

Giải bài toán tổ hợp

Bằng phương Quay lui

Phương pháp Quay lui

Điều kiện để giải bài toán bằng phương pháp sinh

Một bài toán mẫu

- Sinh chuỗi nhị phân n phần tử
- Sinh tập con k phần tử từ tập n phần tử cho trước với điều kiện không lặp và phần tử $A[i] < A[i+1]$
- Liệt kê hoán vị của tập n phần tử theo thứ tự từ điển.

Yêu cầu bài tập:

Bài tập 01: Liệt kê chuỗi nhị phân gồm N phần tử. N nhập từ bàn phím, đếm tổng số số nhị phân tìm được bằng phương pháp quay lui.

Bài tập 02: Liệt kê chuỗi con gồm k phần tử trong tập N phần tử cho trước với các giá trị từ $1-N$. theo thứ tự từ điển.

Bài tập 03: Liệt kê tất cả các hoán vị của tập N phần tử với các giá trị từ $1-N$. theo thứ tự từ điển

Hướng dẫn:

Bài tập 1: Liệt kê chuỗi nhị phân gồm n phần tử. n nhập từ bàn phím, đếm tổng số số nhị phân tìm được bằng phương pháp quay lui.

```
//Bài liệt kê số nhị phân gồm N phần tử
// - Số p tử của cấu hình : N
// - Tập khả năng : [0,1]
#include <iostream>
using namespace std;
#define Max 10
int X[Max], N, Count; //X[Max] mảng chứa cấu hình cần tìm, N là số phần tử của cấu hình.
//khởi tạo giá trị ban đầu
void Init()
{
    //Nhập giá trị N<=10
    cout << "Nhập N: "; cin >> N;
}
//Phương thức xuất kết quả cấu hình
void Output()
```

```

{
    for (int j = 0; j < N; j++)
    {
        cout << X[j];
    }
    Count++;
    cout << endl;
}
}
void Try(int i)
{
    for (int v = 0; v <= 1; v++)
    {
        X[i] = v;
        if (i == N - 1)
        {
            Output();
        }
        else
        {
            Try(i + 1);
        }
    }
}
}
void main()
{
    Init();
    Try(0);
    cout << "Tao duoc :" << Count << " Phan tu";
    system("pause");
}

```

Bài tập 02:

Chia theo từng bước của bài tập 1: thực hiện code sau để cài đặt thuật toán:

```

#include <iostream>
using namespace std;

#define max 10
int n,k;
int x[max]; //mảng chứa cấu hình
int Count;
void Init()

```

```

{
    cout << "Nhap n: "; cin >> n;
    cout << "Nhap k: "; cin >> k;
}
void Output()
{
    for (int i = 1; i <= k; i++)
    {
        cout << x[i];
    }
    Count++;
    cout << endl;
}
void Try(int i)
{
    for (int v = x[i-1]+1; v <= n-k+i; v++)
    {
        x[i] = v;
        if (i == k)
        {
            Output();
        }
        else
        {
            Try(i + 1);
        }
    }
}
void main()
{
    Init();
    Try(0);
    cout << "tao duoc: " << Count << " phan tu";
    system("pause");
}

```

Bài tập 03: Tìm hoán vị của tập n phần tử, theo thứ tự từ điển.

```

#include <iostream>
using namespace std;
#define max 10

int x[max];
int n, Count;

```

```

int F[max]; //chứa trạng thái đã chọn của phần tử thứ i
//f[v]=0 chưa chọn.
void Init()
{
    cout << "Nhap N: "; cin >> n;
    for (int v = 0; v < n; v++)
    {
        F[v] = 0;
    }
}
void Output()
{
    for (int i = 0; i < n; i++)
    {
        cout << x[i];
    }
    Count++;
    cout << endl;
}
void Try(int i)
{
    for (int v = 1; v <= n; v++)
    {
        if (F[v - 1] == 0)
        {
            x[i] = v;
            if (i == n - 1)
                Output();
            else
            {
                F[v - 1] = 1;
                Try(i + 1);
                F[v - 1] = 0;
            }
        }
    }
}
void main()
{
    Init();
    Try(0);
    system("Pause");
}

```

Bài tập áp dụng:

Dạng bài 01: Dạng liệt kê chuỗi nhị phân:

