ÔN TẬP - LÝ THUYẾT ĐỒ THỊ

Bài 1: XÂY DỰNG TRUNG TÂM

Có N địa điểm dân cư đánh số từ 1 đến N. Giữa M cặp địa điểm trong số N địa điểm nói trên có tuyến đường nối chúng. Cần xây dựng một trung tâm dịch vụ tổng hợp tại một địa điểm trùng với một địa điểm dân cư, sao cho tổng khoảng cách từ trung tâm dịch vụ đến N địa điểm dân cư là nhỏ nhất. Ta gọi khoảng cách giữa hai địa điểm là độ dài đường đi ngắn nhất nối chúng. Giả sử N địa điểm trên là liên thông với nhau. Nếu có nhiều phương án thì đưa ra phương án đặt trung tâm dịch vụ tại địa điểm có số hiệu nhỏ nhất.

Dữ liệu vào: Từ tệp **TRUNGTAM.INP** gồm M+1 dòng:

- Dòng 1: Chứa hai số nguyên dương N và M (N ≤ 100);
- Dòng i+1 (1 ≤ i ≤ M): Chứa 3 số nguyên dương x, y, z, ở đó hai số đầu x, y là số hiệu của hai địa điểm dân cư được nối với nhau bởi tuyến đường này, còn số thứ ba z (≤ 32767) là độ dài của tuyến đường này.

Dữ liệu ra: Ghi ra tệp **TRUNGTAM.OUT** gồm 2 dòng:

- Dòng 1: Ghi vị trí trung tâm dịch vụ;
- Dòng 2: Ghi tổng khoảng cách từ trung tâm dịch vụ đến các địa điểm dân cư.

Ví dụ:

TRUNGTAM.INP	TRUNGTAM.OUT
5 7	3
1 2 9	15
2 3 4	
1 4 2	
4 5 5	
5 3 1	
5 1 5	
3 1 4	

Bài 2: Sống trong một thành phố lớn có mật độ dân số cao và giao thông phức tạp, ông Bình rất lo về nạn kẹt xe ảnh hưởng đến cuộc sống của mình. Mỗi sáng thức dậy công việc đầu tiên của ông Bình là nghe bản tin giao thông của thành phố. Ông Bình rất lo là con đường quen thuộc hàng ngày đi đến cơ quan bị kẹt xe thì ông không biết phải chọn con đường nào để thay thế sao cho ông có thể đến cơ quan sớm nhất có thể.

Bản đồ thành phố là gồm có N nút giao thông và M con đường hai chiều nối các nút giao thông này. Độ dài của mỗi con đường là một số nguyên dương. Nhà ông Binh ở nút giao thông 1, cơ quan ông bình ở nút giao thông N. Để thuận tiện cho việc đi lại,

ông Bình đang tìm hiểu các con đường ngắn nhất dẫn từ nhà ông Bình đến cơ quan. Bạn hãy giúp ông Bình thực hiện công việc kho khăn này.

Input:

Dòng thứ nhất ghi hai số nguyên N và M $(1 \le N \le 5000, 1 \le M \le 20000)$

M dòng tiếp theo, mỗi dòng ghi 3 số nguyên dương U, V, L (con đường nối U đến V với độ dài L, L \leq 32000).

Output: Ghi hai số là độ dài đường đi ngắn nhấn và số lượng đường đi ngắn nhất.

Subtask 1(2 điểm): $n \le 20$

Subtask 2 (2 điểm): $n \le 100, m \le 1000$

Subtask 3 (3 điểm): $n \le 5000$, $m \le 20000$

Ví dụ:

Input	Output
3 3	4 2
1 2 3	
1 3 4	
2 3 1	

Bài 3: Guide.

Hòa và An là một cặp hướng dẫn viên du lịch, Hòa rất thành thạo Tiếng Anh để giới thiệu về danh lam thắng cảnh, còn An biết rất rõ các địa điểm tham quan du lịch ở Việt Nam. Công việc của họ là kết hợp với nhau để hướng dẫn một vài tua du lịch cho du khách nước ngoài muốn tham quan từ thành phố này đến thành phố khác trong nước.

Trên **N** thành phố, có **R** con đường hai chiều được nỗi giữa chúng, giữa hai thành phố bất kỳ đều có đường kết nối và có dịch vụ xe buýt chỉ chạy giữa hai thành phố này và chạy theo đường nối trực tiếp giữa chúng. Mỗi dịch vụ xe buýt đều có một giới hạn lớn nhất lượng khách mà xe buýt có thể chở được. An có một tấm bản đồ chỉ các thành phố và những con đường nối giữa chúng. Ngoài ra, An cũng có thông tin về mỗi dịch vụ xe buýt giữa các thành phố. An hiểu rằng An không thể đưa tất cả các khách du lịch đến thành phố thăm quan trong cùng một chuyến đi.

