

Analysis and Design of Algorithms**Lecture 13**
Generating Method

Lecturer: Ha Dai Duong
duonghd@mta.edu.vn

2/2/2017

1

Nội dung

1. Lược đồ chung
2. Bài toán chuỗi 3 ký tự
3. Liệt kê tập con của tập N phần tử
4. Bài toán tập con K phần tử
5. Hoán vị tập N phần tử

2/2/2017

2

Nội dung

- 1. Lược đồ chung**
2. Bài toán chuỗi 3 ký tự
3. Liệt kê tập con của tập N phần tử
4. Bài toán tập con K phần tử
5. Hoán vị tập N phần tử

2/2/2017

3

Bài toán tổ hợp

- Có n biến $x_1, x_2, x_3, \dots, x_n$
- Mỗi biến x_i có thể mang trị thuộc về 1 tập hợp P_i
 \rightarrow Miền của bài toán là tập tích
 $P_1 \times P_2 \times P_3 \times \dots \times P_n$
- Phép gán trị (assignment): Là một bộ trị
 $a_1, a_2, a_3, \dots, a_n$
 Trong đó $a_i \rightarrow a_i \in P_i$
- Một lời giải của bài toán là 1 phép gán trị.
- Một phép gán trị được gọi là một cấu hình.

2/2/2017

4

Ví dụ 1

- Ví dụ:** Có 3 nhân viên bảo vệ làm 3 ca sáng, chiều tối. Trong 1 ca chỉ có 1 bảo vệ. Hỏi các cách bố trí các bảo vệ?
- Mã hóa bài toán:**
 $\{x, y, z\}$ là tập biến có thứ tự mô tả cho 3 ca :sáng, chiều, tối theo thứ tự.
 Miền trị của 3 biến là $\{a, b, c\}$ mô tả cho 3 bảo vệ.

Các phép gán

x	y	z
a	b	c
a	c	b
b	a	c
b	c	a
c	a	b
c	b	a

♦Số lời giải là số hoán vị của tập hợp 3 phần tử này:
 $3! = 3 \times 2 \times 1 = 6$.

2/2/2017

5

Ví dụ 2

- Ví dụ:** Tìm số chuỗi có độ dài 3 ký tự xyz với

$x \in \{a, b, c\}$,

$y \in \{d, e\}$,

$z \in \{m, n, t\}$

- Nhận xét: 3 biến có 3 miền trị khác nhau

x	y	z
a	d	m
a	d	n
a	d	t
a	e	m
a	e	n
a	e	t
b	d	m
b	d	n
b	d	t
b	e	m
b	e	n
b	e	t
c	d	m
c	d	n
c	d	t
c	e	m
c	e	n
c	e	t

Số phép gán:
 $3 \times 2 \times 3 = 18$

Tích của các số phần tử của các miền trị

Độ phức tạp: n^m với

2/2/2017

6

Bài toán tổ hợp

Bài toán tổ hợp
có độ phức tạp là $n!$ hoặc n^m

Làm thế nào tạo ra
các phép gán trị ?
→ **Phương pháp sinh.**

2/2/2017

7

Lược đồ chung

- Phương pháp sinh: Từ dữ liệu ban đầu, sinh ra dữ liệu kế tiếp cho đến khi kết thúc.
- Dùng để giải quyết bài toán liệt kê của lý thuyết tổ hợp.
- Điều kiện của thuật toán sinh:
 - (1) Có thể xác định 1 thứ tự tập các cấu hình của tổ hợp (thứ tự của các phép gán trị, thường dùng thứ tự từ điển).
 - (2) Có một cấu hình cuối (điều kiện kết thúc của giải thuật).
 - (3) Có một cách để suy ra được cấu hình kế tiếp.

