CHƯƠNG 7

BÀI TOÁN LUỒNG CỰC ĐẠI TRONG MẠNG

1. Các khái niệm

- Mạng: là đồ thị có hướng G=(V,E), trong đó có một đỉnh s không có cung vào gọi là đỉnh phát và có một đỉnh t không có cung ra gọi là đỉnh thu. Mỗi cung e được gán với một số không âm c(e) gọi là khả năng thông qua của cung e. Qui ước nếu không có cung e thì c(e) = 0.
- Luồng trong mạng: là ánh xạ f: E-> R_+ , f(e) không âm và gọi là luồng trên cung e, thoả 2 điều kiện sau:

  + Điều kiện thông qua: luồng trên cung bất kỳ không vượt quá khả năng thông qua của nó: 0≤f(e)≤c(e)

  + Điều kiện cân bằng: Tổng luồng trên các cung đi vào một đỉnh bằng tổng luồng trên các cung đi ra khỏi đỉnh.
- Giá trị của luồng: tổng luồng trên các cung đi ra đỉnh s sẽ bằng tổng luồng trên các cung đi vào đỉnh t , gía trị này gọi là giá trị của luồng, ký hiệu val(f)
- Luồng cực đại: là luồng f* trong mạng với val(f*) là lớn nhất.

- Bài toán tìm luồng cực đại có nhiều ứng dụng, ví dụ bài toán giao thông: xác định cường độ lớn nhất của dòng vận tải giữa hai nút giao thông (khả năng thông qua tương ứng với số xe trung bình đi trên đoạn đường trong một đơn vị thời gian), hoặc bài toán dẫn dầu: xác định luồng dầu lớn nhất có thể bơm từ tàu chở dầu vào bể chứa (khả năng thông qua tương ứng với tiết diện của các ống).

2. Lát Cắt

Lát cắt (X,X^*) là một phân hoạch tập đỉnh V thành hai tập X và $X^* = V\setminus X$, với $s \in X$, $t \in X^*$.

Khả năng thông qua của lát cắt (X,X^*) là $c(X,X^*)=\sum_{\substack{v\in X \\ w\in X^*}} c(v,w)$

Lát cắt với khả năng thông qua nhỏ nhất được gọi là lát cắt hẹp nhất.

* Định lý 1:

Giá trị của luồng f trong mạng luôn nhỏ hơn hoặc bằng khả năng thông qua của lát cắt (X,X^*) bất kỳ : $val(f) \le c(X,X^*)$, với mọi lát cắt (X,X^*), và giá trị luồng cực đại bằng khả năng thông qua của lát cắt hẹp nhất.

3. Đồ thị Tăng Luồng, Đường Tăng Luồng

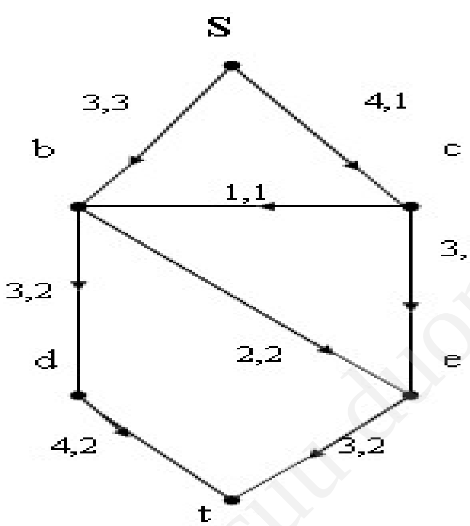
Giả sử f là một luồng trong mạng G = (V,E). Ta xây dựng đồ thị G_f = (V, E_f), với tập cung E_f được xác định theo qui tắc sau: Xét e=(u,v) $\in$ E

+ Nếu f(u,v)<c(u,v), thì (u,v) $\in$ E_f với trọng số c(u,v) - f (u,v): có thể tăng luồng f trên cung (u,v) với giá trị lớn nhất là c(u,v) - f (u,v); cung (u,v) gọi là cung thuận trên G_f

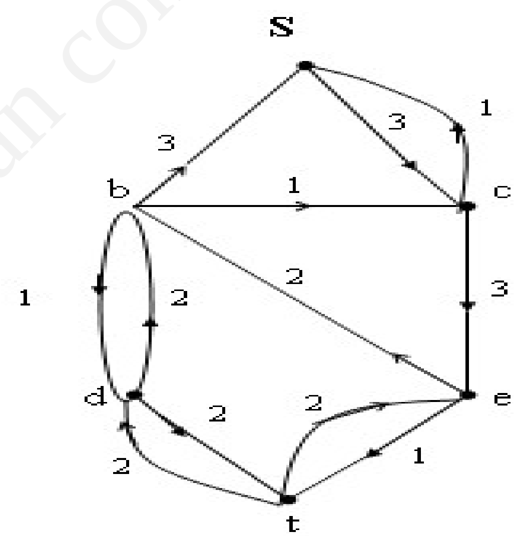
+ Nếu f(u,v) >0, thì (v,u) $\in$ E_f với trọng số f(u,v): : có thể giảm luồng f trên cung (u,v) với giá trị lớn nhất là f (u,v) ; cung (v,u) gọi là cung nghịch trên G_f .

G_f gọi là đồ thị tăng luồng và mọi đường đi từ s đến t trên đồ thị tăng luồng G_f gọi là đường tăng luồng f

Ví dụ: (Các số viết cạnh các cung theo thứ tự là khả năng thông qua và luồng trên cung)



G



Gf

Giả sử P=(s=v_0, v_1, . . . , v_k=t) là một đường tăng luồng trên đồ thị tăng luồng G_f. Gọi δ là giá trị nhỏ nhất của các trọng số của các cung trên đường đi P. Xây dựng luồng f' trên mạng G theo qui tắc sau:

$$f'(u,v) = \begin{cases} f(u,v) + \delta \text{ , nếu } (u,v) \in P \text{ là cung thuận của } G_f \\ f(u,v) - \delta \text{ , nếu } (v,u) \in P \text{ là cung nghịch của } G_f \\ f(u,v), \text{ nếu } (u,v) \notin P \end{cases}$$

Dễ dàng kiểm tra f’ là luồng trong mạng G và val(f’) = val(f) + δ => val(f’) > val(f). Ta sẽ gọi thủ tục biến đổi luồng f thành f’ như trên là thủ tục tăng luồng dọc theo đường P.

Định lý 2: Các mệnh đề dưới đây là tương đương:

(i) f là luồng cực đại trong mạng

(ii) không tìm được đường tăng luồng f;

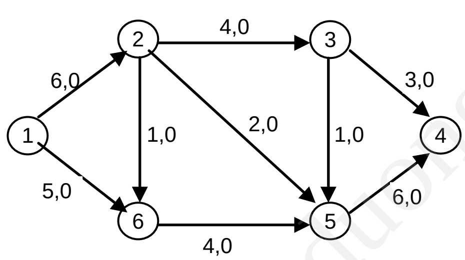
(iii) val(f)=c(X,X^*) với một lát cắt (X,X^*) nào đó.

Định lý 2 là cơ sở để xây dựng thuật toán lặp sau đây để tìm luồng cực đại trong mạng:

4. Thuật Toán Tìm Luồng Cực Đại (thuật tóan Ford-Fulkerson)

Bắt đầu từ một luồng f nào đó trong G (có thể là luồng 0), tìm đường tăng luồng P trong G_f , tăng luồng dọc theo đường P ta được luồng mới f’ trong G. Lặp lại việc tìm đường tăng luồng với luồng mới f’ cho đến khi không tìm được đường tăng luồng thì luồng trong G sẽ đạt cực đại.