Nhưng, An và Hòa nhận thấy thật khó để tìm ra tất cả lộ trình tốt nhất sao cho để họ có thể đưa tất cả khách du lịch đến thành phố thăm quan với số chuyến đi là nhỏ nhất. Do vậy mà họ cần sự trợ giúp của các bạn.

Dữ liệu vào: Cho từ tệp GUIDE.INP gồm:

- Dòng đầu chứa hai số nguyên N (N \leq 100) và R mô tả lần lượt số thành phố và số đường đi giữa các thành phố.
- R dòng tiếp theo, mỗi dòng chứa 3 số nguyên: C1, C2, P. Trong đó C1, C2 mô tả lộ trình đường đi từ thành phố C1 đến thành phố C2 và P ($2 <= P <= 10^6$) là giới hạn lớn nhất có thể phục vụ của dịch vụ xe buýt giữa hai thành phố.
- Các thành phố được đánh dấu bằng một số nguyên từ 1 đến N. Dòng thứ (R+1) chứa ba số nguyên S, D, T mô tả lần lượt thành phố khởi hành, thành phố cần đến và số khách du lịch được phục vụ $(2 < T <= 10^6)$.

Dữ liệu ra: Đưa ra file **GUIDE.OUT** gồm Ghi ra số lộ trình nhỏ nhất cần phải đi qua các thành phố thỏa mãn yêu cầu đề bài.

Ví du:

CHIDE IND	CHIDE OUT	Chú thíob
GUIDE.INP	GUIDE.OUT	Chú thích
7 10	5	Lộ trình đi : $1 - 2 - 4 - 7$
1 2 30		(2) 60 (5)
1 3 15		25.
1 4 10		30/
2 4 25		10 10
2 5 60		$\begin{array}{c} \bullet \\ \bullet $
3 4 40		40 15
3 6 20		34/
4 7 35		3 20 6
5 7 20		
6 7 30		35
1 7 100		

------HÉT-----

- Lưu ý: Bài chấm theo từng bộ test được lưu trong đĩa CD.

CODE THAM KHẢO

```
Bài 1:
Program bangtinbongda;
Const
        inp='BANGDIEM.inp';
        out='BANGDIEM.OUT';
        a: array[1..10000,1..10000] of LONGINT;
var
        s,t,h: array[1..10000] of LONGINT;
        i,j,n,m,max: LONGINT;
        fi,fo:Text;
Procedure docfile xuly;
Var i,j:LONGINT;ss:LONGINT;
Begin
     assign(fi,inp);reset(fi);
     readln(fi,n,m);
     fillchar(s,sizeof(s),0);
     fillchar(t,sizeof(t),0);
     fillchar(h, sizeof(h),0);
     for i:=1 to n do
     begin
     for j:=1 to m do
          begin
          read(fi,a[i,j]);
          s[i]:=s[i]+a[i,j];
          if a[i,j]=3 then t[i]:=t[i]+1;
          if a[i,j]=1 then h[i]:=h[i]+1;
          end;
     readln(fi);
     end;
     close(fi);
     \max:=s[1];
     for i:=2 to n do if max<s[i] then max:=s[i];</pre>
Procedure ghifile;
Var i,j: LONGINT;
Begin
     assign(fo,out);rewrite(fo);
     for i:=1 to n do
     writeln(fo,t[i],' ',h[i],' ',m-(t[i]+h[i]),' ',s[i]);
     for i:=1 to n do if s[i]=max then write(fo,i,' ');
     close(fo);
end;
BEGIN
     docfile_xuly;
     ghifile;
```

END.

Test0 0,5 0,5 Test1 Test2 0,5 Test3 0,5 0,5 Test4 Test5 0,5 0,5 Test6 Test7 0,5 0,5 Test8 Test9 0,5

```
Bài 2:
const fi='SNLN.INP';
    fo='SNLN.OUT';
var x,kq:string;
procedure doc;
begin
    assign(input,fi); reset(input); read(x); close(input);
end;
```

```
procedure hd(var a,b:char);
var tg:char;
begin
        tg:=a; a:=b; b:=tg;
end;
procedure xuly;
var k,i,j,n:byte; TG:CHAR;
begin
        n:=length(x);
        for k:=n downto 1 do if x[k] < x[k-1] then break;
        if k=1 then kq:='0'
        else
                begin
                         for i:=n downto k do if x[i] < x[k-1] then
                                 begin
                                          hd(x[i],x[k-1]);
                                          break;
                                 end;
                         for j:=1 to n-k do
                                 for i:=k to n-j do
                                          if x[i] < x[i+1] then hd(x[i],x[i+1]);
            kq := x;
                 end;
end;
procedure inkq;
begin
        assign(output,fo); rewrite(output); write(kq); close(output);
end;
BEGIN
DOC;
XULY;
INKQ;
END.
```

