```
unit rules350;

 interface

  uses
    globals,cmmnds1,pusherr,pushStack,ruleAtoF;



    procedure r301; procedure r302; procedure r303; procedure
r304; procedure r305;
     procedure r306; procedure r307; procedure r308; procedure
r309; procedure r310;
     procedure r311; procedure r312; procedure r313; procedure
r314; procedure r315;
     procedure r316; procedure r317; procedure r318; procedure
r319; procedure r320;
     procedure r321; procedure r322; procedure r323; procedure
r324; procedure r325;
     procedure r326; procedure r327; procedure r328; procedure
r329; procedure r330;
     procedure r331; procedure r332; procedure r333; procedure
r334; procedure r335;
     procedure r336; procedure r337; procedure r338; procedure
r339; procedure r340;
     procedure r341; procedure r342; procedure r343; procedure
r344; procedure r345;
     procedure r346; procedure r347; procedure r348; procedure
r349; procedure r350;



implementation



procedure r301;
(*********************************************)
(*                                           *)
(*    if cubic,nconn>=2,plnar,not hamil,     *)
(*       bipartite then p >= 26              *)
(*                                           *)
(*********************************************)
begin
  if (activerule[301]) and (min[nodes] < 26) and (min[hamil] = 0)
     and (max[plnar] = 1) and (max[bipart] = 1) and
(max[nconn] >= 2) then
    begin
      rule:='301/ ';
      if (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) and
```

```
(min[nconn] > 1)
         and (min[plnar]=1) and (max[hamil]=0) and (min[bipart]=
1) then
              begin
                z:=26;
                pushmin(nodes);
              end
           else
            if (max[nodes] < 26) and (min[mindeg]=max[maxdeg]) and
               (min[mindeg]=3) and (min[nconn] > 1) and
(min[plnar]=1)
               and (max[hamil]=0) then
                   begin
                     z:=0;
                     pushmax(bipart);
                   end
               else
                 if (min[bipart]=1) and (max[nodes] < 26) and
                    (min[mindeg]=max[maxdeg]) and (min[mindeg]=3)
and
                    (min[nconn] > 1) and (min[plnar]=1) then
                      begin
                        z:=1;
                        pushmin(hamil);
                      end
                 else
                   if (max[hamil]=0) and (min[bipart]=1) and
(max[nodes] < 26)
                     and (min[mindeg]=max[maxdeg]) and
(min[mindeg]=3) and
                     (min[nconn] > 1) then
                        begin
                          z:=0;
                          pushmax(plnar);
                        end
                     else
                       if (min[plnar]=1) and (max[hamil]=0) and
(min[bipart]=1)                              and (max[nodes] < 26)
and (min[mindeg]=max[maxdeg])
                            and (min[mindeg]=3) then
                                begin
                                  z:=1;
                                  pushmax(nconn);
                                end
                        else
                          if (min[nconn] > 1) and (min[plnar]=1)
and
                             (max[hamil]=0) and (min[bipart]=1) and
                             (max[nodes] < 26) then
                                 begin
                                   if min[mindeg]=3 then
                                        begin
```

```
                                        z:=4;
                                        if z > min[maxdeg] then
pushmin(maxdeg);
                                       end
                                  else
                                   if max[maxdeg]=3 then
                                      begin
                                        z:=2;
                                        if z < max[mindeg] then
pushmax(mindeg);                                        end;
                                end;
    end;
end;

procedure r302;
(*************************************************)
(*                                               *)
(*    if cubic, nconn>=2, plnar, not hamil     *)
(*           then p >= 14                        *)
(*                                               *)
(*************************************************)
begin
  if (activerule[302]) and (min[nodes] < 14) and (max[nconn] >=
2)
     and (max[plnar] = 1) and (min[hamil] = 0) then
    begin
      rule:='302/ ';
      if (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) and
(min[nconn] > 1)
         and (min[plnar]=1) and (max[hamil]=0) then
          begin
            z:=14;
            pushmin(nodes);
          end
       else
         if (max[nodes] < 14) and (min[mindeg]=max[maxdeg]) and
            (min[mindeg]=3) and (min[nconn] > 1) and (min[plnar]=
1) then
            begin
              z:=1;
              pushmin(hamil);
            end
        else
          if (max[hamil]=0) and (max[nodes] < 14) and
(min[mindeg]=max[maxdeg])
             and (min[mindeg]=3) and (min[nconn] > 1) then
             begin
               z:=0;
               pushmax(plnar);
             end
        else
          if (min[plnar]=1) and (max[hamil]=0) and (max[nodes]
```

```
< 14) and
              (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) then
              begin
                z:=1;
                pushmax(nconn);
              end
          else
            if (min[nconn] > 1) and (min[plnar]=1) and
(max[hamil]=0) and
               (max[nodes] < 14) then
               begin
                 if min[mindeg]=3 then
                        begin
                          z:=4;
                          if z > min[maxdeg] then
pushmin(maxdeg);
                        end
                     else
                       if max[maxdeg]=3 then
                            begin
                              z:=2;
                              if z < max[mindeg] then
pushmax(mindeg);
                            end;
                end;
    end;
end;

procedure r303;
(************************************)
(*                                  *)
(*  if cubic,nconn>=2,bipart, and   *)
(*     not hamil  then p >= 20      *)
(*                                  *)
(************************************)
begin
  if (activerule[303]) and (min[nodes] < 20) and (max[nconn] >=
2)
     and (max[bipart] = 1) and (min[hamil] = 0) then
    begin
      rule:='303/ ';
      if (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) and
(min[nconn] > 1)
         and (min[bipart]=1) and (max[hamil]=0) then
         begin
           z:=20;
           pushmin(nodes);
         end
      else
        if (max[nodes] < 20) and (min[mindeg]=max[maxdeg]) and
(min[mindeg]=3)
           and (min[nconn] > 1) and (min[bipart]=1) then
```

```
           begin
             z:=1;
             pushmin(hamil);
           end
       else
         if (max[hamil]=0) and (max[nodes] < 20) and
(min[mindeg]=max[maxdeg])
            and (min[mindeg]=3) and (min[nconn] > 1)  then
            begin
              z:=0;
              pushmax(bipart);
            end
        else
          if (min[bipart]=1) and (max[hamil]=0) and
             (max[nodes] < 20) and (min[mindeg]=max[maxdeg]) and
             (min[mindeg]=3) then
              begin
                z:=1;
                pushmax(nconn);
              end
        else
          if (min[nconn] > 1) and (min[bipart]=1) and
             (max[hamil]=0) and (max[nodes] < 20) then
             begin
               if min[mindeg]=3 then
                  begin
                    z:=4;
                    if z > min[maxdeg] then pushmin(maxdeg);
                  end
                 else
                   if max[maxdeg]=3 then
                           begin
                             z:=2;
                             if z < max[mindeg] then
pushmax(mindeg);
                           end;
              end;

    end;
end;


procedure r304;
(*************************************************)
(*                                               *)
(*  if reg,econn>=mindeg-2>=1 then               *)
(*            |(p-2((p+1) div 2x)/2)     p even  *)
(*   eind >= |                            else   *)
(*            |(p-MAX{2((p+1+x) div 2x)-1,1})/2  *)
(*        where x=mindeg((mindeg+3) div 2) - 1   *)
(*                                               *)
(*************************************************)
```

```
var x,s:longint;
begin
  if (activerule[304]) and (min[mindeg] >= 3) and (max[reg] = 1)
then
    begin
      rule:='304/ ';
      z1:=min[mindeg];
      s:=1;
      x:=z1*((z1+3) div 2) - 1;
      if odd(min[nodes]) then
           begin
             z:=(min[nodes]+1+x) div (2*x);
             if z < 1 then
                   begin
                     z:=1;
                     s:=0;
                   end;
             z:=(min[nodes]+1) div 2-z;
           end
         else z:=min[nodes] div 2-((min[nodes]+1) div (2*x));
      if (min[reg]=1) and (min[econn] >= max[mindeg]-2) then
          begin
            if z > min[eind] then pushmin(eind);
            if max[eind] < infinity then
              begin
                if s = 0 then z:=2*max[eind]+1
                         else z:=(x*2*max[eind]+1) div (x-1);
                if z < max[nodes] then pushmax(nodes);
                z1:=min[nodes]-2*max[eind];
                if z1 > 0 then
                   begin
                     z:=0;
                     if s = 0 then
                       begin
                         if z1 > 1 then
                            z:=3+2*(min[nodes] div (z1-1));
                       end
                      else z:=3+2*((min[nodes]+1) div z1);
                     if z > 0 then