Ví dụ: Tìm luồng cực đại trong mạng sau:



Thuật toán không cần xây dựng đồ thị tường minh Gf mà vẫn tìm được đường tăng luồng, bằng cách dùng thuật toán duyệt theo chiều rộng (hoặc chiều sâu) và lần lượt gán nhãn [$\pm$ p(v), d(v)] cho các đỉnh v , với :

p(v) là đỉnh ngay trước đỉnh v trên đường tăng luồng từ s tới v.

d(v): gía trị lớn nhất có thể tăng luồng trên cung (p(v), v) hoặc giảm luồng trên cung (v, p(v))

+p(v): cần tăng luồng theo cung (p(v), v) ; -p(v): cần giảm luồng theo cung (v, p(v))

*** Thuật toán tìm đường tăng luồng**

int Find_Path(){

```
gán tất cả các đỉnh là chưa thăm ;
queue=∅ ; queue<=s ; p[s]=s ; d[s]=+vc ;tham[s]=true;
while (queue!=∅ ){
    u<=queue;
    for (v chua tham) {
        if  ( (c[u,v]>0)&&( f[u,v] <c[u,v]) ){  // Gf co cung thuan (u,v)
            p[v]=u; d[v]=min {d[u], c[u,v] – f[u,v]};
            if (v == t)  return 1;
            queue<=v; tham[v]=true;
        }
        if  ( (c[v,u]>0)&&(f[v,u]>0){ //co cung nghich (u,v)
            p[v]=-u; d[v]=min {d[u], f[v,u]} ;
            if (v == t) return 1;
            queue<=v; tham[v]=true;
        }
    }
}
return 0;
}
```

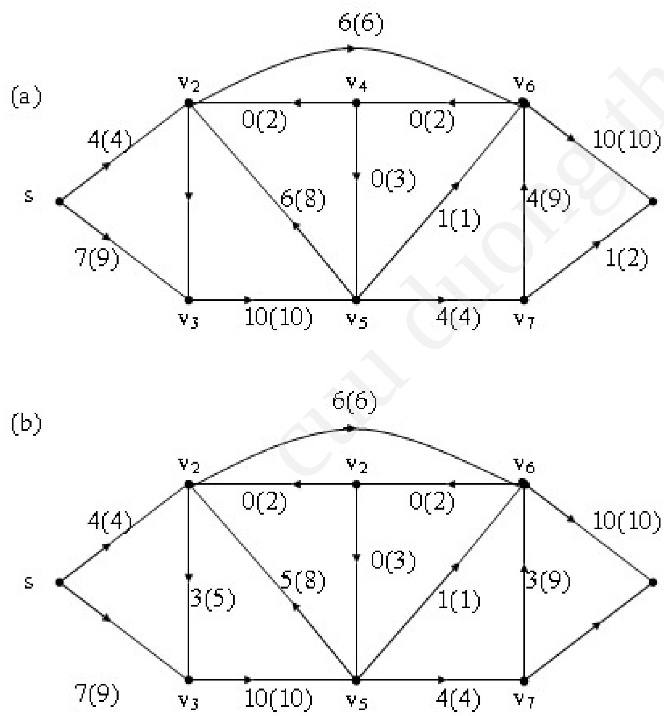
*** Thuật toán tăng luồng**

```
void Inc_Flow(){
    v=p[t]; u=t; tang=d[t];
    while (u!=s){
        if (v>0) f[v,u] = f[v,u] + tang;
        else{
            v = -v; f[u,v] = f[u,v] - tang;
        }
        u = v; v = p[u];
    }
}
```

*** Thuật toán tìm luồng cực đại**

```
void Max_Flow()
{         //luong ban dau la luong khong
          for (u ∈ V)
                    for (v ∈ V)  f(u,v)=0;
          stop=0;
          while (!stop)
                    if  (Find_Path())  Inc_Flow(); else stop=1;
          /*
          Lát cắt hẹp nhất là (VT, V\VT) với VT là các đỉnh đã thăm trong lần tìm đường tăng luồng cuối
          cùng mà không thành công và khi đó f là luồng cực đại
          */
}
```

Ví dụ . Hình a cho mạng G cùng với thông qua của tất cả các cung và luồng giá trị 10 trong nó. Hai số viết bên cạnh mỗi cung là khả năng thông qua của cung (số trong ngoặc) và luồng trên cung. Đường tăng luồng có dạng (s, v_3, v_2, v_6, v_7, t). Ta tính được d(t) = 1, giá trị luồng tăng từ 10 lên 11



Luồng trong hình (b) là cực đại.
Lát cắt hẹp nhất: X = {s, v_2, v_3, v_5},
X* ={v_4, v_6, v_7, t} . Giá trị luồng cực đại = 11.

* Nhận xét:
Giả sử khả năng thông qua của tất cả các cung là các số nguyên. Khi đó sau mỗi lần tăng luồng, giá trị luồng sẽ tăng lên ít nhất là 1. Từ đó suy ra thuật toán Ford_Fulkerson sẽ dừng sau không quá val(f*) lần tăng luồng và cho ta luồng cực đại trong mạng.
Tuy nhiên, nếu các khả năng thông qua là các số rất lớn thì giá trị của luồng cực đại cũng có thể là rất lớn và khi đó thuật toán mô tả ở trên sẽ đòi hỏi thực hiện rất nhiều bước tăng luồng.
Nếu các khả năng thông qua là các số vô tỉ, người ta còn xây dựng được ví dụ để cho thuật toán không dừng

5. Một số bài toán luồng tổng quát

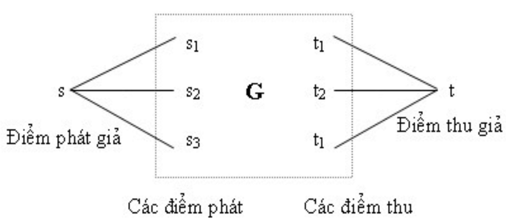
Trong phần này ta nêu ra một số dạng bài toán về luồng tổng quát mà việc giải chúng có thể dẫn về bài toán luồng cực đại trình bày ở trên.

5.1 Mạng với nhiều điểm phát và điểm thu.

Xét mạng G với p điểm phát s_1, s_2, . . ., sp và q điểm thu t_1, t_2,. . . , tq. Giả sử rằng luồng có thể đi từ một điểm phát bất kỳ đến tất cả các điểm thu. Bài toán tìm luồng cực đại từ các điểm phát đến các điểm thu có thể đưa về bài toán với một điểm phát và một điểm thu bằng cách đưa vào một điểm phát giả s và một điểm thu giả t và cạnh nối s với tất cả các điểm phát và cạnh nối các điểm thu với t.

Khả năng thông qua của cung nối s với điểm phát s_k sẽ bằng +vc nếu không có hạn chế về lượng phát của điểm phát s_k, và nếu lượng phát của s_k bị hạn chế bởi b_k thì cung (s, s_k) có khả năng thông qua là b_k. Đối với các cung nối t_k với điểm thu t, giả sử khả năng thông qua của (t_k, t) sẽ là giới hạn hoặc không giới hạn tuỳ theo lượng thu của điểm thu này có bị giới hạn hay không.

Trường hợp một số điểm thu chỉ nhận "hàng" từ một số điểm phát ta có bài toán nhiều luồng là một bài toán phức tạp hơn rất nhiều so với bài toán luồng cực đại giữa điểm phát s và điểm thu t.



