìm số hoán vị có thứ tự từ điển nhỏ hơn dãy a là vt.

- $i = 1$: Tìm số hoán vị có phần tử đầu tiên $< a[1]$ (các số nhỏ hơn $a[1]$ gồm 1 và 2), cứ mỗi phần tử x nhỏ hơn này sẽ có $F[n - i]$ dãy hoán vị có phần tử đầu tiên bằng x và thứ tự từ điển của chúng luôn nhỏ hơn dãy a chẳng hạn với $x=1$ có: các dãy sau sẽ có số thứ tự nhỏ hơn a:

1 2 3 4, 1 2 4 3, 1 3 2 4, 1 3 4 2, 1 4 2 3, 1 4 3 2.

Vậy $vt := vt + 2 * F[n - 1] = 12$.

- $i = 2$: Ta đi tìm số hoán vị có phần tử đầu tiên = 3 mà có thứ tự từ điển nhỏ hơn a. Những phần tử này phải có phần tử thứ 2 nhỏ hơn $a[2]$ và cứ mỗi phần tử sẽ có $F[n - 2]$ hoán vị thỏa mãn. Ở đây có giá trị 1 và 2 thỏa mãn $vt := vt + 2 * F[n - 2] = 16$;
- $i=3$: Ta tìm số lượng các hoán vị có phần tử đầu = 3, phần tử thứ 2 = 4 mà có thứ tự từ điển nhỏ hơn a. Những hoán vị này phải có phần tử thứ 3 nhỏ hơn 1 → Không có phần tử nào nhỏ hơn nữa.
- $i=4$: Ta không cần xét đến vị trí này vì đến đây thì $i=n$ tức là $a[i]$ là số cuối cùng rồi, ko có số nào nhỏ hơn nữa

Vậy kết quả sẽ là $(vt + 1) = 17$; tức hoán vị 3 4 1 2 có số hiệu là 17.

Vậy ta có đoạn chương trình thực hiện yêu cầu 1:

```

Vt:=0;
For i:=1 to n-1 do vt:=vt + Find(i) * F[n-i];

```

Ở đây Find(i) là hàm trả về số lượng các số trong đoạn $a_{i+1}..a_n$ và có giá trị bé hơn a_i .

```

Function Find(i:longint):longint;
  Var F: array[1..101] of boolean;  j,d: longint;
  Begin
    For j:=1 to i do F[a[j]]:=true; {Danh dau cac phan tu da dung tu 1..i}
    d:=0;                          {Dem so luong cac so chua dung nho hon ai}
    For j:=1 to n do
      if (not f[j]) and (j<a[i]) then inc(d);
    Exit(d);
  End;

```

* Yêu cầu 2: Tìm dãy hoán vị tương ứng với số hiệu K.

Gọi dãy hoán vị là $C = C_1, C_2 \dots C_n$

Để tìm dãy hoán vị ứng với số hiệu K ta sẽ tìm lần tìm các số thích hợp cho C_1 , rồi $C_2 \dots$ cho đến C_i .

```

Procedure Yeucan2;
  Var i,j:longint; Free:array[1..100] of boolean;
  Begin
    Fillchar(c,sizeof(c),0);
    Fillchar(free,sizeof(free),true);
    For i:=1 to n do
      Begin
        For j:=1 to n do
          if free[j] then
            if (k>F[n-i]) then k:=k-F[n-i] else break;
          c[i]:=j; free[j]:=false;
        End;
      End;
  End;

```

II. BÀI TẬP TỰ LUYỆN

2.1 Số hiệu tổ hợp:

Cho tập hợp A gồm N phần tử. Mỗi tập con gồm K ($1 \leq K \leq N$) phần tử của A được gọi là một tổ hợp chập K của N phần tử đã cho

Bài toán đặt ra là:

- Cho số hiệu của một tổ hợp chập K của N số nguyên dương đầu tiên, hãy tìm tổ hợp chập đó.
- Cho tổ hợp chập K của N số nguyên dương đầu tiên, hãy tính số hiệu của tổ hợp chập đó.

Input

Gồm 3 dòng có dạng như sau:

- Dòng 1: Ghi 2 số nguyên N, K ($3 \leq N \leq 300$)
- Dòng 2: Ghi số nguyên S
- Dòng 3: Gồm K số nguyên B_1, B_2, \dots, B_K ($B_1 < B_2 < \dots < B_K$)

Output

- Dòng 1: Ghi ra dãy số A_1, A_2, \dots, A_K là tổ hợp chập K của N số nguyên dương đầu tiên có số hiệu S. Các số viết theo thứ tự tăng dần.
- Dòng 2: Ghi số hiệu của tổ hợp chập K: B_1, B_2, \dots, B_K .