                         begin
                           z:=trunk(-1+sqrt(z));
                           if z < max[mindeg] then
pushmax(mindeg);
                         end;
                   end;
              end;
           end
          else
            if (max[eind] < z) and (min[reg]=1) then
              begin
                z:=max[mindeg]-3;
                if z < max[econn] then pushmax(econn);
```

```
               z:=min[econn]+3;
               if z > min[mindeg] then pushmin(mindeg);
             end
           else
             if (max[eind] < z) and (min[econn] >=
max[mindeg]-2) then
                begin
                  z:=0;
                  pushmax(reg);
                end;
    end;
end;

procedure r305;
(*************************************************)
(*                                               *)
(*  if cubic then eind >= p/2 -                 *)
(*             (p+3) div 18 - (Nc+4) div 6       *)
(*                                               *)
(*************************************************)
begin
  if (activerule[305]) and (min[mindeg] <= 3) and (max[mindeg] >=
3)
     and (min[maxdeg] <= 3) and (max[maxdeg] >= 3) then
    begin
      rule:='305/ ';
      z:=min[nodes] div 2-(min[nodes]+3) div 18 - (max[ncomp]+4)
div 6;
      if (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) then
         begin
           if z > min[eind] then pushmin(eind);
           if max[eind] < infinity then
             begin
               z:=(18*max[eind]+3*max[ncomp]+15) div 8;
               if z < max[nodes] then pushmax(nodes);
               z:=(8*min[nodes]-18*max[eind]-13) div 3;
               if z > min[ncomp] then pushmin(ncomp);
             end;
         end
       else
         if max[eind] < z then
             begin
               if min[mindeg]=3 then
                  begin
                    z:=4;
                    pushmin(maxdeg);
                  end
                else
                  if max[maxdeg]=3 then
                         begin
                           z:=2;
                           pushmax(mindeg);
```

```
                          end;
              end;
    end;
end;

procedure r306;
(***********************************************)
(*                                             *)
(* if clique=2 and maxdeg <= 4 then            *)
(*           e >= 6p-13nind                    *)
(*                                             *)
(***********************************************)
begin
  if (activerule[306]) and (min[clique] = 2) and (min[maxdeg] <=
4) then
    begin
      rule:='306/ ';
      z:=6*min[nodes]-13*max[nind];
      if (max[clique]=2) and (max[maxdeg] <= 4) then
       begin
         if max[nind] < infinity then
           begin
             if z > min[edges] then pushmin(edges);
             if max[edges] < infinity then
                begin
                  z:=(max[edges]+13*max[nind]) div 6;
                  if z > max[nodes] then pushmax(nodes);
                end;
            end;
         if max[edges] < infinity then
               begin
                 z:=(6*min[nodes]-max[edges]+12) div 13;
                 if z > min[nind] then pushmin(nind);
               end;
        end
       else
         if (max[edges] < z) and (max[clique]=2) then
             begin
               z:=5;
               pushmin(maxdeg);
             end
          else
            if (max[edges] < z) and (max[maxdeg] <= 4) then
                begin
                  z:=3;
                  pushmin(clique);
                end;
    end;
end;

procedure r307;
(***********************************************)
```

```
(*                                                 *)
(*  if econn > 0 then econn >= min{mindeg,         *)
(*   p*(maxdeg-2)/((max-1)**(di-1)+max(ma-2)-1)}*)
(*                                                 *)
(*************************************************)
begin
  if (activerule[307]) and (max[maxdeg] < infinity)
     and (max[diam] < infinity) and (min[econn] > 0) then
    begin
      rule:='307/ ';
      power(max[maxdeg]-1,max[diam]-1,k);
      if k < infinity then
         begin
           z:=max[maxdeg]-2;
           z1:=k+max[maxdeg]*z-1;
           if z1 > 0 then
            begin
              z1:=(min[nodes]*z-1) div z1+1;
              if z1 > min[mindeg] then
                 begin
                   z:=min[mindeg];
                   if z > min[econn] then pushmin(econn);
                   z:=max[econn];
                   if z < max[mindeg] then pushmax(mindeg);
                 end
               else
                 begin
                   z:=z1;
                   if z > min[econn] then pushmin(econn);
                   if max[econn] < infinity then
                      begin
                        z:=max[maxdeg]-2;
                        z:=(max[econn]*(k+max[maxdeg]*z-1)) div
z;
                        if z < max[nodes] then pushmax(nodes);
                        z:=(min[nodes]-1) div max[econn]-
max[maxdeg]+1;
                        z:=z*(max[maxdeg]-2)+1;
                        if z >= 1 then
                           begin
                             z:
=round(log2(z)/log2(max[maxdeg]-1)+hf)+1;
                              if z > min[diam] then pushmin(diam);
                           end;
                        if min[maxdeg] > 2 then
                          begin
                            z:=min[maxdeg]-1;
                            z1:=(min[nodes]-1) div max[econn];
                            k:=z1+1;
                            while z1 < k do
                              begin
                                z:=z+1;
```

```
                                power(z-1,max[diam]-1,z1);
                                z1:=(z1-1) div (z-2)+z;
                              end;
                            if z > min[maxdeg] then
pushmin(maxdeg);
                          end;
                       end;
                  end;
           end;
         end;
    end;
end;

procedure r308;
(*************************************************)
(*                                               *)
(*  if nconn>=2 and nind>=2 then                 *)
(*       cir >=2(p-2)/nind + 2                   *)
(*                                               *)
(*************************************************)
begin
  if (activerule[308]) and (max[nconn] >= 2) then
    begin
      rule:='308/ ';
      if (min[nconn] >= 2) and (min[nind] >= 2) then
            begin
              if max[nind] < infinity then
                 begin
                   z:=(2*min[nodes]-5) div max[nind] + 3;
                   if z > min[circ] then pushmin(circ);
                   if max[circ] < infinity then
                     begin
                       z:=((max[circ]-2)*max[nind]) div 2 +2;
                       if z < max[nodes] then pushmax(nodes);
                     end;
                  end;
               if max[circ] < infinity then
                 begin
                   z:=(2*min[nodes]-5) div (max[circ]-2) + 1;
                   if z > min[nind] then pushmin(nind);
                 end;
             end
            else
             if max[nind] < infinity then
               begin
                 z:=(2*min[nodes]-5) div max[nind] + 3;
                 if (max[circ] < z) and (min[nconn] >= 2) then
                     begin
                       z:=1;
                       pushmax(nind);
                     end
                   else
```

```
                    if (max[circ] < z) and (min[nind] >= 2) then
                       begin
                         z:=1;
                         pushmax(nconn);
                       end;
               end;
    end;
end;

procedure r309;
(*************************************************)
(*                                               *)
(* if cubic,nconn=3,plnar then circ>=min(p,17) *)
(*      note: if true and circ < 17 then Hamil  *)
(*                                               *)
(*************************************************)
begin
  if (activerule[309]) and (max[plnar] = 1) and (max[nconn] >= 3)
      and (min[maxdeg] <= 3) and (min[hamil] = 0) then
    begin
      rule:='309/ ';
      z:=min[nodes];
      if (z > 17) or (max[hamil] = 0) then z:=17;
      if (min[nconn]=max[maxdeg]) and
         (min[nconn]=3) and (min[plnar]=1) then
          begin
            if z > min[circ] then pushmin(circ);
            if max[circ] < 17 then
                begin
                  z:=1;
                  pushmin(hamil);
                end;
          end
        else
          if (max[circ] < z) and (min[nconn]=max[maxdeg]) and
             (min[nconn]=3) then
              begin
                z:=0;
                pushmax(plnar);
              end
        else
          if (min[plnar]=1) and (max[circ] < z) and (max[maxdeg]=
3) then
                begin
                  z:=2;
                  pushmax(nconn);
                end
         else
           if (min[nconn]=3) and (min[plnar]=1) and (max[circ]
< z) then
               begin
                 z:=4;
```

```
                pushmin(maxdeg);
              end;
    end;
end;

procedure r310;
(*********************************************)
(*                                           *)
(* if cubic,nconn=3,plnar and p <= 36        *)
(*        then hamil                         *)
(*                                           *)
(*********************************************)
begin
  if (activerule[310]) and (min[nodes] < 38) and (min[maxdeg] <=
3)
     and (max[nconn] >= 3) and (max[plnar] = 1) and (min[hamil] =
0) then
    begin
      rule:='310/ ';
      if (min[nconn]=max[maxdeg]) and (min[nconn]=3)
         and (min[plnar]=1) and (max[nodes] <= 36) then
         begin