<u>Test0</u>	0,5
<u>Test1</u>	0,5
Test2	0,5
Test3	0,5
<u>Test4</u>	0,5
Test5	0,5
<u>Test6</u>	0,5
Test7	0,5
Test8	0,5
Test9	0,5

```
Bài 3: const fi='
```

```
fi='NENSO.INP';
      fo='NENSO.OUT';
        nmax=1000000000;
type
        data=longint;
var
        f,f2:text;
        n:string;
        test:data;
procedure xuli;
var
        i,s:data;
begin
        s:=0;
        for i:=1 to length(n) do
                s:=s+ord(n[i])-48;
        s:=s \mod 9;
        if s<>0 then
```

```
writeln(f2,s)
        else
                writeln(f2,9);
end;
procedure nhap;
var
        i:data;
begin
        assign(f,fi); reset(f);
      assign(f2,fo);rewrite(f2);
        readln(f,test);
        for i:=1 to test do
                begin
                         readln(f,n);
                         xuli;
                 end;
end;
begin
        nhap;
      close(f2);
```

end.

Test0 0,5 Test1 0,5 0,5 Test2 Test3 0,5 0,5 Test4 0,5 Test5 Test6 0,5 0,5 Test7 0,5 Test8 0,5 Test9

```
Bài 4:
Program BPrimes;
 n,res :LongInt;
 h :Byte;
 Free :Array[1..1000000] of Boolean;
 procedure Eratosthenes;
  var
    i,j :LongInt;
 begin
    FillChar(Free,n,true); Free[1]:=false;
    for i:=1 to Trunc(Sqrt(n)) do
      if (Free[i]) then
        begin
          j:=i*i;
          while (j<=n) do
            begin
              Free[j]:=false; Inc(j,i);
            end;
       end;
  end;
  function Check(i :LongInt) :Boolean;
    j :Byte;
  begin
    j:=0;
    while (j \le h) and (i > 0) do
      begin
        Inc(j,i mod 2); i:=i div 2;
```

```
end;
    if (j=h) then Exit(true) else Exit(false);
  procedure Solve;
  var
    i :LongInt;
  begin
    res:=0;
    for i:=1 to n do
      if (Free[i]) and (Check(i)) then Inc(res);
  end;
Begin
  Assign(Input, 'CAUDO.INP'); Reset(Input);
  Assign(Output, 'CAUDO.OUT'); Rewrite(Output);
  ReadLn(n,h);
  Eratosthenes;
  Solve;
  Write(res);
  Close(Input); Close(Output);
```

End.

<u>Test0</u>	0,5
Test1	0,5
<u>Test2</u>	0,5
<u>Test3</u>	0,5
<u>Test4</u>	0,5
<u>Test5</u>	0,5
<u>Test6</u>	0,5
<u>Test7</u>	0,5
<u>Test8</u>	0,5
Test9	0,5

```
Bài 5:
Program Cho_trung_tam;
Uses crt;
Const
 FI = 'TRUNGTAM.INP';
 FO = 'TRUNGTAM.OUT';
 MAX N = 100;
 MAX_VALUE = 999999999;
 n, dia diem, min : longint;
 d : array[1..MAX N, 1..MAX N] of longint;
 f : text;
Procedure Doc;
Var i, j, k, m : integer;
Begin
  assign(f, FI); reset(f);
  read(f, n, m);
  for i := 1 to n do
    begin
      d[i, i] := 0;
      for j := i+1 to n do
        begin
          d[i, j] := MAX_VALUE;
          d[j, i] := MAX_VALUE;
        end;
    end;
  for k := 1 to m do
    begin
      read(f, i, j);
```

```
read(f, d[i, j]);
      d[j, i] := d[i, j];
    end;
  close(f);
End;
Procedure Floyd;
Var sum, i, j, k : longint;
Begin
  for k := 1 to n do
    for i := 1 to n do
      for j := 1 to n do
        if d[i, j] > d[i, k] + d[k, j] then d[i, j] := d[i, k] + d[k, j];
  min := MAX VALUE;
  for i := 1 to n do
   begin
      sum := 0;
      for j := 1 to n do sum := sum + d[i, j];
      if sum < min then
        begin
          dia diem := i;
          min := sum;
        end;
    end;
End;
Procedure Ghi;
Begin
 assign(f, FO); rewrite(f);
 writeln(f, dia_diem);
  write(f, min);
  close(f);
End;
Begin
 Doc;
 Floyd;
  Ghi;
End.
```

<u>Test0</u>	0,5
<u>Test1</u>	0,5
Test2	0,5
<u>Test3</u>	0,5
<u>Test4</u>	0,5
<u>Test5</u>	0,5
Test6	0,5
Test7	0,5
Test8	0,5
Test9	0,5