THIẾT KẾ THUẬT TOÁN 5

Click to add Text

Một số kỹ thuật thiết kế

1. Kỹ thuật thiết kế tuần tự
2. Kỹ thuật thiết kế theo cấu trúc
3. Kỹ thuật đệ quy
4. Kỹ thuật tham lam
5. Kỹ thuật chia để trị
6. Kỹ thuật liệt kê
7. Lựa chọn cấu trúc dữ liệu thích hợp
8. Bài tập

6. Kỹ thuật liệt kê

Bài toán:

Cho tập hợp X. Hãy liệt kê toàn bộ các phần tử của tập hợp.

6. Kỹ thuật liệt kê

Kỹ thuật liệt kê các phần tử của một tập hợp có thể dựa vào các yếu tố chính:

- **Lựa chọn một thứ tự có sẵn phù hợp với mô tả toàn bộ các phần tử của tập hợp.**
- **Xây dựng nguyên tắc sinh một phần tử từ một phần tử sẵn có.**
- **Xây dựng một thứ tự mới nhằm mô tả trật tự sẵn có giữa các phần tử trong tập hợp.**

6. Kỹ thuật liệt kê

Khó khăn của kỹ thuật liệt kê

- Tìm ra thứ tự liệt kê
- Không để lặp lại và
- Không bỏ sót phần tử.

6. Kỹ thuật liệt kê

Ví dụ 6.1. Liệt kê các dãy nhị phân có độ dài n.

Nhận xét:

1) **Số dãy nhị phân độ dài *n* là 2n.**
2) **Tồn tại song ánh (toàn ánh, đơn ánh) từ tập**

   $P = \{0, 1, .., 2^n\text{-}1\}$

   vào tập các dãy nhị phân độ dài *n*,

   $Q = \{ (b_1, b_2, ..., b_n) \}$

   mỗi số nguyên $m \in P$,

   $m \rightarrow b = (b_1, b_2, ..., b_n)$,

   sao cho

   $m = 2^0.b_1 + 2^1.b_2 + ...2^{i-1}.b_i + ... + 2^{n-1}.b_n$

   b còn gọi là biểu diễn nhị phân của m.

6. Kỹ thuật liệt kê

Nhận xét:

3) **Phép cộng nhị phân**

$$(b_1,\ b_2, \ldots,\ b_n)$$

$$+ \quad 1$$

$$= \quad (c_1, c_2, \ldots, c_n)$$

Giả sử b có dạng

$$b = (1,1,\ldots,1,\ 0,\ b_{k+1}, \ldots, b_n)$$

thì c có dạng

$$c = (0,0,\ldots,0,\ 1,\ b_{k+1}, \ldots b_n)$$

6. Kỹ thuật liệt kê

Nhận xét:

4) **Định nghĩa quan hệ lớn hơn, nhỏ hơn trên các dãy nhị phân**

$$\mathbf{b = (b_1, b_2, ..., b_n) \quad < \quad a = (a_1, a_2, ..., a_n)}$$

nếu $\exists$ k $\in$ {1, 2, ..., *n*}

$$b_k < a_k, \qquad b_{k+1}....b_n \equiv a_{k+1}.....a_n$$

b < a và không tồn tại một dãy nhị phân c độ dài *n* sao cho b<c<a

nếu như: $b_k < a_k, \qquad b_{k+1}....b_n \equiv a_{k+1}.....a_n$

và $a_i = 0$ với $i = 1, ..., k-1$.

6. Kỹ thuật liệt kê

Thuật toán 1:

input: n

output: $(b_1, b_2, \ldots, b_n)$ - các dãy nhị nhân độ dài n

```
1)  K = 2^n-1;
2)  for (x = 0; x ≤ K; x++)
        {  a)  for ( i=1; i ≤ n;  i++ )
                       b[i]  =  x %2;
                       x     =  x / 2;
           b)  output b;
        }
```

6. Kỹ thuật liệt kê

Thuật toán 1:

input: n

output: $(b_1, b_2, ..., b_n)$ - các dãy nhị nhân độ dài n

1) K = 2n-1;

2) for (x = 0; x ≤ K; x++)

```
{   a)   for ( i=1; i ≤ n;  i++ )
                 b[i]  =  x %2;
                 x     =  x / 2;
     b)   output b;
}
```

<u>**Khó khăn**</u>:

Chương trình hóa thuật toán khi *n* là số lớn sẽ khá khó khăn vì khi này K sẽ là số lớn.