           z:=1;
           pushmin(hamil);
         end
       else
         if (max[hamil]=0) and (min[nconn]=max[maxdeg])
            and (min[nconn]=3) and (min[plnar]=1) then
            begin
              z:=38;
              pushmin(nodes);
            end
        else
         if (max[nodes] <= 36) and (max[hamil]=0)
             and (max[maxdeg]=3) and (min[plnar]=1)then
              begin
                z:=2;
                pushmax(nconn);
              end
         else
           if (min[nconn]=3) and (max[nodes] <= 36) and
(max[hamil]=0)
              and (min[plnar]=1) then
               begin
                 z:=4;
                 pushmin(maxdeg);
               end
         else
           if (min[nconn]=3) and (max[nodes] <= 36) and
(max[hamil]=0)
              and (min[nconn]=max[maxdeg]) then
              begin
```

```
              z:=0;
              pushmax(plnar);
            end;
    end;
end;

procedure r311;
(*********************************************)
(*                                           *)
(* if clique < chr=maxdeg then                *)
(*           |2*maxdeg      when maxdeg >=9   *)
(*      p >= |                               *)
(*           |2*maxdeg-1    when maxdeg <=8   *)
(*                                           *)
(*********************************************)
begin
  if activerule[311] then
    begin
      rule:='311/ ';
      z:=2*min[mindeg];
      if min[maxdeg] < 9 then z:=z-1;
      if (max[clique] < min[chr]) and (min[chr]=max[maxdeg]) then
           begin
             if z > min[nodes] then pushmin(nodes);
           end
        else
          if (max[nodes] < z) and (max[clique] < min[chr]) then
              begin
                z:=max[maxdeg]-1;
                if z < max[chr] then pushmax(chr);
                z:=min[chr]+1;
                if z > min[maxdeg] then pushmin(maxdeg);
              end
         else
           if (min[chr]=max[maxdeg]) and (max[nodes] < z) then
              begin
                z:=min[chr];
                if z > min[clique] then pushmin(clique);
                z:=max[clique];
                if z < max[chr] then pushmax(chr);
              end;
     end;
end;

procedure r312;
(*********************************************)
(*                                           *)
(*  if cubic and econn>=2 then eind=p/2       *)
(*                                           *)
(*********************************************)
begin
  if (activerule[312]) and (max[econn] >= 2) and (max[maxdeg] >=
```

```
3)
     and (min[mindeg] <= 3) then
    begin
      rule:='312/ ';
      if (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) and
         (min[econn] >= 2) then
            begin
              z:=min[nodes] div 2;
              if z > min[eind] then pushmin(eind);
              z:=2*max[eind];
              if z < max[nodes] then pushmax(nodes);
            end
        else
          if (max[eind] < min[nodes] div 2) and
(min[mindeg]=max[maxdeg])
             and (min[mindeg]=3) then
             begin
               z:=1;
               pushmax(econn);
             end
        else
          if (min[econn] >= 2) and (max[eind] < min[nodes] div 2)
then
            begin
              if min[mindeg]=3  then
                 begin
                   z:=4;
                   pushmax(maxdeg);
                 end
               else
                 if max[maxdeg]=3 then
                    begin
                      z:=2;
                      pushmax(mindeg);
                    end;
            end;
    end;
end;

procedure r313;
(*********************************************)
(*                                           *)
(*  if reg,nconn>=3 then  circ >= Q          *)
(*            where Q=min(p,3*mindeg)        *)
(*                                           *)
(*********************************************)
begin
  if (activerule[313]) and (min[hamil] = 0) and (max[reg] = 1)
     and (max[nconn] >= 3) then
    begin
      rule:='313/ ';
      z:=3*min[mindeg];
```

```
      if z > min[nodes] then z:=min[nodes];
      if (min[reg] = 1) and (min[nconn] >= 3) then
         begin
           if z > min[circ] then pushmin(circ);
           if max[circ] < 3*min[mindeg] then
                 begin
                   z:=1;
                   pushmin(hamil);
                 end
               else
                 if (max[circ] < min[nodes]) or (max[hamil] = 0)
then
                    begin
                      z:=max[circ] div 3;
                      if z < max[mindeg] then pushmax(mindeg);
                    end;
         end
        else
          if (max[circ] < z) and (min[reg]=1) then
            begin
              z:=2;
              pushmax(nconn);
            end
          else
            if (min[nconn] >= 3) and (max[circ] < z) then
              begin
                z:=0;
                pushmax(reg);
              end;
    end;
end;

procedure r314;
(***********************************************)
(*                                             *)
(* if reg,nconn>=2,p<=3*mindeg+3 then circ >=Q  *)
(*          where Q=min(p,3*mindeg)            *)
(*                                             *)
(***********************************************)
begin
  if (activerule[314]) and (max[reg] = 1) and (max[nconn] >= 2)
     and (min[hamil] = 0) then
    begin
      rule:='314/ ';
      z1:=3*min[mindeg];
      if z1 > min[nodes] then z:=min[nodes]
                         else z:=z1;
      if (min[reg]=1) and (min[nconn] >= 2) and (max[nodes] <= z1
+3) then
          begin
            if z > min[circ] then pushmin(circ);
            if max[circ] < z1 then
```

```
              begin
                z:=1;
                pushmin(hamil);
              end
            else
              if (max[circ] < min[nodes]) or (max[hamil] = 0)
then
                  begin
                    z:=max[circ] div 3;
                    if z < max[mindeg] then pushmax(mindeg);
                  end;
         end
        else
          if (max[circ] < z) and (min[reg]=1) and (min[nconn] >=
2) then
             begin
               z:=z1+4;
               if z > min[nodes] then pushmin(nodes);
               z:=(max[nodes]-4) div 3;
               if z < max[mindeg] then pushmax(mindeg);
             end
          else
            if (max[nodes] <= z1+3) and (max[circ] < z) and
(min[reg]=1) then
               begin
                 z:=1;
                 pushmax(nconn);
               end
          else
           if (min[nconn] >= 2) and (max[nodes]<= z1+3) and
(max[circ] < z) then
              begin
                z:=0;
                pushmax(reg);
              end;
    end;
end;

procedure r315;
(*********************************************)
(*                                           *)
(*  if reg,nconn>=2  then    circ >= Q       *)
(*        Q = min(p,3*mindeg,2*mindeg+4)     *)
(*                                           *)
(*********************************************)
begin
  if (activerule[315]) and (max[reg] = 1) and (max[nconn] >= 2)
     and (min[hamil] = 0) then
    begin
      rule:='315/ ';
      z:=3*min[mindeg];
      if z > min[nodes] then z:=min[nodes];
```

```
      if z > 2*min[mindeg]+4 then z:=2*min[mindeg]+4;
      if (min[reg] = 1) and (min[nconn] >= 2) then
         begin
           if z > min[circ] then pushmin(circ);
           z1:=2*min[mindeg]+4;
           if z1 > 3*min[mindeg] then z1:=3*min[mindeg];
           if max[circ] < z1 then
               begin
                 z:=1;
                 pushmin(hamil);
               end
             else
               begin
                 z1:=3*min[mindeg];
                 if z1 > min[nodes] then z1:=min[nodes];
                 if max[circ] < z1 then
                     begin
                       z:=(max[circ]-4) div 2;
                       if z < max[mindeg] then pushmax(mindeg);
                     end
                   else
                     begin
                       z1:=2*min[mindeg]+4;
                       if z1 > min[nodes] then z1:=min[nodes];
                       if max[circ] < z1 then
                          begin
                            z:=max[circ] div 3;
                            if z < max[mindeg] then
pushmax(mindeg);
                          end;
                      end;
                 end;
         end
      else
        if (max[circ] < z) and (min[reg]=1) then
          begin
            z:=1;
            pushmax(nconn);
          end
       else
         if (min[nconn] >= 2) and (max[circ] < z) then
             begin
               z:=0;
               pushmax(reg);
             end;

    end;
end;

procedure r316;
(************************************************)
(*                                              *)
```

```
(*  if reg, p even, maxdeg>= 6p/7 then echr=maxdeg *)
(*                                                 *)
(*************************************************)
begin
  if (activerule[316]) and (max[reg] = 1) then
    begin
      rule:='316/ ';
      if (min[reg]=1) and (max[nodes]=min[nodes]) and
(not(odd(max[nodes])))
         and (min[maxdeg] >= 6*max[nodes]/7) then
         begin
           z:=max[maxdeg];
           if z < max[echr] then pushmax(echr);
           z:=min[echr];
           if z > min[maxdeg] then pushmin(maxdeg);
         end
       else
        if (min[echr]>max[maxdeg]) and (min[reg]=1) and
(max[nodes]=min[nodes])
           and (not(odd(max[nodes]))) then
           begin
             z:=(6*max[nodes]-1) div 7;
             if z < max[nodes] then pushmax(maxdeg);
             z:=(7*min[maxdeg]+6) div 6;
             if z > min[nodes] then pushmin(nodes);
            end
        else
          if (min[maxdeg] >= 6*max[nodes]/7) and (min[echr] >