2/2/2017

8

Thứ tự từ điển

- $S1 = "1234589"$
- $S2 = "1235789"$
- $S1 < S2$ nếu có 1 vị trí i tại đó
 $S1[i] < S2[i]$

Thứ tự từ điển ngược (ngược lại với thứ tự từ điển)

2/2/2017

9

Ví dụ

Bài toán: Tìm số chuỗi có độ dài 3 ký tự xyz với $x \in \{a,b,c\}$, $y \in \{d,e\}$, $z \in \{m,n,t\}$

Cấu hình ban đầu: trị đầu tiên của mỗi miền trị

Cách sinh: Lấy trị kết tiếp của mỗi miền trị theo cơ chế vòng tròn

Cấu hình cuối: trị cuối cùng của mỗi miền trị

x	y	z
a	d	m
a	d	n
a	d	t
a	e	m
a	e	n
a	e	t
b	d	m
b	d	n
b	d	t
b	e	m
b	e	n
b	e	t
c	d	m
c	d	n
c	d	t
c	e	m
c	e	n
c	e	t

Dùng thứ tự từ điển để so sánh các phép gán trị.
Ví dụ:
adm < adn

2/2/2017 10

Lược đồ chung

```

Procedure Generate
Begin
  c = InitialConfigure; //cấu hình ban đầu
  Process (c); // xử lý cấu hình đang có
  if c=LastConfigure then Stop:=true
  else stop := false;
  while (not stop) do
  Begin
    //Sinh cấu hình kế tiếp từ cấu hình đang có
    c=getNextConfigure(c);
    Process (c); // xử lý cấu hình này
    if c= LastConfigure then stop = true;
  End;
End;

```

2/2/2017 11

Nội dung

1. Lược đồ chung
2. Bài toán chuỗi 3 ký tự
3. Liệt kê tập con của tập N phần tử
4. Bài toán tập con K phần tử
5. Hoán vị tập N phần tử

2/2/2017 12

Bài toán

Bài toán: *Tìm số chuỗi có độ dài 3 ký tự xyz với*
 $x \in \{a, b, c\},$
 $y \in \{d, e\},$
 $z \in \{m, n, t\}$

2/2/2017

13

Cài đặt

```

1 3RYIU.CPP
2 #include <conio.h>
3 #include <string.h>
4 // Khai báo cấu trúc miền trị và các thuật toán trên miền trị
5 struct DOMAIN
6 {
7     char values[27]; // tập trị
8     int n; // số phần tử
9     int cur; // vị trí hiện hành
10    int serviced; // đã lấy trị chưa?
11};
12
13void Init(DOMAIN &D, char* S)
14{
15    strcpy(D.values, S);
16    D.n = strlen(S);
17    D.cur = 0;
18    D.serviced = 0;
19}
20
21char getValue(DOMAIN &D) // Lấy 1 trị trong domain
22{
23    char x = D.values[D.cur++];
24    if (D.cur == D.n) D.cur = 0; // cập nhật vị trí và trạng thái
25    if (D.serviced == 0) D.serviced = 1;
26    return x;
27}
28
29int isCircular(DOMAIN D) // Kiểm tra đã hết 1 vòng trị chưa?
30{
31    return D.cur == 0 && D.serviced == 1;
32}

```

2/2/2017

14

```

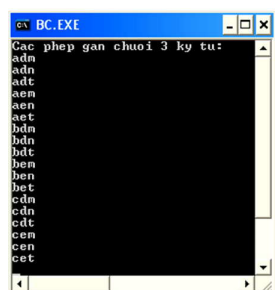
1 3RYIU.CPP
2
3 void Generate()
4 {
5     char vars[4]; // 3 biến là 3 ký tự
6     DOMAIN D0, D1, D2; // 3 domain cho 3 ký tự
7     Init(D0, "abc"); // Khởi tạo 3 domain
8     Init(D1, "de");
9     Init(D2, "mt");
10    // Lấy cấu hình ban đầu
11    vars[0] = getValue(D0);
12    vars[1] = getValue(D1);
13    vars[2] = getValue(D2);
14    vars[3] = 0; // khóa chuỗi
15    // Xu ly cấu hình ban đầu
16    puts(vars);
17    // Kiểm tra ngưng
18    int Stop = isCircular(D0) && isCircular(D1) && isCircular(D2);
19    while(!Stop)
20    {
21        // Lấy cấu hình kế tiếp
22        if (isCircular(D1) && isCircular(D2)) vars[0] = getValue(D0);
23        if (isCircular(D2)) vars[1] = getValue(D1);
24        vars[2] = getValue(D2);
25        // Xu ly cấu hình
26        puts(vars);
27        // Kiểm tra ngưng
28        Stop = isCircular(D0) && isCircular(D1) && isCircular(D2);
29    }
30}
31
32void main()
33{
34    clrscr();
35    puts("Gie phép gần chuỗi 3 ký tự:");
36    Generate();
37    getch();
38}