Hình: Mạng với nhiều điểm phát và thu.

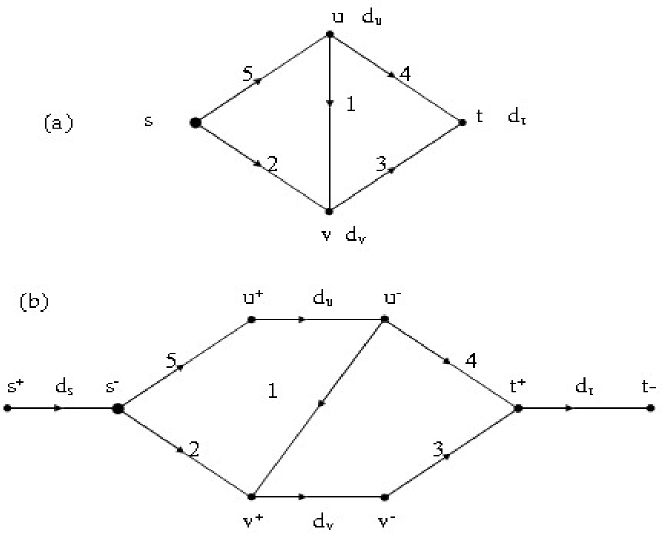
5.2 Bài toán với khả năng thông qua của các cung và các đỉnh.

Giả sử trong đồ thị G, ngoài khả năng thông qua của các cung c(u,v), ở mỗi đỉnh v còn có khả năng thông qua các đỉnh là d(v), và đòi hỏi tổng luồng đi vào đỉnh v không được vượt quá d(v), tức là.

$\sum_{w\in V} f(w,v) \le d(v)$. Tìm luồng cực đại giữa s và t trong mạng.

Xây dựng một mạng G' sao cho: mỗi đỉnh v của G tương ứng với 2 đỉnh v^+, v^- trong G', mỗi cung (u,v) trong G ứng với cung (u-,v^+) trong G', mỗi cung (v, e) trong G ứng với mỗi cung (v^-, w^+) trong G'. Ngoài ra, mỗi cung (v^+, v^-) trong G' có khả năng thông qua là d(v), tức là bằng khả năng thông qua của đỉnh v trong G.

Do luồng đi vào đỉnh v^+ phải đi qua cung (v^+, v^-) với khả năng thông qua d(v), nên luồng cực đại trong G' sẽ bằng luồng cực đại trong G với khả năng thông qua của các cung và các đỉnh.

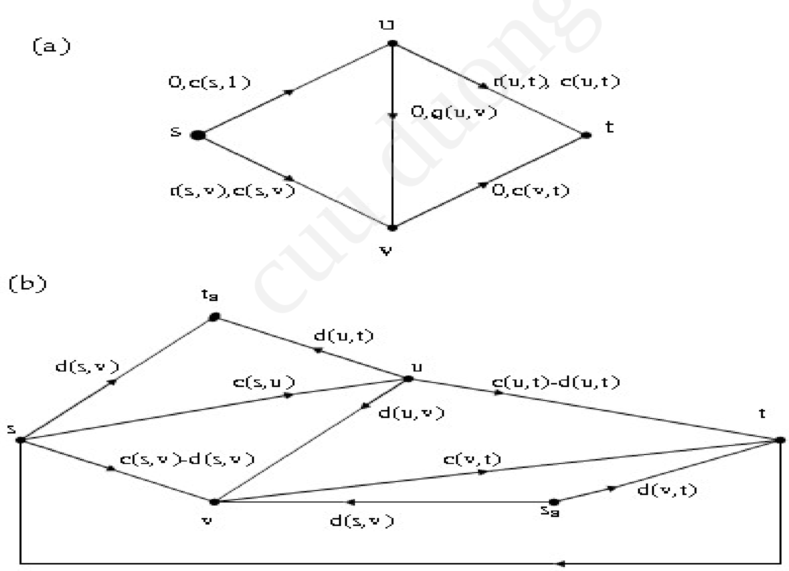


Hình a cho ví dụ mạng G với khả năng thông qua ở cung và đỉnh.

Hình b là mạng G' tương ứng chỉ có khả năng thông qua trên các cung.

5.3 Mạng trong đó khả năng thông qua của mỗi cung bị chặn hai phía.

Xét mạng G mà trong đó mỗi cung (u, v) có khả năng thông qua (cận trên của luồng trên cung) c(u,v) và cận dưới của luồng là d(u,v). Bài toán đặt ra là có tồn tại luồng f từ s đến t thoả mãn ràng buộc d(u,v) ≤ f(u,v) ≤ c(u,v)



Đưa vào mạng G đỉnh phát sa và đỉnh thu ta và xây dựng mạng Ga theo qui tắc:
Mỗi cung (u,v) mà d(u,v) □ 0 sẽ tương ứng với 2 cung (sa, v) va (u, ta) với khả năng thông qua là d(u,v). Giảm c(u,v) đi d(u,v) tức là thay khả năng thông qua của cung (u,v) bởi c(u,v) –d(u,v) còn cận dưới của nó đặt bằng 0. Ngoài ra thêm vào cung (t,s) với c(t,s) = □ .

Hình a cho ví dụ mạng G với khả năng thông qua của các cung bị chặn cả hai phía. Đồ thị Ga tương ứng được cho trong hình b.

Ký hiệu $d^* =$ □ d(u,v).

d(u,v)□ 0

* Định lý .

1) Nếu luồng lớn nhất trong mạng Ga từ sa đến ta bằng d^* thì tồn tại luồng tương thích trong G.

2) Nếu luồng lớn nhất trong mạng Ga từ sa đến ta là khác d^* thì không tồn tại luồng tương thích trong G.

6. Một số ứng dụng trong tổ hợp

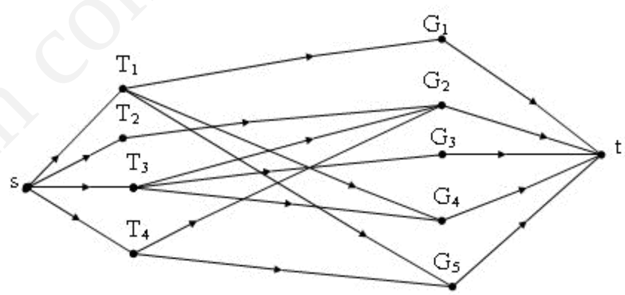
6.1 Bài toán đám cưới vùng quê.

Có m chàng trai ở một vùng quê nọ. Đối với mỗi chàng trai ta biết các cô gái mà anh ta vừa ý. Hỏi khi nào thì có thể tổ chức các đám cưới trong đó chàng trai nào cũng sánh duyên với các cô gái mà mình vừa ý.

Ta có thể xây dựng đồ thị với các đỉnh biểu thị các chàng trai và các cô gái, còn các cung biểu thị sự vừa ý của các chàng trai với các cô gái. Khi đó ta thu được một đồ thị hai phía.

Thí dụ. Có 4 chàng trai T_1, T_2, T_3,T4 và 5 cô gái G_1, G_2, G_3,G_4, G_5 . Sự vừa ý cho trong bảng sau

Chàng trai	Các cô gái mà chàng trai ưng ý
T_1	G_1, G_4, G_5
T_2	G_2
T_3	G_2, G_3,G_4
T_4	G_2, G_4



Đồ thị tương ứng được cho trong hình 7.

