

PHƯƠNG PHÁP QUAY LUI

(Backtracking)

1. Khái niệm

Quay lui là một trong những phương pháp vét cạn không gian lời giải của bài toán để liệt kê nghiệm hoặc tìm nghiệm tối ưu.

Quay lui sử dụng kỹ thuật đệ quy và lần ngược trong quá trình duyệt để tìm nghiệm.

Quay lui thường được áp dụng để giải các bài toán liệt kê hay bài toán duyệt.

2. Thuật toán quay lui

2.1. Bài toán liệt kê

Nghiệm của bài toán liệt kê có dạng một vector gồm n thành phần $X = (x_1, x_2, \dots, x_n)$.

Mỗi phần tử x_i có một tập khả năng P_i (miền giá trị của x_i).

Bài toán liệt kê yêu cầu chỉ ra tất cả trường hợp mang giá trị của từng thành phần của vector X

2.2. Thuật toán quay lui

Tư tưởng

- Xây dựng dần vector X bằng cách thử lần lượt từng khả năng (giá trị) cho từng phần tử x_i
- Khi xác định được giá trị cho phần tử x_n thì tìm được một nghiệm.

Thuật toán tổng quát

- Xây dựng thủ tục Try(i) để tìm giá trị cho thành phần x_i của vector

```
void Try(int i)
{
    for v ∈ {miền giá trị của thành phần thứ i}
    {
        x[i] = v;
        if (i là thành phần cuối cùng)
            <xử lý nghiệm tìm được>;
        else
        {
            Try(i+1);
        }
    }
}
```

<bỏ chọn giá trị v đã chọn trước đó (nếu cần)>;

```
}  
}  
}
```

2.3. Đánh giá thuật toán

Các trường hợp duyệt

- Số trường hợp thử cho x_1 : $|P_1|$
- Số trường hợp thử cho x_2 : $|P_2|$
- ...
- Số trường hợp thử cho x_n : $|P_n|$

Tổng số lần duyệt: $|P_1| * |P_2| * ... * |P_n|$

2.4. Các bước tiếp cận thuật toán quay lui

Mô hình hóa nghiệm của bài toán

- Xác định vector biểu diễn nghiệm và số thành phần của nó $X = (x_1, x_2, \dots, x_n)$
- Xác định miền giá trị của từng thành phần và ràng buộc miền giá trị giữa các thành phần.

Tìm các tối tiểu hóa các thành phần và tối đa hóa các ràng buộc nhằm giảm số trường hợp duyệt.

3. Một số bài toán áp dụng

3.1. Bài toán liệt kê dãy nhị phân độ dài n

Cho số nguyên dương n . Liệt kê tất cả dãy nhị phân độ dài n

Phân tích:

- Nghiệm của bài toán là dãy nhị phân độ dài n được biểu diễn bởi vector $X = (x_1, x_2, \dots, x_n)$
- Miền giá trị của mỗi thành phần là $\{0,1\}$

Cài đặt

```
void TryBin(int i)  
{  
    for (int v = 0; v <= 1; ++v)  
    {
```

```

        x[i] = v;
        if (i == n)
            Print(x, n);
        else TryBin(i+1);
    }
}

```

Độ phức tạp của thuật toán $O(2^n)$. Có thể cài đặt được với $n \leq 25$

Nhận xét:

- Bài toán liệt kê dãy nhị phân là dạng bài toán liệt kê chỉnh hợp lặp
- Việc sinh ra tất cả dãy nhị phân độ dài n tương ứng với việc sinh ra tất cả tập con của tập gồm n phần tử.

3.2. Bài toán liệt kê tổ hợp

Cho 2 số nguyên dương $n, k (k \leq n)$. Hãy liệt kê tất cả cách chọn k phần tử khác nhau từ tập $S = \{1, 2, \dots, n\}$

Phân tích bài toán

- Nghiệm của bài toán là dãy gồm k phần tử được chọn từ tập S
- Nghiệm được mô hình hóa bởi vector $X = (x_1, x_2, \dots, x_k)$

Nhận xét

- Khi liệt kê các cách chọn theo thứ tự từ điển ta có $1 \leq x_1 < x_2 < \dots < x_k \leq n$
- Suy ra $1 \leq x_1 \leq n - k + 1$
- $x_1 + 1 \leq x_2 \leq n - k + 2$
- ...
- $x_{i-1} + 1 \leq x_i \leq n - k + i$

Cài đặt

```

void TryComb(int i)
{
    for (int v = x[i-1]+1; v <= n-k+i; ++v)
    {
        x[i] = v;
        if (i == k)
            Print(x, k);
        else TryComb(i+1);
    }
}

```