max[maxdeg])
             and (min[reg]=1) then
              begin
                z:=max[nodes]-1;
                if odd(z) then pushmax(nodes);
                z:=min[nodes]+1;
                if odd(z) then pushmin(nodes);
              end
         else
           if (max[nodes]=min[nodes]) and (not(odd(max[nodes])))
and
              (min[maxdeg] >= 6*max[nodes]/7) and (min[echr] >
max[maxdeg])
              then
                begin
                  z:=0;
                  pushmax(reg);
                end;
    end;
end;

procedure r317;
(************************************************)
(*                                              *)
```

```
(* if maxdeg=p-1, e <= 2*((p-1) div 2)**2       *)
(*                     (+ mindeg , when p even) *)
(*       then echr = maxdeg                      *)
(*                                               *)
(***********************************************)
begin
  if activerule[317] then
    begin
      rule:='317/ ';
      z:=(min[nodes]-1) div 2;
      z:=2*z*z;
      if not(odd(min[nodes])) then z:=z+min[mindeg];
      if (min[maxdeg]=max[nodes]-1) and (max[edges] <= z) then
         begin
           z:=max[maxdeg];
           if z < max[echr] then pushmax(echr);
           z:=min[echr];
           if z > min[maxdeg] then pushmin(maxdeg);
         end
       else
         if (max[maxdeg] < min[echr]) and
(min[maxdeg]=max[nodes]-1) then
          begin
            z:=z+1;
            if z > min[edges] then pushmin(edges);
            if not(odd(min[nodes])) then
                 begin
                   z:=min[nodes]-2;
                   z:=max[edges]-z*z div 2-1;
                   if z < max[mindeg] then pushmax(mindeg);
                 end;
          end;
    end;
end;

procedure r318;
(***********************************************)
(*                                             *)
(* if spectr > sqrt(maxdeg)*[(p*Ck)/2]**(1/2k)  *)
(*      then g <= 2k+1                         *)
(*            where Ck = ( 2k choose k)/(k+1)  *)
(*              i.e., k\th Catalan Number.     *)
(*                                             *)
(***********************************************)
var k,ks,maxit:longint;
    Ck,zk,zz:real;
begin
  if (activerule[318]) and (min[girth] < infinity)
     and (max[nodes] < infinity) and (max[maxdeg] < infinity)
then
    begin
      rule:='318/ ';
```

```
      maxit:=20;
      rz:=-1.0;
      k:=0;
      Ck:=1;
      ks:=(min[girth]-2) div 2;
      zk:=lammin+1.0;
      zz:=sqrt(max[maxdeg]);
      while (lammin <= zk) and (k <= maxit) do
         begin
           if k=ks then rz:=zz+0.001;
           k:=k+1;
           Ck:=(4*k-2)*Ck/(k+1);
           zz:=root(max[nodes]*Ck/2,2*k);
           zk:=sqrt(max[maxdeg])*zz+0.001;
         end;
      if k <= maxit then
         begin
           z:=2*k+1;
           if z < max[girth] then pushmax(girth);
           if rz > 0.0 then
            begin
              zk:=lammin/rz;
              z:=round(zk*zk+hf);
              if z > min[maxdeg] then pushmin(maxdeg);
              rz:=sqrt(max[maxdeg])*rz;
              if rz < lammax then pushlammax;
             end;
         end;
    end;
end;

procedure r319;
(*****************************************************)
(*                                                   *)
(*  if spectr >= [2e(2maxdeg-1)-2r(maxdeg-r)]**(1/4)  *)
(*          then girth <= 4                          *)
(*      where 2e=q*maxdeg+r and 0 <= r < maxdeg      *)
(*                                                   *)
(*****************************************************)
begin
  if (activerule[319]) and (max[girth] > 4) and
      (max[edges] < infinity) and (max[maxdeg] < infinity) then
    begin
      rule:='319/ ';
      z:=max[edges];
      z1:=2*z-max[maxdeg]*(2*z div max[maxdeg]);
      z:=2*z*(2*max[maxdeg]-1)-2*z1*(max[maxdeg]-z1);
      rz:=sqrt(sqrt(z));
      if lammin >= rz then
            begin
              z:=4;
              pushmax(girth);
```

```
           end
         else
           if min[girth] > 4 then pushlammax;
    end;
end;

procedure r320;
(*************************************************)
(*                                               *)
(* if connected, reg, p odd, p < 5*mindeg/2      *)
(*         then girth <= 3                       *)
(*                                               *)
(*************************************************)
begin
  if (activerule[320]) and (max[girth] > 3)
     and (max[connct] = 1) and (max[reg] = 1) then
    begin
      rule:='320/ ';
      if (min[connct]=1) and (min[reg]=1) and
(max[nodes]=min[nodes])
         and (odd(max[nodes])) and (max[nodes] < 5*min[mindeg]/2)
then
         begin
           z:=3;
           pushmax(girth);
         end
       else
         if (min[girth] > 3) and (min[connct]=1) and (min[reg]=1)
and
            (max[nodes]=min[nodes]) and (odd(max[nodes])) then
            begin
              z:=(5*min[mindeg]+1) div 2;
              if not(odd(z)) then z:=z+1;
              if z > min[nodes] then pushmin(nodes);
              z:=2*max[nodes] div 5;
              if odd(z) then z:=z-1;
              if z < max[mindeg] then pushmax(mindeg);
            end
          else
            if (max[nodes] < 5*min[mindeg]/2) and (min[girth] >
3) and
               (min[connct]=1) and (min[reg]=1) then
               begin
                 if odd(max[nodes]) then
                     begin
                       z:=max[nodes]-1;
                       pushmax(nodes);
                     end;
                 if odd(min[nodes]) then
                      begin
                        z:=min[nodes]+1;
                        pushmin(nodes);
```

```
                    end;
               end
              else
                if (max[nodes]=min[nodes]) and (odd(max[nodes]))
and
                    (max[nodes] < 5*min[mindeg]/2) and
(min[girth] > 3)
                    and (min[connct]=1) then
                    begin
                      z:=0;
                      pushmax(reg);
                    end
              else
               if (min[reg]=1) and (max[nodes]=min[nodes]) and
(odd(max[nodes]))
                   and (max[nodes] < 5*min[mindeg]/2) and
(min[girth] > 3) then
                    begin
                      z:=0;
                      pushmax(connct);
                    end;
    end;
end;

procedure r321;
(************************************************)
(*                                              *)
(* if connct, reg, and not complete then        *)
(*         eccov >= y*p/x                       *)
(*    where:                                    *)
(*    x = mindeg+1, y = 3      when mindeg <= 3 *)
(*      = (mindeg-1)(mindeg-2) when mindeg  > 4 *)
(*                  y = mindeg                  *)
(*                                              *)
(************************************************)
begin
  if (activerule[321]) and (max[connct] = 1) and
     (min[compl] = 0) and (max[reg] = 1) and (min[mindeg] > 1)
then
    begin
      rule:='321/ ';
      if (min[connct]=1) and (min[reg]=1) and
         (max[compl] = 0) then
           begin
             if max[mindeg] <= 3 then
                 begin
                   z:=3*min[nodes];
                   z:=(z-1) div (max[mindeg]+1)+1;
                   if z > min[eccov] then pushmin(eccov);
                   if max[eccov] < infinity then
                      begin
                        z:=((max[mindeg]+1)*max[eccov]) div 3;
```

```
                        if z < max[nodes] then pushmax(nodes);
                        z:=(3*min[nodes]-1) div max[eccov];
                        if z > min[mindeg] then pushmin(mindeg);
                      end;
                  end
                else
                  if max[mindeg] >= 5 then
                    begin
                      z1:=max[mindeg];
                      if z1 < infinity then
                         begin
                           z1:=(z1-1)*(z1-2);
                           if max[eccov] < infinity then
                               begin
                                 z:=(z1*max[eccov]) div
max[mindeg];
                                 if z < max[nodes] then
pushmax(nodes);
                               end;
                           z:=(max[mindeg]*min[nodes]-1) div z1 +
1;
                           if z > min[eccov] then pushmin(eccov);
                         end;
                       if max[eccov] < infinity then
                            begin
                              rz:=3+min[nodes]/max[eccov];
                              if rz*rz >= 8 then
                                begin
                                  rz:=(rz+sqrt(rz*rz-8))/2;
                                  z:=round(rz+hf);
                                  if z > min[mindeg] then
pushmin(mindeg);
                                end;
                            end;
                     end;
            end
          else
            begin
              if max[mindeg] <= 3 then z:=(3*min[nodes]+3) div 4
                 else
                   if min[mindeg] > 4 then
                      begin
                        z1:=(max[mindeg]-1)*(max[mindeg]-2);
                        z:=(max[mindeg]*min[nodes]-1) div z1 +
1;
                      end
                     else z:=0;
               if max[eccov] < z then
                  begin
                    if (min[connct]=1) and (min[reg]=1) then
                      begin
                        z:=1;
```

```
                        pushmin(compl);
                      end
                    else
                      if (max[compl] = 0) and (min[connct]=1)
then
                        begin
                          z:=0;