Đưa vào điểm phát s và điểm thu t. Nối s với tất cả các đỉnh biểu thị các chàng trai, và nối t với tất cả các đỉnh biểu thị các cô gái. Tất cả các cung của đồ thị đều có khả năng thông qua bằng 1. Bắt đầu từ luồng 0, ta tìm luồng cực đại trong mạng xây dựng được theo thuật toán Ford-Fulkerson. Từ định lý về tính nguyên, luồng trên các cung là các số hoặc 1. Rõ ràng là nếu luồng cực đại trong đồ thị có giá trị Vmax = m, thì bài toán có lời giải, và các cung với luồng bằng 1 sẽ chỉ ra cách tổ chức đám cưới thoả mãn điều kiện đặt ra. Ngược lại, nếu bài toán có lời giải thì Vmax = m. Bài toán về đám cưới vùng quê là một trường hợp riêng của bài toán về cặp ghép trên đồ thị hai phía mà để giải nó có thể xây dựng thuật toán hiệu quả hơn.

b) Bài toán về hệ thống đại diện chung.

Cho tập m phần tử $X=\{z_1, z_2, \ldots, z_m\}$. Giả sử $<A_1, A_2, \ldots, A_n>$ và $<B_1, B_2, \ldots, B_n>$ là hai dãy các tập con của X. Dãy gồm n phần tử khác nhau của X: $<a_1, a_2, \ldots, a_n>$ được gọi là hệ thống các đại diện chung của hai dãy đã cho nếu như tìm được một hoán vị σ của tập $\{1, 2, \ldots, n\}$ sao cho $<a_1, a_2, \ldots, a_n>$ là hệ thống các đại diện phân biệt của hai dãy $<A_1, A_2, \ldots, A_n>$ và $<B_{\sigma(1)}, B_{\sigma(2)}, \ldots, B_{\sigma(n)}>$, tức là điều kiện sau được thoả mãn: $a_i \in A_i \cap B_{\sigma(i)}$, i = 1, 2, . . ,n. Xây dựng mạng G=(V,E) với tập đỉnh

$$V = \{s, t\} \cup \{x_1, x_2, \ldots, x_n\} \cup \{u_1, u_2, \ldots, u_n\} \cup \{v_1, v_2, \ldots, v_n\} \cup \{y_1, y_2, \ldots, y_n\}.$$

trong đó đỉnh xi tương ứng với tập Ai, đỉnh yi tương ứng với tập Bi, các phần tử uj, yj tương ứng với phần tử zj. Tập các cung của mạng G được xác định như sau

$$E = \{(s,x_i): 1\le i\le n\} \cup \{(x_i,u_j): \text{với } z_j \in A_i, 1\le i\le n, 1\le j\le m\} \cup$$
$$\cup \{(u_j,v_j): 1\le j\le m\} \cup \{(v_j,y_i): \text{với } z_j \in B_i, 1\le i\le n, 1\le j\le m\} \cup$$
$$\cup \{(y_i, t): 1\le i\le n\}$$

Khả năng thông qua của tất cả các cung được đặt bằng 1. Dễ dàng thấy rằng hệ thống đại diện chung của hai dãy và tồn tại khi và chỉ khi trong mạng G=(V,E) tìm được luồng với giá trị n. Để xét sự tồn tại của luồng như vậy có thể sử dụng thuật toán tìm luồng cực đại từ s đến t trong mạng G=(V,E).

c) Về một bài toán tối ưu rời rạc.

Trong mục này ta sẽ trình bày thuật toán được xây dựng dựa trên thuật toán tìm luồng cực đại để giải một bài toán tối ưu rời rạc là mô hình toán học cho một số bài toán tối ưu tổ hợp.

Xét bài toán tối ưu rời rạc:

$$f(x_1,x_2,\ldots,x_n) = \max_{1\le i\le m} \sum_{j=1}^{n} x_{ij} \rightarrow \min \quad (1)$$

với điều kiện

$\sum_{j=1}^{n} a_{ij} x_{ij} = p_i, i=1, 2, \ldots, m$	(2)
$x_{ij}=0$ hoặc 1, $j=1, 2, \ldots, n$	(3)

trong đó $a_{ij} \in \{0,1\}$, i = 1, 2, . . . , m; j=1, 2, . . . n, p_i- nguyên dương, i = 1, 2, . . .,m.

Bài toán (1)-(3) là mô hình toán học cho nhiều bài toán tối ưu tổ hợp thực tế. Dưới đây ta dẫn ra một vài ví dụ điển hình.

- Bài toán phân nhóm sinh hoạt. Có m sinh viên và n nhóm sinh hoạt chuyên đề. Với mỗi sinh viên i, biết
- a_{ij} =1, nếu sinh viên i có nguyện vọng tham gia vào nhóm j,
- a_{ij} =0, nếu ngược lại,
- và pij là số lượng nhóm chuyên đề mà sinh viên i phải tham dự, i = 1, 2, . . . ,m; j=1, 2,. . . ,n.

Trong số các cách phân các sinh viên vào nhóm chuyên đề mà họ có nguyện vọng tham gia và đảm bảo mỗi sinh viên i phải tham gia đúng pi nhóm, hãy tìm cách phân phối với số người trong nhóm có nhiều sinh viên tham gia nhất là nhỏ nhất có thể được.

Đưa vào biến số

x_{ij}=1, nếu sinh viên i tham gia vào nhóm j,

x_{ij}=0, nếu ngược lại,

i=1, 2, . . .,m, j=1, 2,. . .,n, khi đó dễ thấy mô hình toán học cho bài toán đặt ra chính là bài toán (1)-(3).

Bài toán lập lịch cho hội nghị. Một hội nghị có m tiểu ban, mỗi tiểu ban cần sinh hoạt trong một ngày tại phòng họp phù hợp với nó. Có n phòng họp dành cho việc sinh hoạt của các tiểu ban. Biết

a_{ij} = 1, nếu phòng họp i là thích hợp với tiểu ban j,

aij=0, nếu ngược lại,

i=1, 2, . . .,m, j=1, 2,. . .,n. Hãy bố trí các phòng họp cho các tiểu ban sao cho hội nghị kết thúc sau ít ngày làm việc nhất.

Đưa vào biến số

x_{ij} = 1, nếu bố trí tiểu ban i làm việc ở phòng j,

x_{ij}=0, nếu ngược lại,

i=1, 2, . . .,m, j=1, 2,. . .,n, khi đó dễ thấy mô hình toán học cho bài toán đặt ra chính là bài toán (1)-(3), trong đó p_i=1, i=1, 2, . . .,m.

Bổ đề 2.

Bài toán (1)-(3) có phương án tối ưu khi và chỉ khi

$$\sum_{j=1}^{n} a_{ij} x_{ij} = p_i,\ i=1, 2,\ldots,m$$

x_{ij}=0 hoặc 1, j=1, 2,. . . ,n

Chứng minh. Điều kiện cần của bổ đề là hiển nhiên vì từ sự tồn tại phương án của bài toán suy ra các bất đẳng thức trong (4) được thực hiện ít nhất dưới dạng dấu đẳng thức. Để chứng minh điều kiện đủ, chỉ cần chỉ ra rằng nếu điều kiện (4) được thực hiện thì bài toán luôn có phương án. Thực vậy, giả sử điều kiện (4) được thực hiện. Khi đó nếu ký hiệu

I_{+i}=□ 1≤j≤n: a_{ij}=1□ ,

thì □ I_{+i}□ ≥ pi, i = 1, 2, . . .,m. Do đó nếu gọi

Ii □ I_{+i}, □ Ii□ =pi, i=1, 2,. . . ,m,

thì $X^* = (x^*_{ij})_{mxn}$ với các thành phần được xác định theo công thức

x^*_{ij} = 1, j□ Ii, x^*_{ij} =0, j□ Ii, i= 1, 2, . . .,m,

là phương án của bài toán (1)-(3). Bổ đề được chứng minh.