Tổ chức vòng lặp với số lớn cần có kỹ thuật cài đặt riêng và phức tạp.

6. Kỹ thuật liệt kê

Thuật toán 2:

input: n

output: (b_1, b_2, ..., b_n) - các dãy nhị nhân độ dài n

1) for (i=1; $i \le n$; i++) b[i] = 0;
   output b;
2) while (!stop)
   a) k = 1; while (b[k]>0) k++;
   b) for (i = 1; i ≤ k-1; i++) b[i] = 0;
   c) b[k] = 1;
   d) output b;

<u>Bài tập</u>: Thiết kế chi tiết dừng của thuật toán khi liệt kê hết các dãy nhị phân độ dài n.

6. Kỹ thuật liệt kê

Thuật toán 2:

input: n

output: (b_1, b_2, ..., b_n) - các dãy nhị nhân độ dài n

1) for (i=1; $i \leq n$; i++) b[i] = 0;
   output b;
2) while (!stop)
   a) k = 1; while (b[k]>0) k++;
   b) for (i = 1; i ≤ k-1; i++) b[i] = 0;
   c) b[k] = 1;
   d) output b;

a) k = 1; while (b[k]>0)
{ b[k] = 0; k++; }
c) b[k] = 1;
d) output b;

6. Kỹ thuật liệt kê

Thuật toán 3:

input: n

output: $(b_1, b_2, ..., b_n)$ - các dãy nhị nhân độ dài n

1) for (i=1; $i \le n$; i++) b[i] = 1;

   output b;

2) while (!stop)

   a) k = 1; while (b[k] < 1) k++;

   b) for (i = 1; i ≤ k-1; i++) b[i] = 1;

   c) b[k] = 0;

   d) output b;

a) k = 1; while (b[k]<1)

{ b[k] = 1; k++; }

c) b[k] = 0;

d) output b;

6. Kỹ thuật liệt kê

Ví dụ 6.2:

Liệt kê các hoán vị của tập $X^n = \{1, 2, ..., n\}$

Các hoán vị của dãy X^4 theo thứ tự từ điển ngược

1 2 3 4	1 2 4 3	1 3 4 2	2 3 4 1
2 1 3 4	2 1 4 3	3 1 4 2	3 2 4 1
1 3 2 4	1 4 2 3	1 4 3 2	2 4 3 1
3 1 2 4	4 1 2 3	4 1 3 2	4 2 3 1
2 3 1 4	2 4 1 3	3 4 1 2	3 4 2 1
3 2 1 4	4 2 1 3	4 3 1 2	4 3 2 1

6. Kỹ thuật liệt kê

1 2 3 4	1 2 4 3	1 3 4 2	2 3 4 1
2 1 3 4	2 1 4 3	3 1 4 2	3 2 4 1
1 3 2 4	1 4 2 3	1 4 3 2	2 4 3 1
3 1 2 4	4 1 2 3	4 1 3 2	4 2 3 1
2 3 1 4	2 4 1 3	3 4 1 2	3 4 2 1
3 2 1 4	4 2 1 3	4 3 1 2	4 3 2 1

Nhận xét:

X^4 (để đơn giản, kí hiệu là X) được chia ra thành *4* nhóm, nhóm *4*, 3, 2 và 1

Trong mỗi nhóm, bỏ đi phần tử sau cùng thì là hoán vị của n-1 phân tử

Chuyển từ nhóm k sang nhóm k-1: 3214 -> 1243

đổi chỗ X_k với X_n: 4213

sắp xếp tăng dần từ $X_1,\ldots, X_{n-1}$: 1243

bắt đầu sinh các hoán vị của $X_1 \ldots X_{n-1}$

6. Kỹ thuật liệt kê

Nhận xét:

đổi chỗ X_k với X_n: 4213

sắp xếp tăng dần từ $X_1,..., X_{n-1}$: 1243

bắt đầu sinh các hoán vị của $X_1...X_{n-1}$

1) initial permute $(X_1, X_2, ..., X_n)$
2) nhóm k = 1
3) sinh các hoán vị $(X_1, X_2, ..., X_{n-1})$
4) chuẩn bị cho nhóm tiếp theo:
   a) swap (X_k, X_n);
   b) sort on $\{X_1, X_2, ..., X_{n-1}\}$ (increate)
5) k++;
6) goto 3)

6. Kỹ thuật liệt kê

Nhận xét:

đổi chỗ X_k với X_n: 4213

sắp xếp tăng dần từ $X_1,..., X_{n-1}$: 1243

bắt đầu sinh các hoán vị của $X_1...X_{n-1}$

1) initial permute $(X_1, X_2, ..., X_n)$
2) nhóm k = 1
3) sinh các hoán vị $(X_1, X_2, ..., X_{n-1})$
4) chuẩn bị cho nhóm tiếp theo:
   a) swap (X_k, X_n);
   b) sort on $\{X_1, X_2, ..., X_{n-1}\}$ (increate)
5) k++;
6) goto 3)

hoặc:

a) sort on $\{X_1, X_2, ..., X_{n-1}\}$

b) swap (X_{n-k}, X_n);

nhóm 1: 321 4 -> 123 4 -> 124 3

nhóm 2: 421 3 -> 124 3 -> 134 2

nhóm 3: 431 2 -> 134 2 -> 234 1

6. Kỹ thuật liệt kê

Nhận xét:

đổi chỗ X_k với X_n: 4213

sắp xếp tăng dần từ $X_1, \ldots, X_{n-1}$: 1243

bắt đầu sinh các hoán vị của $X_1 \ldots X_{n-1}$

1) initial permute $(X_1, X_2, \ldots, X_n)$
2) nhóm k = 1
3) sinh các hoán vị $(X_1, X_2, \ldots, X_{n-1})$
4) chuẩn bị cho nhóm tiếp theo:
   a) sort on $\{X_1, X_2, \ldots, X_{n-1}\}$
   b) swap (X_{n-k}, X_n);
5) k++;
6) goto 3)

Nhận xét:

Do ở hoán vị cuối cùng có thứ tự ngược đối với $\{X_1, X_2, \ldots, X_{n-1}\}$

nên việc sắp xếp chỉ cần gập đôi để đổi chỗ

for (j = 1; j ≤ m/2; j++)
swap(X_j, X_{m-j+1})

hoặc

for (j = 1; j ≤ n/2; j++)
swap(X_j, X_{n-j})

6. Kỹ thuật liệt kê

```
void  LK (int  m)
{
        if  (m = =1)  Viết hoán vị;
        else
                for  (i=1; i≤m; i++)
                     {       LK( m-1);
                             if  ( i<m)
                                         {  swap( X[i], X[m] );
                                           sort on X[1],...., X[m-1];
                                         }
                     }
}
```

6. Kỹ thuật liệt kê

```
void  LK (int  m)
{
        if  (m = =1)  Viết hoán vị;
        else
                for  (i=1; i≤m; i++)
                     {        LK( m-1);
                              if  ( i<m)
                                          {  swap( X[i], X[m] );
                                             for ( j = 1; j ≤ m/2; j++)  swap( X[j], X[m-j]);
                                          }
                     }
}
```

6. Kỹ thuật liệt kê

Ví dụ 6.3. Liệt kê phân hoạch một tập hợp.

Giải thích:

Ví dụ, X = {1, 2, 3} có 5 phân hoạch:

{1}, {2}, {3}

{1, 2}, {3}

{1, 3}, {2}

{1}, {2, 3}

{1, 2, 3}

6. Kỹ thuật liệt kê

Ví dụ 6.3. Liệt kê phân hoạch một tập hợp.