                          pushmax(reg);
                        end
                       else
                        if (min[reg]=1) and (max[compl] = 0)
then
                           begin
                             z:=0;
                             pushmax(connct);
                           end;
                  end;
         end;
   end;
end;

procedure r322;
(*************************************************)
(*                                               *)
(* if connct, reg, mindeg <= 4, and not Kp then *)
(*      eccov >= 3*p/5  + x                      *)
(*         where      |1 if p = 7, or 3 mod 5   *)
(*               x =  |          except [13,18] *)
(*                    |0  otherwise             *)
(*                                               *)
(*************************************************)
begin
  if (activerule[322]) and (max[connct] = 1) and (max[reg] = 1)
     and (min[compl] = 0) and (min[mindeg] <= 4) then
    begin
      rule:='322/ ';
      z:=min[nodes];
      k:=0;
      if (z=7) or ((z = 3 mod 5) and (z <> 13) and (z <> 18))
then k:=1;
      z1:=max[eccov];
      z:=(3*z+4) div 5+k;
      if (min[connct]=1) and (min[reg]=1) and
         (max[mindeg] <= 4) and (max[compl] = 0) then
         begin
           if z > min[eccov] then pushmin(eccov);
           if min[mindeg] > 1 then
             begin
               z:=(5*(z1-k)) div 3;
               if z < max[nodes] then pushmax(nodes);
             end;
         end
```

```
        else
          if (max[compl] = 0) and (max[eccov] < z) and
(min[connct]=1)
               and (min[reg]=1) then
               begin
                 z:=5;
                 pushmin(mindeg);
               end
             else
               if (max[mindeg] <= 4) and (max[compl] = 0)
                  and (max[eccov] < z) and (min[connct]=1) then
                    begin
                      z:=0;
                      pushmax(reg);
                    end
                else
                  if (min[reg]=1) and (max[mindeg] <= 4) and
                     (max[compl] = 0) and (max[eccov] < z) then
                     begin
                       z:=0;
                       pushmax(connct);
                     end
                   else
                     if (min[reg] = 1) and (max[mindeg] <= 4) and
                        (max[eccov] < z) and (min[connct] = 1)
then
                          begin
                            z:=1;
                            pushmin(compl);
                          end;
    end;
end;

procedure r323;
(*************************************************)
(*                                               *)
(*  if girth >= 6 then                           *)
(*          nind >= p(2x-1)/(x**2+2x-1)          *)
(*          where x = maxdeg                     *)
(*                                               *)
(*************************************************)
begin
  if (activerule[323]) and (max[girth] >= 6) and (min[girth]
< infinity) then
    begin
      rule:='323/ ';
      z:=max[maxdeg];
      if z < infinity then
        begin
          z:=z*z+2*z-1;
          z:=((2*max[maxdeg]-1)*min[nodes]+z-1) div z;
        end;
```

```
      if min[girth] >= 6 then
           begin
             z1:=max[maxdeg];
             if z1 < infinity then
                begin
                  if z > min[nind] then pushmin(nind);
                  z:=max[nind];
                  if z < infinity then
                       begin
                         rz:=z1;
                         z:=trunk((z*(rz*rz+2*rz-1))/(2*rz-1));
                         if z < max[nodes] then pushmax(nodes);
                       end;
                end;
             z:=max[nind];
             if z < infinity then
                  begin
                    z1:=min[nodes]-z;
                    z1:=z1*(z1-z);
                    if z1 >= 0 then
                       begin
                         z:=round((min[nodes]-z+sqrt(z1))/z+hf);
                         if z > min[maxdeg] then pushmin(maxdeg);
                       end;
                  end;
          end
        else
          if max[maxdeg] < infinity then
              if max[nind] < z then
                  begin
                     z:=5;
                     pushmax(girth);
                  end;
    end;
end;

procedure r324;
(*********************************************)
(*                                           *)
(*  if cubic  then  nind >=   19p/52, if gi >=6 *)
(*                            20p/53, if gi >=8 *)
(*                                           *)
(*********************************************)
begin
  if (activerule[324]) and (max[girth] >= 6) and (min[girth]
< infinity)
     and (min[maxdeg] <= 3) and (max[mindeg] >= 3) then
    begin
      rule:='324/ ';
      if (min[mindeg]=max[maxdeg]) and (min[mindeg]=3) then
           begin
             if min[girth] >= 8 then
```

```
                  begin
                    z:=(20*min[nodes]+52) div 53;
                    if z > min[nind] then pushmin(nind);
                    if max[nind] < infinity then
                       begin
                         z:=(53*max[nind]) div 20;
                         if z < max[nodes] then pushmax(nodes);
                       end;
                  end
                else
                  if min[girth] >= 6 then
                     begin
                       z:=(19*min[nodes]+51) div 52;
                       if z > min[nind] then pushmin(nind);
                       if max[nind] < infinity then
                          begin
                            z:=(52*max[nind]) div 19;
                            if z < max[nodes] then
pushmax(nodes);
                          end;
                     end
                  else
                  begin
                    if 52*max[nind] < 19*min[nodes] then z:=5
                       else
                         if 53*max[nind] < 20*min[nodes] then
z:=7
                           else z:=max[girth];
                    if z > max[girth] then pushmax(girth);
                  end;
          end
        else
          if ((min[girth] >= 6) and (52*max[nind] < 19
*min[nodes])) or
             ((min[girth] >= 8) and (53*max[nind] < 20
*min[nodes])) then
            begin
              if min[mindeg]=3 then
                    begin
                      z:=4;
                      pushmin(maxdeg);
                    end
                  else
                    if max[maxdeg]=3 then
                      begin
                        z:=2;
                        pushmax(mindeg);
                      end;
            end;
    end;
end;
```

```
procedure r325;
(*************************************************)
(*                                               *)
(* if reg, girth even and >= 6, connct, and      *)
(*    p <= [x(x-3)+2(x-1)**(g/2)]/(x-2)          *)
(*           where x=mindeg                      *)
(*      then   bipartite and diam=(g/2)+1        *)
(*                                               *)
(*************************************************)
begin
 if (activerule[325]) and (min[mindeg] >= 3) and (max[reg] = 1)
and
     (max[connct] = 1) and (max[girth] < infinity) and
(min[girth] >= 6) and
     (not(odd(min[girth]))) and (min[girth]=max[girth]) then
    begin
      rule:='325/ ';
      z:=min[mindeg];
      power(z-1,min[girth] div 2,z1);
      z:=(z*(z-3)+2*z1) div (z-2);
      if (min[reg]=1) and (min[connct]=1) and (max[nodes] <= z)
then
         begin
           z:=1;
           if z > min[bipart] then pushmin(bipart);
           z:=min[girth] div 2 +1;
           if z < max[diam] then pushmax(diam);
           if z > min[diam] then pushmin(diam);
         end
       else
         if ((max[bipart]=0) or (min[diam] > max[girth] div 2 +
1)) and
            (min[reg]=1) and (min[connct]=1) then
            begin
              z:=z+1;
              if z > min[nodes] then pushmin(nodes);
            end
          else
            if (max[nodes] <= z) and ((max[bipart]=0) or
(min[diam] >
                 max[girth] div 2 + 1)) and (min[reg]=1) then
               begin
                 z:=0;
                 pushmax(connct);
               end
           else
            if (min[connct]=1) and (max[nodes] <= z) and
               ((max[bipart]=0) or (min[diam] > max[girth] div 2
+1)) then
                begin
                  z:=0;
                  pushmax(reg);
```

```
                end;
    end;
end;

procedure r326;
(************************************************)
(*                                              *)
(* if bipart and p is odd then                  *)
(*          t <= ceil[(p*p-1)/(8(p-2))]         *)
(*                                              *)
(************************************************)
begin
  if (activerule[326]) and (max[nodes] = min[nodes]) and
     (odd(max[nodes])) and (max[bipart] = 1) then
    begin
      rule:='326/ ';
      z:=max[nodes];
      z:=(z*z-2) div (8*(z-2))+1;
      if min[bipart]=1 then
         begin
           if z < max[thick] then pushmax(thick);
         end
        else
          if min[thick] > z then
             begin
               z:=0;
               pushmax(bipart);
             end;
    end;
end;

procedure r327;
(************************************************)
(*                                              *)
(*  if e > 0 then maxdeg >= 2*thick-1           *)
(*     note: here, e is always > 0              *)
(*                                              *)
(************************************************)
begin
  if activerule[327] then
    begin
      rule:='327/ ';
      z:=2*min[thick]-1;
      if z > min[mindeg] then pushmin(maxdeg);
      z:=max[maxdeg];
      if z < infinity then