Do (4) là điều kiện cần để bài toán (1)-(3) có phương án, nên trong phần tiếp theo ta sẽ luôn giả thiết rằng điều kiện này được thực hiện.

Bây giờ ta chỉ ra rằng việc giải bài toán (1)-(3) có thể dẫn về việc giải một số hữu hạn bài toán luồng cực đại trong mạng. Trước hết, với mỗi số nguyên dương k, xây dựng mạng G(k) =(V,E) với tập đỉnh

$V= \{s\} \cup \{u_i :i=1, 2,. . . ,m\} \cup \{w_j : j=1, 2, . . .,n\} \cup \{t\}$

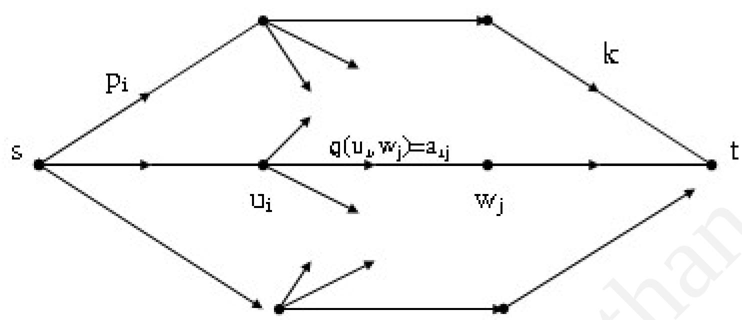
trong đó s là điểm phát, t là điểm thu, và tập cung

$E= \{(s,u_i):i=1, 2,. . .,m\} \cup \{(u_i,w_j) :i=1, 2,. . . ,m; j=1, 2, . . .,n\} \cup \{(w_j,t): j=1, 2, . . .,n\}$.

Mỗi cung $e \in E$ được gán với khả năng thông qua q(e) theo qui tắc sau:

- $q(s,u_i)=p_i$, i=1, 2,. . .m,
- $q(u_i,w_j)=a_{ij}$, i=1, 2,. . .m; j=1, 2, . . .,n;
- $q(w_j,t)=k$, j=1, 2, . . .,n.

Hình 8 chỉ ra cách xây dựng mạng G(k).



Hình 8. Mạng G(k).

Ký hiệu: $□ = \sum_{i=1}^{m} p_i$

Bổ đề sau đây cho thấy mối liên hệ giữa luồng cực đại trong mạng G(k) và phương án của bài toán (1)-(3).

Bổ đề 3.

Giả sử đối với số nguyên dương k nào đó, luồng cực đại nguyên trong mạng G(k) có giá trị là □ . Khi đó $X^* = (x^*_{ij})_{mxn}$ với các thành phần được xác định theo công thức

$x^*_{ij} = □^*(u_i,w_j)$, i=1, 2,. . .,m; j=1, 2,. . . ,n.

là phương án của bài toán (1)-(3).

Chứng minh. Thực vậy, do luồng cực đại trong mạng có giá trị □ và là luồng nguyên nên

$□^*(s,u_i)=p_i$, i=1, 2,. . .,m,

$□^*(u_i,w_j) \in \{0,1\}$, i=1, 2,. . .,m, j=1, 2,. . .,n,

từ đó suy ra

$$\sum_{j=1}^{n} x^*_{ij} = \sum_{j=1}^{n} \xi^*(u_i, w_j) = \xi^*(s, u_i) = p_i,\ i=1, 2, \ldots, m.$$

Vậy X^* là phương án của bài toán (1)-(3). Bổ đề được chứng minh.

Bổ đề 4.

Giả sử $X^* = (x^*_{ij})$ là phương án tối ưu và k^* là giá trị tối ưu của bài toán (1)-(3). Khi đó luồng cực đại trong mạng $G(k^*)$ có giá trị □ .

Chứng minh. Do giá trị của luồng cực đại trong mạng $G(k^*)$ không vượt quá □ , nên để chứng minh bổ đề ta chỉ cần chỉ ra luồng với giá trị □ trong mạng $G(k^*)$.

Xây dựng luồng □ * theo công thức sau:

□ * $(s,u_i) = p_i$, □ * $(u_i, w_j) = x^*_{ij},$

$$\xi^*(w_j, t) = \sum_{j=1}^{n} x^*_{ij},\ i=1, 2, \ldots, m;\ j=1, 2, \ldots, n.$$

Dễ dàng kiểm tra được □ * là luồng trong mạng G(m) có giá trị □ . Bổ đề được chứng minh.

Bổ đề 5. Nếu k=m thì luồng cực đại trong mạng G(m) có giá trị □ .

Chứng minh. Lập luận tương tự như trong Bổ đề 4, ta chỉ cần chỉ ra luồng với giá trị □ trong mạng G(m). Thực vậy, giả sử $X^* = (x^*_{ij})_{mxn}$ là phương án của bài toán (1)-(3) xây dựng theo công thức (5). Xây dựng luồng □ * theo công thức giống như trong chứng minh bổ đề 4, ta có luồng với giá trị □ . Bổ đề được chứng minh.

Từ bổ đề 3 và 4 suy ra việc giải bài toán (1)-(3) dẫn về việc tìm giá trị k^* nguyên dương nhỏ nhất sao cho luồng cực đại trong mạng $G(k^*)$ có giá trị □ . Bổ đề 5 cho ta thấy giá trị k^* □ [1,m]. Vì vậyđể giải bài toán (1)-(3) ta có thể áp dụng phương pháp tìm kiếm nhị phân trên đoạn [1,m] để tìm giá trị k^*, trong đó ở mỗi bước cần giải một bài toán luồng cực đại. Để giải bài toán tìm luồng cực đại trong mạng có thể sử dụng các thuật toán đa thức như đã nói ở trên. Từ đó suy ra kết quả sau

Định lý 5.

Bài toán (1)-(3) giải được nhờ thuật toán đa thức với độ phức tạp tính toán là $\log_2 m$.ONF, trong đó ONF là độ phức tạp tính toán của bài toán tìm luồng cực đại trong mạng G(k).

BÀI TẬP CHƯƠNG 7

Bài 1

Cho G=(V,E) đồ thị có hướng trong đó không có cung (s,t). Chứng minh rằng số đường đi cơ bản nối hai đỉnh s và t là bằng số ít nhất các đỉnh của đồ thị cần loại bỏ để trong đồ thị không còn đường đi nối s với t.

Bài 2

Xây dựng thuật toán tìm tập E_1 tất cả các cung của đồ thị mà việc tăng khả năng thông qua của bất kỳ cung nào trong E đều dẫn đến tăng giá trị của luồng cực đại trong mạng.

Bài 3

Cho hai dãy số nguyên dương $\{p_i,\ i=1,2,\dots,m\}$ và $\{q_j,\ j=1,2,\dots,n\}$. Hãy xây dựng ma trận $A=\{a_{ij} : i=1,2,\dots,m;\ j=1,2,\dots n\}$ với các phần tử a_{ij} □ $\{0,1\}$ có tổng các phần tử trên dòng i là p_i, tổng các phần tử trên cột j là q_j.