1. Giả sử ổ khóa có n công tắc. Mỗi công tắc có 1 trong 2 trạng thái “đóng” và “mở”. khóa mở được nếu có ít nhất một nửa công tắc ở trạng thái mở. Hãy lập trình liệt kê tất cả các cách mở khóa. (Bài toán nhị phân).
2. Cho số nguyên dương N . Hãy lập trình liệt kê tất cả các dãy nhị phân độ dài N bit thỏa điều kiện có ít nhất là 2 bit 0. Mỗi dãy bit nằm trên 1 dòng.
3. Cho số nguyên dương N . Hãy lập trình liệt kê tất cả dãy nhị phân độ dài N bit thỏa điều kiện số lượng bit 1 chiếm từ 60% trở lên. Mỗi dãy bit nằm trên 1 dòng.
4. Cho số nguyên dương N . Hãy lập trình liệt kê tất cả dãy nhị phân độ dài N bit thỏa điều kiện số bit 1 luôn nhiều hơn số bit 0. mỗi dãy bit nằm trên 1 dòng.
5. Cho số nguyên dương N . Hãy lập trình liệt kê tất cả dãy nhị phân độ dài N bit thỏa điều kiện số bit 1 xen kẽ 0. mỗi dãy bit nằm trên 1 dòng.

Dạng bài 02: Tìm tập con k phần tử:

6. Cho số nguyên dương N . Hãy lập trình để liệt kê các tập con K phần tử của tập $S = \{1, 2, \dots, N\}$. trong đó $K \leq N$. (Bài toán liệt kê K phần tử trong tập N Phần tử)
7. Tại một đại hội đoàn trường, đại biểu cần bỏ phiếu bầu ra 7 thành viên vào BCH từ danh sách gồm 9 ứng viên (An, Bình, Châu, Cường, Minh, Thắng, Hạnh, Toàn, Khánh). Biết rằng Châu, Hạnh là nữ, các ứng viên còn lại là Nam. Hãy lập trình liệt kê tất cả các cách chọn ra BCH từ danh sách trên sao cho BCH phải có nữ. mỗi cách chọn nằm trên 1 dòng.

//Tại một đại hội đoàn trường, đại biểu cần bỏ phiếu bầu ra 7 thành viên vào BCH từ danh sách gồm 9 ứng viên(An, Bình, Châu, Cường, Minh, Thắng, Hạnh, Toàn, Khánh).Biết rằng Châu, Hạnh là nữ, các ứng viên còn lại là Nam.Hãy lập trình liệt kê tất cả các cách chọn ra BCH từ danh sách trên sao cho BCH phải có nữ.mỗi cách chọn nằm trên 1 dòng.

```
#include <iostream>
```

```
using namespace std;
```

```
//Khai báo
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
#define n 9
```

```
#define k 7
```

```
typedef struct ungvien
```

```
{
```

```
    char Ten[10];
```

```
    int Phai;//1 là nữ, 0 là nam
```

```
} UNGVIEN;
```

```
int A[k+1]; //mảng sẽ chứa cấu hình chọn được
```

```
int stop;//n=9, k=7,
```

```
//Mảng ứng viên là toàn cục.
```

```

UNGVIENT ungvien[n+1] = {
    { "An", 0 }, //1
    { "Binh", 0 }, //2
    { "Chau", 1 }, //3
    { "Cuong", 0 }, //4
    { "Minh", 0 }, //5
    { "Thang", 0 }, //6
    { "Hanh", 1 }, //7
    { "Toan", 0 }, //8
    { "Khanh", 0 } //9
};
//Khởi tạo giá trị ban đầu
void Init()
{
    //khởi tạo giá trị để kiểm tra đã đủ cấu hình chưa
    //stop=false chưa đủ cấu hình. true : đủ cấu hình.
    stop = FALSE;
    //Khởi tạo cấu hình đầu tiên của cách chọn.
    for (int i = 1; i <=k; i++)
    {
        A[i] = i;
    }
}
void PrintDanhSachUngVien()
{
    for (int i = 0; i < n; i++)
    {
        cout << ungvien[i].Ten;
    }
}
//Hàm hiển thị thông tin
void OutPut()
{
    int count = 0;
    for (int i = 1; i <=k; i++)
    {
        if (ungvien[A[i]-1].Phai == 1)
            count++;
    }
    if (count ==1 )
    {
        for (int i = 1; i <= k; i++)
        {
            //cout << A[i];
            cout << ungvien[A[i]-1].Ten<<" ";

```

```

        }
        cout << endl;
    }
}

void main()
{
    //PrintDanhSachUngVien();
    List();
    system("pause");
}