         begin
           z:=(z+1) div 2;
           if z < max[thick] then pushmax(thick);
         end;
    end;
end;
```

```
procedure r328;
(*****************************)
(*                           *)
(*     nconn <= 6*thick-1    *)
(*                           *)
(*****************************)
begin
  if activerule[328] then
    begin
      rule:='328/ ';
      if max[thick] < infinity then
         begin
           z:=6*max[thick]-1;
           if z < max[nconn] then pushmax(nconn);
         end;
      z:=min[nconn] div 6 + 1;
      if z > min[thick] then pushmin(thick);
    end;
end;

procedure r329;
(*********************************************)
(*                                           *)
(* if cubic and girth=10 then p >= 70        *)
(*                                           *)
(*********************************************)
begin
  if (activerule[329]) and (max[reg] = 1) and (min[girth] =
max[girth]) and
     (min[girth] = 10) and (min[mindeg] = max[mindeg]) and
(min[mindeg] = 3)
     and (min[nodes] < 70) then
    begin
      rule:='329/ ';
      if min[reg]=1 then
        begin
          z:=70;
          pushmin(nodes);
        end
      else
        if max[nodes] < 70 then
           begin
             z:=0;
             pushmax(reg);
           end;
    end;
end;

procedure r330;
(*********************************************)
(*                                           *)
```

```
(*  if e >= max[a,b] then hamiltonian          *)
(*     where                                   *)
(*        a = p(p-1)/2-xp+(3x**2+x+2)/2        *)
(*              x=mindeg                       *)
(*        b = m*(m-1)/2+n**2+1                 *)
(*             where m = (p+2) div 2           *)
(*                   n = (p-1) div 2           *)
(*                                             *)
(***********************************************)
var m,n,z2:longint;
begin
  if (activerule[330]) and (max[nodes] < infinity) and
     (max[mindeg] < infinity) and (min[hamil] = 0) then
    begin
      rule:='330/ ';
      z:=max[mindeg];
      z1:=max[nodes];
      z:=z1*(z1-1) div 2-z*z1+(3*z*z+z+2) div 2;
      z2:=min[mindeg];
      z2:=z1*(z1-1) div 2 -z2*z1+(3*z2*z2+z2+2) div 2;
      if z < z2 then z:=z2;
      m:=(z1+2) div 2;
      n:=(z1-1) div 2;
      z1:=m*(m-1) div 2+n*n+1;
      if z1 > z then z:=z1;
      if min[edges] >= z then
           begin
             z:=1;
             pushmin(hamil);
           end
        else
          if max[hamil]=0 then
              begin
                z:=z-1;
                if z < max[edges] then pushmax(edges);
              end;

    end;
end;

procedure r331;
(***********************************************)
(*                                             *)
(*   e <= p(p-1)/2-(x-k+1)*(p-x-1)             *)
(*      where x = mindeg  and k = nconn        *)
(*                                             *)
(***********************************************)
var k,x,y:longint;
begin
  if (activerule[331]) and (max[nconn] < infinity) then
    begin
      rule:='331/ ';
```

```
      z1:=max[nodes];
      k:=max[nconn];
      x:=min[mindeg];
      if x < k then x:=k;
      if z1 < infinity then
       begin
         z:=(z1*(z1-1)) div 2-(x-k+1)*(z1-x-1);
         if z < max[edges] then pushmax(edges);
         if k < z1-1 then
           begin
             rz:=min[edges]-z1*(z1-1)/2+(z1-k)*(z1-k)/4;
             if rz <= 0 then z:=(z1+k-2) div 2
                        else z:=trunk((z1+k-2-2*sqrt(rz))/2);
             if z < max[mindeg] then pushmax(mindeg);
           end;
         y:=min[mindeg];
         if z1-y-1 > 0 then z:=y+1-((z1*(z1-1)-2*min[edges]) div
(2*(z1-y-1)))
                       else z:=y;
         z1:=min[nodes];
         if z1-y > 1 then
              begin
                z1:=y+1-((z1*(z1-1)-2*min[edges]) div (2*(z1-
y-1)));
                if z1 < z then z:=z1;
              end;
         if z > min[nconn] then pushmin(nconn);
        end;
       z:=8*min[edges]+1-4*(x-k+1)*(x+k);
       if z >= 0 then z:=round((2*(x-k)+3+sqrt(z))/2+hf);
       x:=max[mindeg];
       z1:=8*min[edges]+1-4*(x-k+1)*(x+k);
       if z1 >= 0 then z1:=round((2*(x-k)+3+sqrt(z1))/2+hf);
       if z1 < z then z:=z1;
       if z > min[nodes] then pushmin(nodes);
    end;
end;

procedure r332;
(*********************************************)
(*                                           *)
(* if tree, maxdeg < p-1 then Bwd <= (p-1)/2 *)
(*                                           *)
(*********************************************)
begin
  if (activerule[332]) and (max[tree]=1) and (min[maxdeg]
< min[nodes]-1) then
    begin
      rule:='332/ ';
      z:=max[nodes];
      z1:=(z-1) div 2;
      if (min[tree]=1) and (max[maxdeg] < min[nodes]-1) then
```

```
        begin
          if z < infinity then
               begin
                 z:=z1;
                 if z < max[bwidth] then pushmax(bwidth);
               end;
          z:=2*min[bwidth]+1;
          if z > min[nodes] then pushmin(nodes);
         end
       else
         if (min[bwidth] > z1) and (min[tree]=1) then
           begin
             z:=min[nodes]-1;
             pushmin(maxdeg);
             z:=max[maxdeg]+1;
             if z < max[nodes] then pushmax(nodes);
            end
         else
           if (max[maxdeg] < min[nodes]-1) and (min[bwidth] > z1)
then
                begin
                  z:=0;
                  pushmax(tree);
                end;
    end;
end;

procedure r333;
(*************************)
(*                       *)
(*   e >= 2*bwdth-1      *)
(*                       *)
(*************************)
begin
  if activerule[333] then
    begin
      rule:='333/ ';
      z:=2*min[bwidth]-1;
      if z > min[edges] then pushmin(edges);
      if max[edges] < infinity then
        begin
          z:=(max[edges]+1) div 2;
          if z < max[bwidth] then pushmax(bwidth);
        end;
    end;
end;

procedure r334;
(*************************************************)
(*                                               *)
(* if girth >= 5, mindeg >= 3, and connected     *)
(*    then dom <= [p-(diam div 3)*x-1-x(x-1)/2]/2  *)
```

```
(*                x = mindeg - 1                  *)
(*                                                *)
(**************************************************)
begin
  if (activerule[334]) and (max[girth] >= 5) and (max[mindeg] >=
3)
       and (max[connct] = 1) then
    begin
      rule:='334/ ';
      z1:=min[mindeg]-1;
      z:=2*min[dom]+(min[diam] div 3)*z1+1+(z1*(z1-1)) div 2;
      if (min[connct]=1) and (min[girth] >= 5) and
(min[mindeg] >= 3) then
        begin
          if z > min[nodes] then pushmin(nodes);
          if max[nodes] < infinity then
             begin
               z:=(max[nodes]-z+2*min[dom]) div 2;
               if z < max[dom] then pushmax(dom);
               z:=min[mindeg]-1;
               z:=(max[nodes]-1-2*min[dom]-(z*(z-1)) div 2) div
z;
               z:=2+3*z;
               if z < max[diam] then pushmax(diam);
               z:=2*(min[diam] div 3)-1;
               z:=trunk((-z+2+sqrt(z*z+8*(max[nodes]-1-2
*min[dom])))/2);
               if z < max[mindeg] then pushmax(mindeg);
             end;
        end
       else
         if (max[nodes] < z) and (min[connct]=1) and
(min[girth] >= 5) then
           begin
             z:=2;
             pushmax(mindeg);
           end
         else
           if (min[mindeg] >= 3) and (max[nodes] < z) and
(min[connct]=1)then
            begin
              z:=4;
              pushmax(girth);
            end
          else
           if (min[girth] >= 5) and (min[mindeg] >= 3) and
(max[nodes] < z) then
            begin
              z:=0;
              pushmax(connct);
            end;
    end;
```

```
end;

procedure r335;
(*************************************************)
(*                                               *)
(* if p/2 <= B (=bwidth) <= p-1 then             *)
(*       e >= (2(p div 2)-1)*(p/(p-2))**x        *)
(*                x = B-(p div 2)                *)
(*                                               *)
(*************************************************)
begin
  if (activerule[335]) and (max[nodes] = min[nodes]) then
    begin
      rule:='335/ ';
      z1:=max[nodes];
      z:=min[bwidth]-(z1 div 2);
      if z > 0 then
        begin
          realPower(z1/(z1-2),z,rz);
          z:=round((2*(z1 div 2)-1)*rz+hf);
          if (max[nodes] <= 2*min[bwidth]) and (max[bwidth]
< min[nodes]) then
            begin
              if z > min[edges] then pushmin(edges);
              z:=z1 div 2;
              z:=trunk(z+log2(max[edges]/(2
*z-1))/log2(z1/(z1-2)));
              if z < max[bwidth] then pushmax(bwidth);
            end
          else
            if max[edges] < z then
               begin
                 z:=(z1-1) div 2;
                 if z < max[bwidth] then pushmax(bwidth);
               end;
      end;
    end;
end;

procedure r336;
(*************************************************)
(*                                               *)
(*  if B (=bwidth) >= p/2 then                   *)
(*        e >= p(p-1)/(2p-2B)                    *)
(*                                               *)
(*************************************************)
begin
  if (activerule[336]) and (max[nodes] < infinity) then
    begin
      rule:='336/ ';
      z1:=max[nodes];
      z:=z1-(z1*(z1-1)-1) div (2*max[edges])-1;
```

```
      if z < (z1-1) div 2 then z:=(z1-1) div 2;
      if z < max[bwidth] then pushmax(bwidth);
      z:=2*z1-2*min[bwidth];
      z:=(z1*(z1-1)+z-1) div z;
      if z1 <= 2*min[bwidth] then
          begin
            if z > min[edges] then pushmin(edges);
            if max[edges] < infinity then
              begin