Ví dụ:

SHTH.inp	SHTH.out
3 2	1 3
2	3
2 3	

2.2 Liên lạc vũ trụ:

Để liên lạc với tàu thăm dò tự động người ta chuẩn bị danh sách các thông báo, đánh số từ 1 trở đi và cài vào bộ nhớ trong của máy tính trên trạm thăm dò. Số lượng thông báo là 10^5 . Trạm điều khiển

mặt đất hoặc tàu thăm dò chỉ cần phát đi số thứ tự thay vì cho chuyển bằng hệ thống phát xung laser định hướng. Nhưng việc phát xung cần phải chuyển các số thứ tự này thành dãy nhị phân trước khi phát đi.

Em hãy lập trình cài vào máy phát việc chuyển đổi từ giá trị số thứ tự sang xâu nhị bit tương ứng cần phát đi.

Dữ liệu vào: **impulse.inp**

- Dòng 1: Số lượng thông báo cần phát đi R (10^5)
- Các dòng sau: Chứa các số nguyên dương (Số thứ tự thông báo), các số cách nhau ít nhất một dấu cách.

Kết quả:

- Mỗi dòng là một xâu bit cần phát. Bỏ qua các số 0 trước 1 đầu tiên trong xâu, trừ trường hợp số cần phát là 1 thì kết quả được ghi là 0.

Ví dụ:

impulse.inp	impulse.out
6	0
1 2 3 4 5	1
100	10
	11
	100
	100000110

2.3 Đánh số tập con:

Giả sử A là tập N số nguyên dương đầu tiên ($N \leq 30$). Với mỗi tập con B của A ta luôn viết các phần tử của B theo thứ tự tăng dần.

Ta xếp thứ tự các tập con này của A theo nguyên tắc: giả sử B và C là hai tập con, nếu có i sao cho $B_i < C_i$. Trên cơ sở các tập con này ta đánh số các tập con từ 1 đến 2^N , tập rỗng được đánh số 1.

Ví dụ: Với $n = 3$ ta có

Thứ tự	Tập con
1	
2	1
3	1 2
4	1 2 3
5	1 3
6	2
7	2 3
8	3

Yêu cầu:

- Cho một số hiệu $S > 1$, tìm tập B có số hiệu này?
- Cho tập C không rỗng, tìm số hiệu của tập C

Dữ liệu vào: **Subset.inp**

- Dòng 1: Ghi N
- Dòng 2: S
- Dòng 3: Tập C

Kết quả: **Subset.out**

- Dòng 1: Tập B ứng với số hiệu S
- Dòng 2: Dãy số C

Subset.inp	Subset.out
3	3
8	7
2 3	

2.4 Chuỗi hạt Nlace

Khi tiến hành khai quật khảo cổ một vương quốc nọ, các nhà khoa học khai quật được rất nhiều chuỗi hạt lạ. Sau khi quan sát các nhà khoa học thấy rằng các chuỗi hạt này có một số đặc điểm chung.

Mỗi chuỗi hạt là một sợi dây được đính các hạt ngọc làm bằng chất liệu cổ xưa. Các chuỗi hạt đều có số lượng hạt ngọc bằng nhau. Hơn nữa mỗi hạt ngọc là một hình cầu có đường kính là một số nguyên dương. Nếu lần từ trái sang phải ta thấy đường kính các hạt ngọc tăng dần. Nếu đánh số vị trí các hạt ngọc bắt đầu từ 1, theo thứ tự từ trái sang phải người ta nhận thấy rằng hạt ngọc thứ i có đường kính không vượt quá $2i$. Các nhà khoa học cho rằng dân tộc cổ xưa này hẳn đã làm ra tất cả các chuỗi hạt có cùng những đặc điểm này.

Sau đó không lâu, các nhà khoa học tìm ra một mảnh da trên đó có ghi một con số kỳ lạ theo chữ số cổ xưa. Họ cho rằng mảnh da này có liên quan đến các chuỗi hạt. Sau nhiều cố gắng các nhà khoa học đã đưa được con số trên mảnh da về hệ thập phân và ký hiệu X.

Các nhà khoa học nhận định hãy xác định chuỗi hạt có thứ tự từ điển là X, biết đâu đây sẽ là manh mối về mối liên hệ với các chuỗi hạt.

Bạn hãy lập trình giúp các nhà khoa học tìm chuỗi hạt thứ X

Dữ liệu vào: Nlace.inp

- Dòng 1: N là số ngọc trong mỗi chuỗi hạt
- Dòng 2: X

Kết quả: Nlace.out

- Là tập đường kính của các hạt trong chuỗi hạt thứ X.

Ví dụ:

Nlace.inp	Nlace.out	Giải thích
2	2 3	Các chuỗi hạt là: 1 2, 1 3, 1 4, 2
4		3, 2 4.