Giải thích:

Ví dụ, X = {1, 2, 3} có 5 phân hoạch:

{1}, {2}, {3}

{1, 2}, {3}

{1, 3}, {2}

{1}, {2, 3}

{1, 2, 3}

6. Kỹ thuật liệt kê

Ví dụ, X = {1, 2, 3, 4} có 15 phân hoạch:

{1}, {2}, {3}, {4}
{1, 2}, {3}, {4}
{1, 3}, {2}, {4}
{1}, {2, 3}, {4}
{1, 2, 3}, {4}
{1, 4}, {2}, {3}
{1}, {3}, { 2, 4}
{1}, {2}, { 3, 4}
{1, 2, 4}, { 3}
{1, 2}, {3, 4}
{2}, {1, 3, 4}
{1, 3}, {2, 4}
{1, 4}, {2, 3}
{1}, { 2, 3, 4}
{1, 2, 3, 4}

6. Kỹ thuật liệt kê

Phân tích:

$X^1 = \{1\}$

có 1 phân hoạch

{1}

Các phân hoạch $X^2 = \{1, 2\}$

sinh ra bằng cách tạo 2 nhóm:

1) Thêm tập {2} vào các phân hoạch X^1

2) Thêm phần tử (2) vào các tập của từng phân hoạch X^1

Nhóm 1:

{1}, {2}

Nhóm 2:

{1, 2}

Phân hoạch của X^2:

{1}, {2}

{1, 2}

Các phân hoạch của X^3:

Nhóm 1:

{1}, {2} {3}

{1, 2} {3}

Nhóm 2:

{1}, {2, 3}

{1, 3}, {2}

{1, 2, 3}

6. Kỹ thuật liệt kê

Các phân hoạch của X^3:

{1} {2} {3}

{1, 2} {3}

{1} {2, 3}

{1, 3} {2}

{1, 2, 3}

Phân hoạch của X^4:

Nhóm 1:

{1} {2} {3} {4}

{1, 2} {3} {4}

{1} {2, 3} {4}

{1, 3} {2} {4}

{1, 2, 3} {4}

Nhóm 2:

{1} {2} {3, 4}

{1} {2, 4} {3}

{1, 4} {2} {3}

{1, 2} {3, 4}

{1, 2, 4} {3}

{1} {2, 3, 4}

{1, 4} {2, 3}

{1, 3} {2, 4}

{1, 3, 4} {2}

{1, 2, 3, 4}

6. Kỹ thuật liệt kê

Quản lý phân hoạch bằng mảng
b = (b_1, b_2, ..., b_n)
b_i = k nghĩa là X_i thuộc tập con k.
Dùng chỉ số tối thiểu để quản lý chỉ số tập con.

Các phân hoạch của X^3:

{1} {2} {3}
{1, 2} {3}
{1} {2, 3}
{1, 3} {2}
{1, 2, 3}

Chỉ sô quản lý:
(1, 2, 3)
(1, 1, 2)
(1, 2, 2)
(1, 2, 1)
(1, 1, 1)

Phân hoạch của X^4:
Từ một phân hoạch của X^3 là b = (b_1, b_2, b_3)
ví dụ
(1, 2, 2)
sinh các phân hoạch của X^4:
(1,2, 2, 3) – X^4_4 đứng riêng trong một tập con (mới)
(1,2, 2, 2) – X^4_4 đứng trong tập con sô 2 (đã có từ trước)
(1,2, 2, 1) – X^4_4 đứng trong tập con sô 1 (đã có từ trước)

6. Kỹ thuật liệt kê

Quản lý phân hoạch bằng mảng
$b = (b_1, b_2, ..., b_n)$
$b_i = k$ nghĩa là X_i thuộc tập con k.
Dùng chỉ số tối thiểu để quản lý chỉ số tập con.