                z:=2*max[edges]+1;
                z1:=z*z-8*max[edges]*min[bwidth];
                if z1 >= 0 then
                   begin
                     z:=round((z-sqrt(z1))/2+hf);
                     if z > min[nodes] then pushmin(nodes);
                   end;
               end;
          end
         else
           if max[edges] < z then
             begin
               z:=2*min[bwidth]+1;
               if z > min[nodes] then pushmin(nodes);
             end;
    end;
end;

procedure r337;
(*************************************************)
(*                                               *)
(* if girth >= 5 then dom <= (2p-x(4e/p-x-3))/4 *)
(*                 x = mindeg-1                  *)
(*                                               *)
(*************************************************)
var z2,z3:longint;
begin
  if (activerule[337]) and (max[girth] >= 5) then
    begin
      rule:='337/ ';
      z1:=max[nodes];
      if z1 < infinity then
         begin
           z:=min[mindeg]-1;
           z:=trunk((2*z1-z*(4*min[edges]/z1-z-3))/4);
         end
        else z:=min[dom];
      if min[girth] >= 5 then
        begin
          if z1 < infinity then
            begin
              if z < max[dom] then pushmax(dom);
              z:=min[mindeg]-1;
```

```
            if z > 0 then
              begin
                z:=(2*z1*z1+(z*(z+3)-4*min[dom])*z1) div (4*z);
                z2:=max[mindeg]-1;
                z3:=(2*z1*z1+(z2*(z2+3)-4*min[dom])*z1) div (4
*z2);
                if z3 > z then z:=z3;
                if z < max[edges] then pushmax(edges);
              end;
            rz:=4*min[edges]/z1-3;
            rhb:=rz*rz-8*z1+16*min[dom];
            if rhb > 1 then
               begin
                 z:=1+trunk((rz-sqrt(rhb))/2);
                 if z < max[mindeg] then pushmax(mindeg);
               end;
          end;
        (************************************)
        (*   p is not bounded because it is   *)
        (*   unclear which value of mindeg    *)
        (*   should be used.                  *)
        (************************************)
      end
     else
       if min[dom] > z then
            begin
              z:=4;
              pushmax(girth);
            end;
  end;
end;

procedure r338;
(************************************************)
(*                                              *)
(* if connected, girth >= 5, mindeg >= 4 then   *)
(*       dom <= (p-maxdeg-x)/2                  *)
(*                x=mindeg(mindeg-3)/2          *)
(*                                              *)
(************************************************)
begin
  if (activerule[338]) and (max[girth] >= 5) and
     (max[connct] = 1) and (max[mindeg] >= 4) then
    begin
      rule:='338/ ';
      if (min[girth] >= 5) and (min[connct] = 1)
         and (max[nodes] < infinity) then
          begin
            z:=8*max[nodes]-16*min[dom]-8*min[maxdeg]+9;
            if z >= 0 then
               begin
                 z:=trunk((3+sqrt(z))/2);
```

```
                if z < 3 then z:=3;
                if z < max[mindeg] then pushmax(mindeg);
              end;
          end;
      z1:=min[mindeg];
      if (min[girth] >= 5) and (min[connct] = 1) and
         (min[mindeg] >= 4) then
          begin
            z:=2*min[dom]+min[maxdeg]+(z1*(z1-3)) div 2;
            if z > min[nodes] then pushmin(nodes);
          end;
      if max[nodes] < infinity then
         begin
           z:=(max[nodes]-min[maxdeg]-(z1*(z1-3)) div 2) div 2;
           if (min[girth] >= 5) and (min[connct] = 1) and
              (min[mindeg] >= 4) then
               begin
                 if z < max[dom] then pushmax(dom);
                 z:=max[nodes]-2*min[dom]-(z1*(z1-3)) div 2;
                 if z < max[maxdeg] then pushmax(maxdeg);
               end
             else
               if (min[connct] >= 1) and (z1 >= 4)
                   and (min[dom] > z) and (min[dom] >=
max[mindeg]-1) then
                    begin
                      z:=4;
                      pushmax(girth);
                    end
                  else
                    if (min[girth] >= 5) and (z1 >= 4) and
                       (min[dom] > z) then
                         begin
                           z:=0;
                           pushmax(connct);
                         end;
          end;
    end;
end;

procedure r339;
(*********************************************)
(*                                           *)
(*    if girth >= 5 then dom >= mindeg*Ncomp    *)
(*                                           *)
(*********************************************)
begin
  if (activerule[339]) and (max[girth] >= 5) then
    begin
      rule:='339/ ';
      if min[girth] >= 5 then
         begin
```

```
            z:=min[mindeg]*min[ncomp];
            if z > min[dom] then pushmin(dom);
            if max[dom] < infinity then
               begin
                 z:=max[dom] div min[ncomp];
                 if z < max[mindeg] then pushmax(mindeg);
                 z:=max[dom] div min[mindeg];
                 if z < max[ncomp] then pushmax(ncomp);
               end;
          end
        else
          if max[dom] < min[mindeg]*min[ncomp] then
             begin
               z:=4;
               pushmax(girth);
             end;
    end;
end;

procedure r340;
(*************************************************)
(*                                               *)
(* if girth >= 6 then dom >= 2(mindeg-1)         *)
(*                                               *)
(*************************************************)
begin
  if (activerule[340]) and (max[girth] >= 6) then
    begin
      rule:='340/ ';
      if min[girth] >= 6 then
         begin
           z:=2*(min[mindeg]-1);
           if z > min[dom] then pushmin(dom);
           if max[dom] < infinity then
             begin
               z:=max[dom] div 2+1;
               if z < max[mindeg] then pushmax(mindeg);
             end;
          end
        else
         if max[dom] < 2*(min[mindeg]-1) then
            begin
              z:=5;
              pushmax(girth);
            end;
    end;
end;

procedure r341;
(*****************************************************)
(*                                                   *)
(* if mindeg >= 2 and girth >= 7 then dom >= maxdeg+1  *)
```

```
(*                                                    *)
(******************************************************)
begin
  if (activerule[341])and (max[girth] >= 7) and (max[mindeg] >=
2) then
    begin
      rule:='341/ ';
      if (min[mindeg] >= 2) and (min[girth] >= 7) then
           begin
             z:=min[maxdeg]+1;
             if z > min[dom] then pushmin(dom);
             if max[dom] < infinity then
                begin
                  z:=max[dom]-1;
                  if z < max[maxdeg] then pushmax(maxdeg);
                end;
           end
        else
          if (max[dom] < min[maxdeg]+1) and (min[mindeg] >= 2)
then
              begin
                z:=6;
                pushmax(girth);
              end
         else
           if (min[girth] >= 7) and (max[dom] < min[maxdeg]+1)
then
              begin
                z:=1;
                pushmax(mindeg);
              end;
    end;
end;

procedure r342;
(************************************************)
(*                                              *)
(*  if 5 <= girth <= p/2 then                   *)
(*         e <= (p*p-p*g+2*g) div g             *)
(*                                              *)
(************************************************)
begin
  if (activerule[342]) and (max[nodes] < infinity)
      and (min[girth] < infinity) and (max[girth] >= 5) then
    begin
      rule:='342/ ';
      z1:=max[nodes];
      if 2*max[girth] <= min[nodes] then
          begin
            z:=z1*z1 div (min[edges]+z1-2);
            if z < 4 then z:=4;
            if z < max[girth] then pushmax(girth);
```

```
          end;
      z:=min[girth];
      z:=(z1*z1) div z -z1+2;
      if (min[girth] >= 5) and (2*max[girth] <= min[nodes]) then
         begin
           if z < max[edges] then pushmax(edges);
           z:=min[girth];
           if z <= min[nodes] div 2 then
             begin
               z:=round((z+sqrt(z*z+4*z*(min[edges]-2)))/2+hf);
               if z > min[nodes] then pushmin(nodes);
             end;
         end
        else
          if (min[edges] > z) and (min[girth] >= 5) then
                     begin
                       z:=min[nodes] div 2+1;
                       if z > min[girth] then pushmin(girth);
                       z:=2*max[girth]-1;
                       if z < max[nodes] then pushmax(nodes);
                     end;
    end;
end;

procedure r343;
(*********************************************)
(*                                           *)
(* if girth >= 5 then e<=(p*sqrt(p-1))/2     *)
(*                                           *)
(*********************************************)
var z2:longint;
begin
  if (activerule[343]) and (max[girth] >= 5) then
    begin
      rule:='343/ ';
      z1:=max[nodes];
      z:=trunk(z1*sqrt(z1-1)/2);
      if min[girth] >= 5 then
         begin
           if (z1 < infinity) and (z < max[edges]) then
pushmax(edges);
           z1:=min[nodes];
           z2:=4*min[edges]*min[edges];
           rz:=z2+z1*z1;
           z:=round(root(rz,3)+hf);
           if z1 < z then
             begin
               k:=0;
               while (z1 < z) and (k < 100) do
                   begin
                     k:=k+1;
                     z1:=z;
```

```
                    rz:=z2+z*z;
                    z:=round(root(rz,3)+hf);
                  end;
               pushmin(nodes);