```

2/2/2017

15

Minh họa



2/2/2017

16

Nội dung

1. Lược đồ chung
2. Bài toán chuỗi 3 ký tự
- 3. Liệt kê tập con của tập N phần tử**
4. Bài toán tập con K phần tử
5. Hoán vị tập N phần tử

2/2/2017

17

Bài toán

- Mã hóa tập biến: Tập biến gồm n biến ký tự theo thứ tự các phần tử \rightarrow mảng n ký tự.
- Miền trị của mỗi biến $\{ '0', '1' \}$. '0' mô tả cho tình huống phần tử này **không có** trong tập con, '1': mô tả cho tình huống phần tử này **có mặt** trong tập con.
- Với tập cha là 4 phần tử $X = \{ a, b, c, d \}$, có thể dùng mảng "0111" mô tả cho tập con $\{ b, c, d \}$.
 \rightarrow Mỗi tập con được biểu diễn là một chuỗi (xâu) nhị phân.
- Trạng thái khởi tạo: "0000" mang ý nghĩa tập trống.
- Trạng thái kết thúc: "1111" mang ý nghĩa là tập cha.

2/2/2017

18

Ví dụ

- Với tập cha gồm 4 phần tử, có 2^4 tập con b với các biểu diễn:

vars	p(b)	vars	p(b)
0000	0	1000	8
0001	1	1001	9
0010	2	1010	10
0011	3	1011	11
0100	4	1100	12
0101	5	1101	13
0110	6	1110	14
0111	7	1111	15

2/2/2017

19

Cộng 1 đơn vị

0000	0001
0001	0010
0011	0100
0111	1000

- Gọi i : vị trí bit 0 đầu tiên từ bên phải.
- Cho các bit 1 bên phải vị trí i thành 0
- Cho bit i mang trị 1

```
i = n-1;
while (i >= 0 && vars[i] == '1') vars[i--] = '0';
vars[i] = '1';
```

2/2/2017

20

Cài đặt

```
// Bài toán liệt kê các tập con- KHÔNG TỐI ƯU MÀ LENH
#include <stdio.h>
#include <conio.h>
#include <string.h>
#define TRUE 1
#define FALSE 0

void InitConfigure(char vars[], int n)
{
    for (int i=0; i<n; i++) vars[i]='0';
}

void Output(char vars[], int n, int count)
{
    printf("\nTẬP con thứ %d:\t", count);
    for (int i=0; i<n; i++) putchar(vars[i]);
}

void NextConfigure(char vars[], int n)
{
    int i=n-1;
    while(i>=0 && vars[i]=='1') // cho các bit 1 ở bên phải về 0
    {
        vars[i]='0';
        i--;
    }
    vars[i]='1'; // i là vị trí bit 0 đầu tiên từ bên phải
}

int main()
{
    char vars[10];
    InitConfigure(vars, 4);
    Output(vars, 4, 1);
    while (1)
    {
        NextConfigure(vars, 4);
        Output(vars, 4, 2);
    }
    return 0;
}
```

