DP Bitmask

Kỹ thuật Quy hoạch động trạng thái ứng dụng trong bài toán Hamilton

- Giang Vi -
(Email : gvi_kei@yahoo.com)

1. Bài toán

Bài toán tìm đường đi Hamilton là một bài toán khá quen thuộc: cho N đỉnh, tìm đường đi sao cho mỗi đỉnh được thăm duy nhất một lần, và chi phí thăm N đỉnh là thấp nhất.
Với dạng toán này, thông thường nếu N nhỏ (<=10), ta có thể giải quyết bài toán bằng phương pháp nhánh cận. Tuỳ vào cách đặt cận mà tốc độ chương trình có thể được nâng lên, tuy nhiên thời gian chạy chương trình vẫn có thể không đảm bảo với một số test khó.

Không những vậy, nếu giới hạn N tăng lên đến 20, thì việc sử dụng phương pháp nhánh cận để giải quyết sẽ không còn khả thi nữa. Trong trường hợp này, ta có cách tiếp cận khác để giải quyết bài toán một cách tối ưu hơn, với độ phức tạp O(N^2*2^N). Đó chính là kỹ Quy hoạch động trạng thái (QHĐ TT) được bàn đến trong bài viết này.

Lấy ví dụ một bài toán cụ thể, ta sẽ tìm hiểu kỹ thuật QHĐ TT thông qua bài toán này:

Từ tập các bài có trên SPOJ (oi)

2645. TRIP

Mã bài: LEM3

Trong kì nghỉ hè năm nay sherry được bố thưởng cho 1 tour du lịch quanh N đất nước tươi đẹp với nhiều thắng cảnh nổi tiếng (vì sherry rất ngoan). Tất nhiên sherry sẽ đi bằng máy bay.
Giá vé máy bay từ đất nước i đến đất nước j là Cij (dĩ nhiên Cij có thể khác Cji). Tuy được bố thưởng cho nhiều tiền để đi du lịch nhưng sherry cũng muốn tìm cho mình 1 hành trình với chi phí rẻ nhất có thể để dành tiền mua quà về tặng mọi người (Các chuyến bay của sherry đều được đảm bảo an toàn tuyệt đối).
Bạn hãy giúp sherry tìm 1 hành trình đi qua tất cả các nước, mỗi nước đúng 1 lần sao cho chi phí là bé nhất nhé.

Input

Dòng 1: N (5 < N < 16)
Dòng thứ i trong N dòng tiếp theo: Gồm N số nguyên, số thứ j là Cij (0 < Cij < 10001)

Output

Gồm 1 dòng duy nhất ghi chi phí bé nhất tìm được

Example

Input:
6
0 1 2 1 3 4
5 0 3 2 3 4
4 1 0 2 1 2
4 2 5 0 4 3
2 5 3 5 0 2
5 4 3 3 1 0

Output:
8

URL: http://www.google.com/url?q=http%3A%2F%2Fvn.spoj.pl%2Fproblems%2FLEM3%2F&sa=D&sntz=1&usg=AFQjCNGydcn7b6tVLy15A3Rq1K3Z0Qo1Ug

2. Giải thuật

a. Ý tưởng chính

Để giải quyết bài toán này theo phương pháp quy hoạch động, các bước thiết lập lời giải phải được lưu trữ để có thể tính toán từ các bài toán con. Rõ ràng ở đây, chúng ta phải lưu các đỉnh đã đi qua và đỉnh cuối cùng của một hành trình bất kỳ. Để làm việc này, ta dùng một mảng nhị phân N phần tử với ý nghĩa bit 1 tượng trưng cho các đỉnh đã được thăm và ngược lại. Từ đó, ta định nghĩa Fx [i,j] là chi phí tối ưu trong trạng thái i, j là đỉnh kết thúc hành trình (0 <= i <= 2^N -1; 1 <= j <= N).