Bài 4

Có m chàng trai, n cô gái và k bà mối, Mỗi bà mối p (p=1,2,…,k) có một danh sách L_p một số chàng trai và cô gái trong số các chàng trai và cô gái nói trên là khách hàng của bà ta. Bà mối p có thể se duyên cho bất cứ cặp trai gái nào là khách hàng của bà ta, nhưng không đủ sức tổ chức quá d_p đám cưới. Hãy xây dựng thuật toán căn cứ vào danh sách L_p, d_p, p=1,2,…,k; đưa ra cách tổ chức nhiều nhất các đám cưới giữa m chàng trai và n cô gái với sự giúp đỡ của các bà mối.

Bài 5 : Chuyển bi

Cậu bé vẽ N (N<=100) vòng tròn, đánh số từ 1 tới N và tô màu các vòng tròn đó (có thể có các vòng tròn có màu giống nhau), sau đó nối từng cặp các cung định hướng, mỗi cung có một màu nhất định. Các màu (của cung và vòng tròn) được đánh số từ 1 đến 100.

Cậu bé chọn 3 số nguyên khác nhau L, K và Q nằm trong phạm vi từ 1 tới N, đặt vào trong các vòng tròn số L và K mỗi vòng một hòn bi, sau đó bắt đầu di chuyển bi theo nguyên tắc sau:

- Bi chỉ được chuyển theo cung có màu trùng với màu của vòng tròn chứa viên bi thứ 2.
- Bi chỉ được chuyển theo chiều cung
- Hai viên bi không được đồng thời ở cùng một vòng tròn;
- Không nhất thiết phải di chuyển lần lượt các viên bi,
- Quá trình di chuyển kết thúc, khi một trong hai viên bi tới vòng tròn Q.

Hãy lập trình xác định cách di chuyển để chấm dứt quá trình sau một số ít nhất các bước chuyển.

Dữ liệu vào từ file BL.INP:

- Dòng đầu: 4 số nguyên N L K Q
- Dòng thứ 2: N số nguyên $C_1, C_2,\dots,C_n$; C_i là màu vòng tròn i
- Dòng thứ 3: số nguyên M (0□ M□ 10000)
- M dòng sau: mỗi dòng 3 số nguyên A_i B_i D_i; xác định cung màu D_i từ vòng tròn A_i tới vòng tròn B_i.

Các số trên một dòng cách nhau một dấu cách.

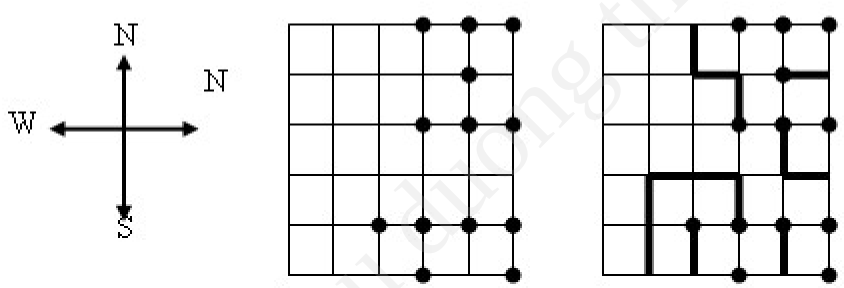
Kết quả đưa ra file BL.OUT:

- Dòng đầu: CO hoặc KHONG, cho biết quá trình có kết thúc được hay không,
- Nếu dòng đầu là CO thì dòng 2 chứa số nguyên xác định số bước chuyển tối thiểu .

BL.INP	BL.OUT
5 3 4 1 2 3 2 1 4 8 2 1 2 4 1 5 4 5 2 4 5 2 5 1 3 3 2 2 2 3 4 5 3 1 3 5 1	CO 3

Bài 6 : Bảng điện

Một lưới ô vuông được phủ trên một bảng điện hình vuông. Vị trí nằm trên giao của 2 đường kề của lưới sẽ được gọi là nút. Tất cả có nxn nút trên lưới.



Hình 2a. Bảng điện   Hình 2b. Lời giải

Có một số nút chứa tiếp điểm. Nhiệm vụ của bạn là cần nối các tiếp điểm với các nút ở trên biên của bảng bởi các đoạn dây dẫn (gọi là các mạch). Các mạch chỉ được chạy dọc theo các đường kẻ của lưới (nghĩa là không được chạy theo đường chéo). Hai mạch không được phép có điểm chung, vì vậy hai mạch bất kỳ không được phép cùng chạy qua cùng một đoạn đường kẻ của lưới cũng như không được chạy qua cùng một nút của lưới. Các mạch cũng không được chạy dọc theo các đoạn kẻ của lưới ở trên biên (mạch phải kết thúc khi nó gặp biên) và cũng không được chạy qua nút chứa tiếp điểm khác.

Ví dụ: Bảng điện và các tiếp điểm được cho trong hình 2a. Nút tô đậm trong hình vẽ thể hiện vị trí các tiếp điểm.

Yêu cầu: Viết chương trình cho phép nối được một số nhiều nhất các tiếp điểm với bbiên. Các tiếp điểm ở trên biên đã thỏa mãn đòi hỏi đặt ra, vì thế không nhất thiết phải thực hiện mạch nối chúng. Nếu như có nhiều lời giải thì chi cần đưa ra một trong số chúng.

Dữ liệu vào: file văn bản ELE.INP:

- Dòng đầu tiên chứa số nguyên n (3 <= n <= 15).
- Mỗi dòng trong số n dòng tiếp theo chứa n ký tự phân cách nhau bởi một dấu cách. Mỗi ký tự chỉ là 0 hoặc 1. Ký tự 1 thể hiện tiếp điểm, ký tự 0 thể hiện nút không có tiếp điểm trên vị trí tương ứng của lưới. Các nút được đánh số từ 1 đến n*n theo thứ tự từ trái sang phải, từ trên xuống dưới. Chỉ số của nút chứa tiếp điểm sẽ là chỉ số của tiếp điểm.

Kết quả: ghi ra file ELE.OUT:

- Dòng đầu tiên chứa k là số tiếp điểm lớn nhất có thể nối với biên bởi các mạch.
- Mỗi dòng trong số k dòng tiếp theo mô tả một mạch nối một trong số k tiếp điểm với biên theo qui cách sau: đầu tiên là chỉ số của tiếp điểm được nối, tiếp đến là dãy các ký tự mô tả hướng của mạch nối: E: đông, W: tây, N: bắc, S: nam. Giữa chỉ số và dãy ký tự phải có đúng một dấu cách, còn giữa các ký tự trong dãy ký tự không được có dấu cách.

Kết quả phải được đưa ra theo thứ tự tăng dần của chỉ số tiếp điểm.

Ví dụ:

ELE.IN	ELE.OUT
6 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1	11 E 16 NWN 17 SE 27 S 28 NWWSS 29 S

CÁC BÀI TẬP KHÁC

Bài 1:

Một khóa học gồm N môn học, môn học i phải học trong ti ngày. Giữa các môn học có mối quan hệ trước/sau: có môn học chỉ học được sau khi đã học một số môn học khác. Mối quan hệ đó được thể hiện bởi một mảng hai chiều A[i, j];

i, j = 1, …, N trong đó A[i, j] = 1/0 và A[i, i] bằng 0 với mọi i, A[i, j] = 1 khi và chỉ khi môn học i phải được dạy xong trước khi học môn j (ngày kết thúc môn i phải trứơc ngày bắt đầu môn j). Môn học i phải dạy trước môn học j nếu có một dãy môn học i_1, i_2, …, ik sao cho a[it, it_{+1}] = 1, 1 <= t <= k-1, i_1=i và ik=j. Nếu có một nhóm các môn học từng đôi một không có quan hệ trước/sau thì trong mỗi ngày, về nguyên tắc, ta có thể học đồng thời tất cả những môn học này (nếu không vi phạm quan hệ với các môn học khác). Mảng A[i, j] được gọi là bế tắc nếu có một dãy các môn học i_1, i_2,…, ik, k > 1, mà môn i_1 phải dạy trước môn i_2, môn i_2 phải dạy trước môn i_3, …, môn ik_{-1} phải dạy trước môn ik, môn ik phải dạy trước môn i_1.