Các phân hoạch của X^3:

{1} {2} {3}
{1, 2} {3}
{1} {2, 3}
{1, 3} {2}
{1, 2, 3}

Chỉ sô quản lý:
(1, 2, 3)
(1, 1, 2)
(1, 2, 2)
(1, 2, 1)
(1, 1, 1)

Từ một phân hoạch của X^{n-1} là $b = (b_1, b_2, ..., b_{n-1})$
sinh các phân hoạch "tương ứng" của $X = X^n$:

def: $k = \max \{b_1, b_2, ..., b_{n-1}\}$
$b_n = k+1$ - X_n đứng riêng trong một tập con (tập con mới)
$b_n = k$ X_n đứng trong tập con sô k (đã có sẵn)
$b_n = k-1$ X_n đứng trong tập con sô k-1 (đã có sẵn)
...
$b_n = 1$ X_n đứng trong tập con sô k-1 (đã có sẵn)
chẳng hạn:
từ (1, 2, 3)
có một số phân hoạch của X^4:
k = 3
(1, 2, 3, 4)
(1, 2, 3 ,3)
(1, 2, 3 ,2)
(1, 2, 3 ,1)

Bằng cách này ta có tất cả phân hoạch của X^4.

6. Kỹ thuật liệt kê

Câu hỏi: Nếu theo cách này sẽ phải lưu danh sách các phân hoạch.

Rất tốn kém bộ nhớ.

Từ một phân hoạch

$$b = (b_1, b_2, \ldots, b_{n-1}, b_n)$$

làm thế nào để sinh phân hoạch tiếp theo?

6. Kỹ thuật liệt kê

Nhận xét:

Trong $b = (b_1, b_2, ..., b_m, b_{m+1}, ..., b_n)$
nếu biết $(b_1, b_2, ..., b_m)$

có thể sinh phân hoạch tất cả các phân hoạch tiếp theo?

6. Kỹ thuật liệt kê

Sử dụng kỹ thuật đệ quy:

void Liệt kê (int m)

điểm dừng: m == n
thực hiện: in phân hoạch
các bài toán con:
$k = \max(b_1, b_2, ..., b_m) + 1$
$b_{m+1} = k$; Liệt kê(m+1)
$b_{m+1} = k-1$; Liệt kê(m+1)
....
$b_{m+1} = 1$; Liệt kê(m+1)

6. Kỹ thuật liệt kê

```
void Liệt kê (int m)
  {
      if (m == n)  output (b1, b2, ..., bn)
      else
      {
            k = max (b1, b2, ..., bm) + 1
            for (i = k; i ≥ 1; i--)
          {      bm+1 = i;
                 Liệt kê( m+1);
          }
      }
  }
```

6. Kỹ thuật liệt kê

Phân tích 2:

Xét phân hoạch:

$(b_1, b_2,..., b_n)$ (1)

Chỉ ra phân hoạch đứng ngay sau (1).

Ký hiệu

$(c_1, c_2, ..., c_n)$ (2)

là phân hoạch đứng ngay sau (1)

Ký hiệu k là chỉ số lớn nhất có $b_k > 1$:

$(b_1, ..., b_{k-1}, b_k, 1, ..., 1)$

Đặt $x = \max(b_1, b_2, ..., b_{k-1}, b_k - 1)$

(2) có dạng

$(b_1, b_2, ..., b_{k-1}, x+1, x+2, ..., x+n-k)$

(bài tập: tự chứng minh)

6. Kỹ thuật liệt kê

Phân tích 2:

input b = $(b_1, b_2, ..., b_n)$

output c = $(c_1, c_2, ..., c_n)$ ngay sau b

1) k = n;
2) while ((k ≥1) & (b[k] = = 1)) k--;
3) for (i = 1; i ≤ k-1; i++) c[i] = b[i];
4) c[k] = b[k] – 1;
5) x = max { c[1], c[2], .., c[k]}
6) for (i =k+1; i ≤ n; i++) c[i] = $x + i - k$;

6. Kỹ thuật liệt kê

Ví dụ 6.4. Liệt kê phân hoạch của một số nguyên dương

Cho số nguyên dương *n*. Liệt kê tất cả các phân tích *n* thành tổng các số nguyên dương không vượt quá *k* (không tính các hoán vị số hạng).

chẳng hạn, $n = 5$, $k = 3$

có 5 phân hoạch:

5 = 3 + 2

5 = 3 + 1 + 1

5 = 2 + 2 + 1

5 = 2 + 1 + 1 + 1

5 = 1 + 1 + 1+ 1 + 1

input n, k

output các phân hoạch

6. Kỹ thuật liệt kê

Nhận xét:

- Sắp các số hạng theo thứ tự không tăng.
- Kí hiệu q = n/k và p = n mod k.