2/2/2017

21

Cài đặt

```

1 // TAPCON.CPP
2 int LastConfigure(char * vars, int n)
3 {
4     for (int i=0; i<n; i++) if (vars[i]!='0') return 0;
5     return 1;
6 }
7 void Generate(char vars[], int n)
8 {
9     int Stop=FALSE;
10    InitConfigure (vars,n);
11    int count=1;
12    OutputVars(n,count);
13    while (!Stop)
14    {
15        NextConfigure(vars,n);
16        count++;
17        OutputVars(n,count);
18        Stop=LastConfigure(vars,n);
19    }
20 }
21 void main()
22 {
23     clrscr();
24     int n; // so phan tu
25     puts("Chuong trinh liet ke cac tap con");
26     printf("Nhap so phan tu cua tap:");
27     scanf("%d",&n);
28     char* vars= new char[n];
29     Generate(vars,n);
30     getch();
31 }

```

2/2/2017

22

Minh họa

2/2/2017

23

Nội dung

1. Lược đồ chung
2. Bài toán chuỗi 3 ký tự
3. Liệt kê tập con của tập N phần tử
- 4. Bài toán tập con K phần tử**
5. Hoán vị tập N phần tử

2/2/2017

24

Bài toán

- Liệt kê các tập con k phần tử của tập n phần tử.
- Ví dụ: Các tập con 3 phần tử của tập

{ 1,2,3,4,5 } là:

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

{ 1,2,4 } { 2,3,5 }

{ 1,2,5 } { 2,4,5 }

{ 1,3,4 } { 3,4,5 }

{ 1,3,5 }

{ 1,4,5 }

$$\begin{aligned} C_5^3 &= 5! / (3! * (5-3)!) \\ &= 5! / (3! * 2!) \\ &= 4 * 5 / 2 = 10 \end{aligned}$$

Tổ hợp n chập k

2/2/2017

25

Bài toán

- Ánh xạ tập hợp bất kỳ n phần tử vào tập $X = \{ 1, 2, \dots, n \}$
- Một tập con k phần tử của X là một bộ có thứ tự $a_1 a_2 a_3 \dots a_k$ với
 $1 \leq a_1 < a_2 < a_3 < \dots < a_k \leq n$

2/2/2017

26

Ý tưởng

- Tập con đầu: $\{ 1, 2, 3, \dots, k \}$
 Ví dụ $\{ 1, 2, 3 \}$ với $k=3, n=5$
- Tập con cuối: $\{ (n-k+1), (n-k+2), \dots, n \}$
 Ví dụ: $\{ 3, 4, 5 \}$ với $k=3, n=5$

2/2/2017

27

Cách sinh tập con kế tiếp từ tập con đã có $a_1 a_2 a_3 a_4 \dots a_k$, chỉ số ở đây đi từ 1

- (1) Tìm vị trí đầu tiên từ bên phải 1 vị trí i sao cho $a[i] \neq n-k+i$

$i=k$;

while ($a[i] == n-k+i$) $i--$;

- (2) Thay $a[i]$ bằng $a[i] + 1$

$a[i] = a[i] + 1$;

- (3) Thay các trị sau i ($a[j]$) bằng các trị $a[j]+1, \dots$

for ($j=i+1; j \leq k; j++$)

$a[j] = a[j] + j - i$;

$n=8, k=6$

$\{1, 2, 5, 6, 7, 8\}$
($i=6, n-k+i=8$)

$\{1, 3, 5, 6, 7, 8\}$

$\{1, 3, 4, 5, 6, 7\}$

Tìm vị trí đầu tiên khác với nhóm trị ở cuối tập cha theo thứ tự

2/2/2017

28

Cài đặt

```
//TOHOPN_K.CPP ** To hop n chap k phan tu
#include <stdio.h>
#include <conio.h>
void Init (int* result, int k)
{ for (int i=1; i<=k; i++) result[i]=i;
}
void Process (int* result, int n)
{ printf("\n");
  for (int i=1; i<=n; i++) printf("%d,", result[i]);
}
void NextCombination (int* result, int n, int k, int & Stop)
{ int i = k;
  while (i>0 && result[i]==n-k+i) i--;
  if (i>0)
  { result[i]++;
    for (int j=i+1; j<=k; j++) result[j] = result[i] + j - i;
  }
  else Stop = 1;
}
```