Hãy viết chương trình với tên KT3.PAS làm các việc sau:

1. Hãy xét xem mảng A có bế tắc hay không.
2. Nếu mảng A không bế tắc, hãy tính xem khóa học có thể kết thúc trong thời gian nhanh nhất là bao nhiêu ngày.
3. Theo các học bảo đảm thời gian hoàn thành ngắn nhất ở câu 2, hãy tính xem một học sinh trong quá trình học phải học đồng thời trong một ngày nhiều nhất bao nhiêu môn.

Dữ liệu vào được cho bởi file text có tên MH.DAT trong đó số N ghi ở dòng thứ nhất, trong nhóm N dòng tiếp theo, dòng thứ i ghi N số A[i, 1], …, A[i, N] dòng cuối cùng ghi N số nguyên dượng ti không lớn hơn 30, 1 <= i <= N; N <= 30.

▪Kết quả ghi ra file TKB.DAT như sau: dòng thứ nhất ghi số 1/0 tùy theo mảng A bế tắc / không bế tắc. Nếu dòng thứ nhất ghi số 0, ta mới ghi tiếp kết quả câu 2 và 3.

▪Kết quả câu 2 ghi tiếp vào file TKB.DAT N+1 dòng như sau: dòng dầu ghi số T là số ngày tối thiểu có thể hoàn thành khóa học, tiếp theo là N dòng trong đó dòng thứ i ghi 2 số X, Y với ý nghĩa môn học thứ i học từ ngày thứ X đến ngày thứ Y (chú ý rằng Y – X = ti – 1).

▪Kết quả câu 3 ghi tiếp vào file TKB.DAT như sau: dòng thứ nhất ghi 2 số Z, W với ý nghĩa trong ngày Z phải học W môn (W là số nhiều nhất các môn học phải học đồng thời trong một ngày), tiếp theo là một dòng ghi tên các môn học phải học đồng thời trong ngày Z.

Trong các câu 2 và 3, có thể có nhiều lời giải tương đương chỉ cần đưa ra một lời giải.

Ví dụ 1

MH.DAT	TKB.DAT

4 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 1 1 1	1

Ví dụ 2

MH.DAT	TKB.DAT
7 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 2 2 8 4 10 2 3	0 22 1 2 3 4 1 8 12 13 22 13 14 15 17 1 2 1 3

Bài 2:

Giám đốc một công ty quyết định tổ chức buổi tiệc trà gặp gỡ toàn thể nhân viên trong công ty. Công ty được tổ chức theo mô hình phân cấp lãnh đạo và mối quan hệ thủ trưởng – nhân viên tạo thành cây có gốc là giám đốc. Để đảm bảo không khí tự nhiên, giám đốc quyết định không để thủ trưởng và nhân viên dưới quyền ngồi cùng một bàn. P gọi là thủ trưởng của Q, nếu P là thủ trưởng trực tiếp của Q hoặc tồn tại dãy P_1, P_2, …, Pk (1 < k), sao cho P = P_1, Q = Pk và Pi là thủ trưởng trực tiếp của Pi_{+1} (i = 1, 2, … , k -1). Tất cả mọi người trong công ty được đánh số từ 1 đến N (N là tổng số người trong công ty với giám đốc bắt đầu từ 1).

Yêu cầu: tính số lượng bàn ít nhất cần thiết để có thể bố trí cho mọi người ngồi theo yêu cầu nêu trên và cho một phương án bố trí người ở mỗi bàn.

Dữ liệu vào: file text COMPANY.INP, dòng đầu tiên là số nguyên m – số ghế tối đa cho một bàn, dòng thứ 2 – số nguyên N – số người trong công ty, dòng thứ ba (và các dòng sau nếu cần) là dãy số nguyên, các số cách nhau ít nhất một dấu cách hoặc nhóm ký tự xuống dòng, số nguyên thứ i trong dãy cho biết ai là thủ trưởng trực tiếp của nhân viên i. Giám đốc không có thủ trưởng nên số này bằng 0.

2 <= m <= 10, 1 <= N <= 200.

Kết quả: đưa ra file text COMPANY.OUT, dòng đầu tiên là số bàn ít nhất cần thiết, các dòng sau mỗi dòng tương ứng với một bàn và chứa dãy số nguyên xác định ai được bố trí ngồi sau bàn đó.

Ví dụ:

COMPANY.INP

4

13

0 1 9 9 9 2 2 1 1 7 8 8 10

File kết quả COMPANY.OUT có thể có nội dung:

5

13 3 4 5

10 6 8

7 9 11 12

2

1

Bài 3:

Trên bàn cờ NxN, hãy tìm cách sắp xếp số lượng tối đa các con hậu sao cho không con nào có thể ăn con nào.

Bài 4:

Cho 1 đồ thị có hướng G, hãy tìm một tập hợp X_0 ít nhất các đỉnh của G sao cho mọi đỉnh i của G hoặc thuộc X_0 hoặc i nối trựctiếp với định j thuộc X_0. Làm bài 14 trong trường hợp G vô hướng.

Bài 5:

Trên bàn cờ NxN, hãy tìm cách sắp xếp số lượng tối thiểu các con hậu sao cho mọi ô cờ trên bàn cờ bị chiếu bởi ít nhất 1 con.

Bài 6:

Một ký túc xá nuôi 15 cô gái. Hàng ngày các cô đi chơi với nhau theo bộ 3. Hỏi có thể đưa các cô đi chơi trong tối đa bao nhiêu ngày để không có 2 cô nào đi chung trong một bộ 3 quá 1 lần.

Hãy tổng quát hóa bài toán.

Bài 7:

Trong 1 trại giam ở thành phố A, mỗi nhà giam có một trạm gác độc lập, nhưng người đứng gác, chẳng hạn ở nhà giam x_0, cũng có thể thấy những gì xảy ra ở các trại giam x_1, x_2, x_3… khác do các nhà giam này thông với x_0 bởi 1 hành lang thẳng. Giả sử biết các thông tin trên, hãy xác định số lượng tối thiểu các lính canh để có thể quan sát được mọi nhà giam.

Bài 8:

Một số hải cảng x_1, x_2, x_3… có các mặt hàng mà các hải cảng y_1, y_2, y_3…cần đến. Lượng hàng có ở xi là si và yêu cầu hàng hóa của yi là di. Nếu có tàu đi từ xi tới yj thì ta ký hiệu cij là tổng lượng hàng mà các tàu có thể vận chuyển từ xi tới yj. Vậy có thể thỏa mãn mọi yêu cầu không? Tổ chức vận chuyển ra sao? Hãy viết chương trình giải quyết bài toán trên.

Bài 9:

Trong một cuộc du lịch, m gia đình phân nhau đi trên n xe. Các gia đình tương ứng có r_1, r_2, …, rm người và các xe tương ứng có s_1, s_2, …, sn chỗ ngồi. Hãy tìm cách phân phối sao cho 2 người cùng gia đình không ngồi chung một xe hoặc cho biết không thể làm như vậy.