Phân hoạch đầu tiên là

$b_1 + ... + b_q + b_{q+1}$

với $b_1 = ... = b_q = k$

và $b_{q+1} = p$.

chẳng hạn, n = 5, k = 3 thì q = 1, p = 2

phân hoạch đầu tiên là 3 + 2

chẳng hạn, n = 7, k = 3 thì q = 2, p = 1

phân hoạch đầu tiên là 3 + 3 + 1

6. Kỹ thuật liệt kê

<u>Nhận xét</u> (xuất phát từ thứ tự không tăng của các số hạng)

Xét phân hoạch

$$b_1 + \ldots + b_{k-1} + b_k + 1 + \ldots + 1$$

với $b_k > 1$

Kí hiệu

***x* là tổng các số 1 đứng sau cùng,**

$a = b_k - 1$

$m = (x+1)/a$ và

$t = (x+1) \bmod a$

Phân hoạch tiếp theo sẽ là

$$b_1 + \ldots + b_{k-1} + a + \ldots + a + t$$

trong đó có *m* số hạng a.

6. Kỹ thuật liệt kê

<u>Nhận xét</u> (xuất phát từ thứ tự không tăng của các số hạng)

Phân hoạch sau cùng

$$1 + \ldots + 1 + 1 + 1 + \ldots + 1$$

với n số hạng bằng 1

6. Kỹ thuật liệt kê

1) Phân hoạch đầu tiên:

```
for (i=1;i ≤ n/k; i++) b[i]= k; b[i] = n %k;
in PH (b[1],...,b[i]);
```

2) Các phân hoạch tiếp theo

```
while (b[1]>1)
    {   ts = 0; i = 1; while (b[i]>1) {ts = ts + b[i];i++}
        i--;
        tcl = n – ts + b[i];
        a = b[i] – 1;
        for ( t= 1; t ≤ tcl/a; t++)  b[i+t-1] = a;
        i = i + tcl/a -1;
        if (tcl%a>0)
                { i++;    b[i] = tcl%a; }
        in PH (b[1],...,b[i]);
    }
```

6. Kỹ thuật liệt kê

Nhận xét:

- **Sắp các số hạng theo thứ tự không tăng.**
- **Kí hiệu q = n/k và p = n mod k. Phân hoạch đầu tiên là**

  $b_1 + \ldots + b_q + b_{q+1}$

  với $b_1 = \ldots = b_q = k$

  và $b_{q+1} = p$.
- **Xét phân hoạch b1+ ...+ bk-1+bk + 1 ... + 1, với bk>1. Kí hiệu *x* là tổng các số 1 đứng sau cùng, *a* = bk-1, *m* = (*x*+1)/a và t = (*x*+1) mod a. Phân hoạch tiếp theo sẽ là**

  $b_1 + \ldots + b_{k-1} + a + \ldots + a + t$

  trong đó có *m* số hạng a.

Do phân hoạch sau cùng chỉ gồm các số hạng bằng 1 nên quá trình phân tích sẽ dừng khi $b_1 = 1$.

Một số kỹ thuật thiết kế

1. **Kỹ thuật thiết kế tuần tự**
2. **Kỹ thuật thiết kế theo cấu trúc**
3. **Kỹ thuật đệ quy**
4. **Kỹ thuật tham lam**
5. **Kỹ thuật chia để trị**
6. **Kỹ thuật liệt kê**
7. **Lựa chọn cấu trúc dữ liệu thích hợp**
8. **Bài tập**

7. Lựa chọn cấu trúc dữ liệu thích hợp

Ưu điểm:

- Tìm được cách thiết kế đơn giản, mạch lạc hơn;
- Tính đơn giản cộng với cấu trúc dữ liệu rõ ràng làm cho việc kiểm soát thuật toán (tính đúng đắn, tính dừng,...) trở lên dễ dàng hơn.
- Cài đặt thuật toán thuận tiện hơn.