```

8. Để chuẩn bị cho giải bóng đá mini toàn tỉnh. HLV phải lập đội tuyển bằng cách chọn ra 6 VĐV trong số 8 cầu thủ được triệu tập (Chiến, Bảo, Đức, Sỹ, Sơn, Hùng, Trường, Vinh). Hãy lập trình liệt kê tất cả cách thành lập đội tuyển từ danh sách tuyển thủ trên. biết rằng Đức là tuyển thủ số 1 nên anh luôn được HLV ưu tiên chọn vào đội tuyển. mỗi cách chọn hiển thị trên 1 dòng. tên các VĐV được cách nhau khoảng trắng.

```

#include <iostream>
using namespace std;
//Khai báo

#define TRUE 1
#define FALSE 0
#define n 8
#define k 6
typedef struct ungvien
{
    char Ten[10];
    int Utien;//1 là uutien, 0 là khonguutien
} UNGVIEN;

int A[k + 1]; //mảng sẽ chứa cấu hình chọn được
int stop; //n=9, k=7,
//Mảng ứng viên là toàn cục.
UNGVIEEN ungvien[n + 1] = {
    { "Chien", 0 }, //1
    { "Bao", 0 }, //2
    { "Duc", 1 }, //3
    { "Si", 0 }, //4
    { "Son", 0 }, //5
    { "Hung", 0 }, //6
    { "Truong", 0 }, //7
    { "Vinh", 0 }, //8
}

```

```

};
//Khởi tạo giá trị ban đầu
void Init()
{
    //khởi tạo giá trị để kiểm tra đã đủ cấu hình chưa
    //stop=false chưa đủ cấu hình. true : đủ cấu hình.
    stop = FALSE;
    //Khởi tạo cấu hình đầu tiên của cách chọn.
    for (int i = 1; i <= k; i++)
    {
        A[i] = i;
    }
}
//Hàm hiển thị thông tin
void OutPut()
{
    int count = 0;
    for (int i = 1; i <= k; i++)
    {
        if (ungvien[A[i]-1].Utien == 1)
            count++;
    }
    if (count == 1)
    {
        for (int i = 1; i <= k; i++)
        {
            //cout << A[i];
            cout << ungvien[A[i]-1].Ten << " ";
        }
        cout << endl;
    }
}
}
void main()
{
    //PrintDanhSachUngVien();
    List();
    system("pause");
}

```


Dạng bài 03: Bài toán hoán vị

9. Trong chuyến du lịch đến thành phố X, một nhóm bạn gồm 6 người { Thủy, Phương, Tú, Nam, Nhân, Thanh} muốn chụp hình lưu niệm các cảnh đẹp mà nhóm đến thăm. các bạn chỉ thích đứng thành hàng ngang để chụp ảnh. để có được nhiều kiểu hình khác nhau cho một cảnh, các bạn trong nhóm sẽ thay đổi vị trí trong mỗi kiểu chụp. Biết rằng Phương và Nhân yêu nhau, nên họ luôn đứng cạnh nhau. Hãy lập trình liệt kê tất cả các cách sắp xếp vị trí để chụp ảnh của nhóm bạn. mỗi cách nằm trên 1 dòng. các tên hiển thị cách nhau khoảng trắng. ([Bài toán hoán vị](#))
10. 10 hiệp sĩ của thành Sim là {Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct} được đức vua mời đến dự yến tiệc tại cung điện. Các hiệp sĩ được sắp ngồi cạnh nhau trên cùng 1 dãy bàn. Tuy nhiên, các hiệp sĩ Jan, Jun và Jul lại có hiềm khích với nhau nên họ không thích ngồi kề nhau trên bàn tiệc. Hãy lập trình liệt kê tất cả cách sắp chỗ ngồi các hiệp sĩ sao cho những hiệp sĩ có hiềm khích với nhau không ngồi cạnh nhau trên bàn tiệc. Mỗi cách sắp chỗ được hiển thị bằng cách liệt kê tên các hiệp sĩ cách nhau khoảng trắng trên cùng 1 dòng ([Bài toán hoán vị](#)).
11. Có 3 nhân viên bảo vệ là A, B, C làm 3 ca sáng, chiều và tối. Biết rằng, trong một ca làm việc chỉ có một bảo vệ và một bảo vệ trong một ngày chỉ làm một ca. Hãy lập trình liệt kê tất cả các cách bố trí bảo vệ trong một ca làm việc.

Dạng bài 03: Bài toán chỉnh hợp lặp

12. Cho 3 ký tự A, B, C và một số nguyên dương n. Viết chương trình liệt kê tất cả các chuỗi tạo ra từ 3 ký tự trên, với chiều dài n, thỏa điều kiện có ít nhất là 2 ký tự A. Mỗi chuỗi nằm trên một dòng.
13. Cho 3 ký tự A, B, C và n là một số nguyên dương
Hãy lập trình liệt kê tất cả các chuỗi tạo ra từ 3 ký tự trên, với chiều dài n sao cho số ký tự B là 3.

Ghi chú:

Sinh viên thực hiện đầy đủ các bài tập theo hướng dẫn