Ví dụ:
Dãy [010011] <=> trạng thái có 3 đỉnh được đi, và đỉnh kết thúc hành trình là đỉnh 5 (tính từ phải qua), với biểu diễn trong hệ thập phân tương ứng của dãy này là 19.
Vậy hàm quy hoạch động của trạng thái này là Fx[19,5].

b. Chi tiết thuật toán

Với mỗi trạng thái I có j là đỉnh đến cuối cùng trong trạng thái này, xét mọi đỉnh k <> j mà k chưa được thăm trong trạng thái I. Gọi trạng thái mà hành trình của nó có chứa đỉnh k là X.

Ví dụ:
I = [010011] có các đỉnh 1,2,5 đã thăm. Giả sử k = 3 (thoả vì chưa đến được thành phố 3 trong trạng thái I), ta được trạng thái X = [010111].

Tóm lại, ở bước này: trạng thái X và trạng thái I chỉ khác nhau duy nhất ở đỉnh k. Dễ nhận thấy số thành phố đi được thấy đi được ở trạng thái X = số thành phố đi được ở trạng thái I + 1, suy ra việc tính toán Fx[X,k] sẽ dựa vào kết quả bài toán có kích thước nhỏ hơn nó là Fx[I,j]

Tiếp theo, ta sẽ tối ưu Fx[X,k] theo công thức : `Fx[X,k] = Min (Fx[X,k], Fx[I,j] + C[j,k]);`

Công thức trên sẽ lấy chi phí nhỏ hơn giữa trạng thái hiện tại (hành trình kết thúc tại k) và trạng thái hành trình kết thúc tại j cộng với chi phí di chuyển từ j -> k.

Bước tìm đường đi có chi phí ngắn nhất chính là giá trị nhỏ nhất trong tất cả đường đi qua N đỉnh mà đường đi đó kết thúc tại thành phố i. Từ đó ta có kết quả bài toán là Min (Fx [2^N - 1, i]) với i =1..N.

Số 2^N-1 thể hiện trạng thái mà N đỉnh đã được thăm. Ví dụ: 2^6 - 1 = 63 (dec) = [111111] (bin)

3. Cài đặt

```
          (* Tác giả source code: Hoàng Minh Trung *)

Procedure Work;
Var
       i,j,k,X:Integer;
Begin
{Chuẩn bị cho bước QHĐ}
       Min:=1000000000;
       For i:=1 to Sq[N]-1 do
              For j:=1 to N do
                     Fx[i,j]:=Min;

 {Đặt cơ sở QHĐ}
        Fx[1,1]:=0;
        For j:=1 to N-1 do
              Fx[Sq[j],j+1]:=0;

        For i:=1 to Sq[N]-1 do {Xét tất cả các trạng thái}
              For j:=1 to N do {Với mỗi trạng thái xét tất cả các thành phố j}
                     If Fx[i,j]>=0 then {Nếu trong trạng thái i đã thăm j}
                            For k:=1 to N do
                                   If k<>j then
```

```
                                        Begin
                                                {Gán X = trạng thái I đã bật bit k}
                                               X:=i Or (1 Shl (k-1));
                                                {Nếu thành phố k chưa được thăm}
                                                If ((i shr (k-1)) And 1 = 0) then
                                                                                Fx[X,k]:=GetMin(Fx[X,k],C[j,k]+Fx[i,j]);
                                                  End;

    {Xét trạng thái tất cả các thành phố đều đã thăm với j là đỉnh kết thúc hành trình}
    For j:=1 to N do
                    If (Fx[Sq[N]-1,j]>=0) and (Fx[Sq[N]-1,j]<Min) then
                            Min:=Fx[Sq[N]-1,j];
    End;
```

Ghi chú:

Ở đây, mảng hằng Sq[i] = 2^i, ta cũng có thể thay mảng này bằng cách sử dụng trực tiếp hàm Shl : 2^i = Sq[i] = 1 ShL i.