2/2/2017

29

Cài đặt

```
void Generate (int *result, int n, int k)
{ int Stop=0;
  Init(result, k);
  Process(result, k);
  while (!Stop)
  { NextCombination(result, n, k, Stop);
    if (!Stop) Process(result, k);
  }
}
void main()
{ int n, k, *result;
  printf("Nhap so phan tu cua tap n="); scanf("%d", &n);
  printf("Nhap so phan tu cua tap con k="); scanf("%d", &k);
  result = new int [k+1];
  Generate(result, n, k);
  getch();
}
```

Thêm dòng: delete[] result;

2/2/2017

30

Nội dung

1. Lược đồ chung
2. Bài toán chuỗi 3 ký tự
3. Liệt kê tập con của tập N phần tử
4. Bài toán tập con K phần tử
5. **Hoán vị tập N phần tử**

2/2/2017

31

Bài toán

- Cho tập $X = \{1, 2, 3, \dots, n\}$. Hãy liệt kê tất cả các hoán vị của tập này.
- Một hoán vị của X là một bộ $A = (a_1, a_2, \dots, a_n)$ với $a_i \neq a_j$ nếu $i \neq j$
- Định nghĩa 1 thứ tự:
 $A = (a_1, a_2, \dots, a_{k-1}, \mathbf{a_k}, \dots, a_n)$ là hoán vị trước của
 $A' = (a'_1, a'_2, \dots, a'_{k-1}, \mathbf{a'_k}, \dots, a'_n)$ nếu tìm được vị trí k sao cho $\mathbf{a_k < a'_k}$
- Ví dụ : 1234**5**67 là hoán vị trước của
 1234**6**57
- Đây chính là thứ tự từ điển.
- Độ phức tạp $n!$

2/2/2017

32

Các hoán vị của $X = \{1, 2, 3, 4\}$

{1,2,3,4} (đầu)	{3,1,2,4}	{3,4,2,1}	j=1
{1,2,4,3}	{3,1,4,2}	{3,4,2,1}	k=2
{1,3,2,4}	{3,2,1,4}	{3,4,2,1}	
{1,3,4,2}	{3,2,4,1}	{3,4,2,1}	
{1,4,2,3}	{3,4,1,2}	{4,3,2,1}	hoán vị a[j], a[k]
{1,4,3,2}	{3,4,2,1}	{4,3,2,1}	
{2,1,3,4}	{4,1,2,3}	{4,1,2,3}	Đảo mảng con từ a[j+1] đến a[n]
{2,1,4,3}	{4,1,3,2}		
{2,3,1,4}	{4,2,1,3}		
{2,3,4,1}	{4,2,3,1}		
{2,4,1,3}	{4,3,1,2}		
{2,4,3,1}	{4,3,2,1} (cuối)		chỉ số đi từ 1

2/2/2017

33

Ý tưởng

- trạng thái trước {1,3,4,2} trạng thái sau: {1,4,2,3}

Giải thuật

- Tìm chỉ số lớn nhất j mà $a_j < a_{j+1}$ từ phía phải vì đây là phần tử sẽ bị hoán vị.

1 3 4 2 ($j=2$) $\rightarrow j=n-1$; while ($a[j] > a[j+1]$) $j--$;

- Tìm vị trí đầu tiên k đi ngược từ cuối tập trị với $a[k] > a[j]$

1 3 4 2 ($k=3$) $\rightarrow k=n$; while ($a[j] > a[k]$) $k--$;

- Hoán vị $a[j]$ với $a[k]$

1 4 3 2

- Lật ngược đoạn $a_{j+1} \dots a_n$

1 4 2 3 trạng thái kế tiếp

2/2/2017

34

Cài đặt

{1,2,3,4,5,..., n}

{n,..., 5,4,3,2,1}

```

// Bài toán hoán vị tập n phần tử
#include <stdio.h>
#include <conio.h>
void InitConfigure (int vars[], int n)
{
    for (int i=1; i<=n; i++) vars[i]=i;
}
int isLastConfigure (int vars[], int n)
{
    for (int i=1; i<=n; i++) if (vars[i]<vars[i+1]) return 0;
    return 1;
}
void Output (int vars[], int n, int count)
{
    printf("\nHoán vị thu %d:\t", count);
    for (int i=1; i<=n; i++) printf("%d, ", vars[i]);
}