Bài 10:

Trong một trường trung học, mỗi học sinh nữ có m bạn nam và mỗi học sinh nam có m bạn nữ. Hãy chỉ ra cách sắp xếp để mỗi cô gái có thể lần lượt khiêu vũ với các bạn trai của mình và các chàng trai có thể lần lượt khiêu vũ với các bạn gái của mình.

Bài 11:

Một nhà in phải sản xuất n cuốn sách bằng 2 máy: một để in, một để đóng sách. Gọi ak là thời gian cần cho việc in cuốn thứ k và bk là thời gian cần cho việc đóng cuốn đó. Tất nhiên là sách phải in xong mới đòng, do đó máy đóng có thể phải chờ đợi lâu hay chóng. Vậy tiến hành theo thứ tự nào để có thể xong việc sớm nhất.

Bài 12: Ổn định

Trong một lớp có N dãy bàn và mỗi dãy có M chỗ ngồi. Trong lớp có K cán sự lớp. Mỗi một cán sự cầm một đề bài tập. Các cán sự này có nhiệm vụ chuyển đề bài tập đến các học sinh khác ngồi kề mình ở phía trước, sau, trái và phải. Sau khi các cán sự làm xong công việc của mình, mỗi học sinh thông báo số lượng đề bài tập mình đã nhận được. Dựa trên thông tin này hãy xác định vị trí của các cán sự trong lớp.

Bài 13:

Có một máy thu và một máy phát tín hiệu. Giả sử máy phát có thể phát đi 5 loại tín hiệu khác nhau a, b, c, d, e. Ở máy thu mỗi tín hiệu có thể được hiểu theo 2 cách khác nhau: tín hiệu a có thể hiểu p hay q, tín hiệu b có thể hiểu q hay r, ... Số cực đại các tín hiệu mà ta có thể sử dụng là bao nhiêu để cho ở máy thu không xảy ra nhầm lẫn giữa các tín hiệu được sử dụng.

Bài 14:

Cho 1 đồ thị vô hướng G. Người ta muốn tô các đỉnh của G bằng 2 màu đen, trắng sao cho 2 đỉnh kề nhau không được tô bởi cùng 1 màu. Hãy cho biết có thể làm được điều này không. Nếu được hãy chỉ ra cách tô.

Bài 15:

Cho một đồ thị không định hướng trong một tệp văn bản với cách mã hóa như sau

Dòng đầu là số đỉnh (n). Các đỉnh được xem đánh số liên tiếp từ 1 đến n.

Mỗi dòng sau là mô tả một cạnh cho bằng chỉ số 2 đỉnh là đầu mút của cạnh đó.

Yêu cầu tô màu một số đỉnh thành mà đỏ sao cho số đỉnh đỏ là lớn nhất nhưng không có cạnh nào được phép có cả hai đầu mút là các đỉnh đỏ. Kết quả cho ra trên hai dòng. Dòng thứ nhất là số đỉnh màu đỏ, dòng thứ hai là số thứ tự các đỉnh có màu đỏ.

Bài 16:

Giám đốc một công ty quyết định tổ chức buổi tiệc trà gặp gỡ toàn thể nhân viên trong công ty. Công ty được tổ chức theo mô hình phân cấp lãnh đạo và mối quan hệ thủ trưởng – nhân viên tạo thành cây có gốc là giám đốc. Để đảm bảo không khí tự nhiên, giám đốc quyết định không để thủ trưởng và nhân viên dưới quyền ngồi cùng một bàn. P gọi là thủ trưởng của Q, nếu là P thủ trưởng trực tiếp của Q hoặc tồn tại dãy P1, P2, ..., Pk (1 < k), sao cho P = P1, Q = Pk và Pi là thủ trưởng trực tiếp của Pi+1 (i = 1,2, ..., k-1). Tất cả mọi người trong công ty được đánh số từ 1 đến N – tổng số người trong công ty, bắt đầu từ giám đốc với số là 1.

Yêu cầu tính số lượng bàn ít nhất cần thiết để có thể bố trí cho mọi người ngồi theo yêu cầu nêu trên và cho một phương án bố trí người ở mỗi bàn.

- Dữ liệu vào : File ASCII COMPANY.INP, dòng đầu tiên là số nguyên m – số ghế tối đa cho một bàn, dòng thứ 2 – số nguyên N – số người trong công ty, dòng thứ 3 (và các dòng sau nếu cần) là dãy số nguyên, các số cách nhau ít nhất một dấu cách hoặc nhóm ký tự xuống dòng, số nguyên thứ i trong dãy cho biết ai là thủ trưởng trực tiếp của nhân viên i. Giám đốc không có thủ trưởng nên số này bằng 0. 2 <= m <= 10, 1 <= N <= 200.
- Kết quả : đưa ra file ASCII COMPANY.OUT, dòng đầu tiên là số bàn ít nhất cần thiết, các dòng sau mỗi dòng tương ứng với một bàn và chứa dãy số nguyên xác định ai được bố trí ngồi sau bàn đó.
- Ví dụ :

COMPANY.INP

4

13

0 1 9 9 9 2 2 1 1 7 8 8 10

File kết quả COMPANY.OUT có thể có nội dung :

5

13 3 4 5

10 6 8

7 9 11 12

2

1

Bài 17:

Một hệ thống có N (N<=50) thành phố (được đánh số từ 1 đến N) được nối với nhau bằng một hệ thống giao thông được cho bằng các đường nối trực tiếp các cặp hai thành phố: mỗi đoạn nối trực tiếp có độ dài là 1 và là đường đi hai chiều. Hệ thống này được gọi là "chẵn-lẻ" nếu như khi hai thành phố được nối với nhau (có thể qua các thành phố khác) bởi một đường đi với độ dài chẵn thì luôn tìm được đường đi có độ dài lẻ nối hai thành phố đó. Các bài toán được đặt ra :

Câu 1 : Với một hệ thống giao thông đã cho, kiểm tra xem nó có tính "chẵn-lẻ" hay không ?

Câu 2 : Nếu câu trả lời của câu 1 là phủ định thì hãy chỉ ra tập con X trong hệ thống các thành phố nói trên và có nhiều nhất các thành phố mà đảm bảo điều kiện sau: Nếu có đường đi nối hai thành phố bất kỳ trong X thì độ dài đường đi đó phải chẵn (giữ nguyên hệ thống giao thông).

Điều kiện kỹ thuật :

File dữ liệu vào tên CHANLE.INP:

- Dòng đầu tiên chứa giá trị N.
- Các dòng tiếp theo chứa lượng chẵn các số nguyên không âm mà hai số nguyên cuối cùng là 0, còn mọi số nguyên khác chỉ là chỉ số thành phố; từng cặp hai số nguyên sẽ chỉ ra số hiệu c3a hai thành phố được nối trực tiếp với nhau.

File kết quả ra có tên CHANLE.OUT:

- Dòng đầu có số 0 hoặc 1 cho câu 1, trong đó 1 chỉ ra rằng hệ thống nói trên có tính chẵn lẻ, 0 phủ định.
- Nếu có kết quả phủ định thì các dòng tiếp theo có dạng :

+ Dòng thứ 2 có số lượng phần tử của X.

+ Các dòng tiếp theo (từ dòng thứ 3 trở đi) chứa số hiệu các thành phố có trong X.

Ví dụ:

CHANLE.INP có nội dung :

4

1 2 1 3 0 0

CHANLE.OUT

0

3

2 3 4

Nếu file CHANLE.INP có nội dung :

5

1 2 2 3

3 4 4 5 5 1

0 0

CHANLE.OUT

1