7. Lựa chọn cấu trúc dữ liệu thích hợp

Ví dụ 7.1: **Xác định tên theo âm lịch của một năm.**

Tên một năm âm lịch được cấu thành từ hai yếu tố là can và chi. Có tất cả 10 can

(Giáp, Ất, Bính, Đinh, Mậu, Kỷ, Canh, Tân, Nhâm, Quý)

và 12 chi

(Tí, Sửu, Dần, Mão, Thìn, Tỵ, Ngọ, Mùi, Thân, Dậu, Tuất, Hợi),

sử dụng theo lối quay vòng. Như vậy, tên năm âm lịch sau chu kỳ 60 năm sẽ lặp lại.

Ví dụ, năm 1945 là Ất Dậu thì 2005 cũng là Ất Dậu.

Giả thiết là không có đổi lịch thì có thể nhẩm được năm 0 (hay năm 1 trước công nguyên) là Canh Thân. Fs

7. Lựa chọn cấu trúc dữ liệu thích hợp

Ví dụ 7.1: Xác định tên theo âm lịch của một năm.

Nếu tổ chức dữ liệu thành 2 mảng **can**, **chi**

```
char *can[10] =
   {"Canh","Tân","Nhâm","Quý","Giáp",“Ất","Bính",“Đinh","Mậu","Kỷ"};
char *chi[12] =
   {"Thân","Dậu","Tuất","Hợi","Tí","Sửu","Dần","Mão","Thìn","Tỵ","Ngọ","Mùi"};
```

năm dương lịch *n* bất kỳ, có tên âm lịch tương ứng là

can[n mod 10] + chi[n mod 12]

input   n;

output   can[n mod 10] + chi[n mod 12]

7. Lựa chọn cấu trúc dữ liệu thích hợp

Ví dụ 7.2: **Cho dãy số A = {A_1, A_2, ..., A_n}.**

Tìm dãy con liên tiếp S có tổng dương nhỏ nhất.

minh họa: **cho dãy A:**

{ 2, 3, -4, 3, 4, -10, 8, -2, 7, -12, -2, 1}

dãy S

{3, 4, -10, 8, -2, 7, -12, -2, 1}

có tổng bằng 1

7. Lựa chọn cấu trúc dữ liệu thích hợp

input $A = \{A_1, A_2, ..., A_n\}$

output $S = \{A_{io}, A_{io+1}, ..., A_{jo-1}, A_{jo}\}$ có tổng dương nhỏ nhất

```
1) while (A[i]<0) i++;
2) if (i>n)  output {S = (); pos_sum = -1};
else
    {  io = 1; jo = io; sum = A[io];
       for ( i= 1; i ≤ n; i++)
        {   ts = A[i];
            for ( j= i+1; j ≤ n; j++)
                {  ts = ts + A[j];
                   if ((ts>0) & (ts<sum))
                         { sum = ts; io = i; jo = j};
                }
        }
      output  io; jo;
    }
```

8. Bài tập

<u>Câu hỏi ôn tập và bài tập tự làm</u>:

1) Xây dựng thuật toán liệt kê dãy nhị phân độ dài *n* theo từ điển ngược.
2) Liệt kê mọi phương án có thể để tô màu đồ thị G = {V, E}, với các đỉnh được đánh số từ 1 đến *n* và các cạnh được cho bởi danh sách cạnh.
3) Cho tập S chứa các phần tử là các số nguyên dương a1, ..., an và số nguyên dương T, trong đó ai ≤ T, với ∀ i = 1, 2, .., n. Liệt kê mọi phương án xếp các đồ vật vào các túi có sức chứa T.
4) Giả thiết có balô với dung lượng T và *n* đồ vật với trọng lượng a1, a2, ..., an và giá trị tương ứng c1, c2, ..., cn. Liệt kê mọi phương án xếp các đồ vật vào ba lô có sức chứa T.
5) Có *n* thành phố và giữa các thành phố xác định một thông số là chi phí đi lại. Một hành trình là một cách đi xuất phát từ một thành phố, qua tất cả các thành phố còn lại và quay lại vị trí xuất phát. Liệt kê mọi hành trình có thể.