```

Hoán vị thu 8: 2, 1, 4, 3,

2/2/2017

35

Cài đặt

```

void NextConfigure (int vars[], int n)
{
    int j=n-1;
    while (j>0 && vars[j]>vars[j+1]) j-- ;
    int k=n;
    while (vars[j]>vars[k]) k--;
    int t=vars[j];
    vars[j]=vars[k];
    vars[k]=t;
    int left=j+1, right=n;
    while (left<right)
    {
        t=vars[left];
        vars[left]=vars[right];
        vars[right]=t;
        left++; right--;
    }
}

```

Tìm chỉ số lớn nhất j mà $a_j < a_{j+1}$ từ phía phải vì đây là phần tử sẽ bị hoán vị.

Tìm vị trí đầu tiên k đi ngược từ cuối tập trị với $a[k] > a[j]$

Hoán vị $a[j]$, $a[k]$

Đảo ngược nhóm trị $a_{j+1} \dots a[n]$

2/2/2017

36

Cài đặt

```

1 // HOANVI.CPP
2 void Generate(int vars[],int n)
3 {
4     InitConfigure (vars,n);
5     int count=1;
6     Output(vars,n,count);
7     int Stop=isLastConfigure(vars,n);
8     while (!Stop)
9     {
10         NextConfigure(vars,n);
11         count++;
12         Output(vars,n,count);
13         Stop=isLastConfigure(vars,n);
14     }
15 }
16 void main()
17 {
18     clrscr();
19     int n; // so phan tu
20     puts("Chương trình hoan vi tap n phan tu");
21     printf("Nhap so phan tu cua tap:");
22     scanf("%d",&n);
23     int* vars= new int[n+1];
24     Generate(vars,n);
25     getch();
26     // Thêm dòng: delete[] vars;
27 }

```

2/2/2017

37

Kết quả

```

1 BC.EXE
2 Chương trình hoan vi tap n phan tu
3 Nhap so phan tu cua tap:4
4
5 Hoan vi thu 1: 1 2 3 4
6 Hoan vi thu 2: 1 2 4 3
7 Hoan vi thu 3: 1 3 2 4
8 Hoan vi thu 4: 1 3 4 2
9 Hoan vi thu 5: 1 4 2 3
10 Hoan vi thu 6: 1 4 3 2
11 Hoan vi thu 7: 2 1 3 4
12 Hoan vi thu 8: 2 1 4 3
13 Hoan vi thu 9: 2 3 1 4
14 Hoan vi thu 10: 2 3 4 1
15 Hoan vi thu 11: 2 4 1 3
16 Hoan vi thu 12: 2 4 3 1
17 Hoan vi thu 13: 3 1 2 4
18 Hoan vi thu 14: 3 1 4 2
19 Hoan vi thu 15: 3 2 1 4
20 Hoan vi thu 16: 3 2 4 1
21 Hoan vi thu 17: 3 4 1 2
22 Hoan vi thu 18: 3 4 2 1
23 Hoan vi thu 19: 4 1 2 3
24 Hoan vi thu 20: 4 1 3 2
25 Hoan vi thu 21: 4 2 1 3
26 Hoan vi thu 22: 4 2 3 1
27 Hoan vi thu 23: 4 3 1 2
28 Hoan vi thu 24: 4 3 2 1

```

2/2/2017

38

Bài tập

1. Liệt kê các tập con của tập {a,b,c,d,e,f}.
2. Liệt kê các tập con 3, 4, 5 phần tử từ tập 6 phần tử.
3. Liệt kê các hoán vị của tập 3,4,5 phần tử

2/2/2017

39