             end;
         end
       else
         if (z1 < infinity) and (min[edges] > z) then
            begin
              z:=4;
              pushmax(girth);
            end;
    end;
end;

procedure r344;
(*********************************************)
(*                                           *)
(*  if not a forest and                      *)
(*   p >= (3g-3) div 2, then e <= p*(p-1) div x *)
(*                                  - Ncomp + 1  *)
(*          where x = (3g-5) div 2           *)
(*                                           *)
(*********************************************)
begin
  if (activerule[344]) and (max[nodes] < infinity)
     and (min[girth] < infinity) then
    begin
      rule:='344/ ';
      z1:=max[nodes];
      z:=(z1*(z1-1)) div ((3*min[girth]-5) div 2)-min[ncomp]+1;
      if (max[girth] < infinity) and
         (min[nodes] >= (3*max[girth]-3) div 2) then
         begin
           if z < max[edges] then pushmax(edges);
           z:=z+min[ncomp]-min[edges];
           if z < max[ncomp] then pushmax(ncomp);
           if max[girth] < infinity then
              begin
                z:=(2*((z1*(z1-1)) div
(min[edges])+min[ncomp]-1)) div 3+2;
                if z < max[girth] then pushmax(girth);
              end;
           z:=(3*min[girth]-5) div 2;
           z:=round((1+sqrt(4*(min[edges]+min[ncomp]-1)*z+1))/2
+hf);
           if z > min[nodes] then pushmin(nodes);
         end
      else
        if min[edges] > z then
           begin
             if max[girth] < infinity then
```

```
              begin
                z:=(3*max[girth]-3) div 2-1;
                if z < max[nodes] then pushmax(nodes);
              end;
            z:=(2*min[nodes]+7) div 3;
            if z > min[girth] then pushmin(girth);
          end;
    end;
end;

procedure r345;
(*************************************************)
(*                                               *)
(*   if not a forest and                         *)
(*     t = (g-1) div 2 >= 2 then                 *)
(*        e <= p*m**(1/t)+p-1                    *)
(*             where m = max[1,(p-2**(t+1)+8)/4]  *)
(*                                               *)
(*************************************************)
var t:longint;
    m:real;
begin
  if (activerule[345]) and (max[nodes] < infinity)
     and (min[girth] < infinity) and (min[girth] >= 5) then
    begin
      rule:='345/ ';
      z1:=max[nodes];
      t:=(min[girth]-1) div 2;
      if t >= 2 then
       begin
         power(2,t+1,z);
         if z < infinity then
         begin
           m:=(z1-z+8)/4;
           if m < 1 then m:=1;
           z:=trunk(z1*root(m,t)+z1-1);
           if z < max[edges] then pushmax(edges);
           rhb:=(min[edges]-z1+1)/z1;
           if rhb > 1 then
             begin
               z:=trunk(2*log2(m)/log2(rhb))+2;
               if z < max[girth] then pushmax(girth);
             end;
          end;
       end;
    end;
end;

procedure r346;
(*************************************************)
(*                                               *)
(*  if g exists and (nconn > 0 or mindeg > 1)    *)
```

```
(*    then  2*genus >= e*x+2*Ncomp                    *)
(*            where x = (1-2/g-2/mindeg)              *)
(*                                                    *)
(******************************************************)
var z2:integer;
begin
  if (activerule[346]) and (min[forest]=0) and
     ((max[nconn] > 0) or (max[mindeg] > 1)) then
    begin
      rule:='346/ ';
       z1:=(min[girth]-2)*(min[mindeg]-2)-4;
      rz:=min[edges]*z1/(min[girth]*min[mindeg]);
       if z1 <= 0 then z:=0
                  else z:=round(rz/2+hf)+min[ncomp];

      if (max[forest]=0) and ((min[nconn] > 0) or (min[mindeg] >
1)) then
        begin
          if z > min[genus] then pushmin(genus);
          if max[genus] < infinity then
           begin
             z:=max[genus]-z+min[ncomp];
             if z < max[ncomp] then pushmax(ncomp);
                 z2:=2*(max[genus]-min[ncomp]);
             if (z1 > 0) and (z2 >0) then
               begin
                 z:=trunk(z2*min[girth]*min[mindeg]/z1);
                 if z < max[edges] then pushmax(edges);
               end;
             if z2 > 0 then
              begin
                rz:=min[edges]*(1-2/min[girth])-z2;
                if rz > 0 then
                    begin
                      z:=trunk((2*min[edges])/rz);
                      if z < max[mindeg] then pushmax(mindeg);
                    end;
                z:=(min[mindeg]-2)*min[edges];
                z:=z-min[mindeg]*z2;
                if z > 0 then
                  begin
                    z:=(2*min[mindeg]*min[edges]) div z;
                    if z < max[girth] then pushmax(girth);
                  end;
              end
                 else
                   begin
                     if z2 < 0 then z2:=3
                               else z2:=4;
                     z:=2+z2 div (min[girth]-2);
                       if z < max[mindeg] then pushmax(mindeg);
                       if min[mindeg] >= 3 then
```

```
                            begin
                              z:=2+z2 div (min[mindeg]-2);
                              if z < max[girth] then
pushmax(girth);
                             end;
                    end;
            end;
          end
        else
           if max[genus] < z then
              begin
                z:=1;
                if max[forest]=0 then
                  begin
                    if z < max[mindeg] then pushmax(mindeg);
                    z:=0;
                    if z < max[nconn] then pushmax(nconn);
                  end
                 else
                   if (min[nconn] > 0) or (min[mindeg] > 1) then
pushmin(forest);
               end;
    end;
end;

procedure r347;
(*********************************************)
(*                                           *)
(*  if diam = 2 then p <= nconn*maxdeg+1     *)
(*                                           *)
(*********************************************)
begin
  if (activerule[347]) and (min[diam] <= 2) and (max[maxdeg]
< infinity)
     and (max[nconn] < infinity) then
    begin
      rule:='347/ ';
      if max[diam]=2 then
          begin
            z:=max[maxdeg]*max[nconn]+1;
            if z < max[nodes] then pushmax(nodes);
            z:=(min[nodes]-2) div max[maxdeg]+1;
            if z > min[nconn] then pushmin(nconn);
            z:=(min[nodes]-2) div max[nconn]+1;
            if z > min[maxdeg] then pushmin(maxdeg);
          end
        else
          if min[nodes] > max[maxdeg]*max[nconn]+1 then
             begin
               z:=3;
               pushmin(diam);
             end;
```

```
    end;
end;

procedure r348;
(*************************************************)
(*                                               *)
(* if not a forest and                           *)
(*   E >= P+2-Nc then girth <= 2*(P+3-2*Nc)/3    *)
(*                                               *)
(*************************************************)
begin
  if (activerule[348]) and (max[nodes] < infinity)
     and (min[girth] < infinity) and (min[forest] = 0) then
    begin
      rule:='348/ ';
      z1:=(3*min[girth]+1) div 2+2*min[ncomp]-3;
      z:=min[edges]-1+min[ncomp];
      if z > z1 then z:=z1;
      if max[forest] = 0 then
        begin
          if z > min[nodes] then pushmin(nodes);
          z1:=(2*max[nodes]-3*min[girth]+6) div 4;
          z:=max[nodes]+1-min[edges];
          if z < z1 then z:=z1;
          if z < max[ncomp] then pushmax(ncomp);
          z:=max[nodes];
          z:=(2*(z+3-2*min[ncomp])) div 3;
          if min[edges] >= max[nodes]+2-min[ncomp] then
               begin
                 if z < max[girth] then pushmax(girth);
               end
             else
               if min[girth] > z then
                begin
                  z:=max[nodes]+1-min[ncomp];
                  if z < max[edges] then pushmax(edges);
                end;
         end
       else
        if max[nodes] < z then
            begin
              z:=1;
              pushmin(forest);
            end;
    end;
end;

procedure r349;
(*************************************************)
(*                                               *)
(*  if not a forest and                          *)
(*     E >= p+3-Nc then girth <= p/2+2-Nc        *)
```

```
(*                                                *)
(**************************************************)
begin
  if (activerule[349]) and (max[nodes] < infinity)
     and (min[girth] < infinity) and (min[forest] = 0) then
    begin
      rule:='349/ ';
      z1:=2*min[girth]-4+2*min[ncomp];
      z:=min[edges]-2+min[ncomp];
      if z > z1 then z:=z1;
      if max[forest] = 0 then
        begin
          if z > min[nodes] then pushmin(nodes);
          z1:=max[nodes] div 2+2-min[girth];
          z:=max[nodes]+2-min[edges];
          if z < z1 then z:=z1;
          if z < max[ncomp] then pushmax(ncomp);
          if min[edges] >= max[nodes]+3-min[ncomp] then
              begin
                z:=max[nodes] div 2+2-min[ncomp];
                if z < max[girth] then pushmax(girth);
              end
           else
             if min[girth] > max[nodes] div 2+2-min[ncomp]   then
                begin
                  z:=max[nodes]+2-min[ncomp];
                  if z < max[edges] then pushmax(edges);
                end;
         end
        else
          if max[nodes] < z then
              begin
                z:=1;
                pushmin(forest);
              end;
    end;
end;

procedure r350;
(**************************************************)
(*                                                *)
(* if not a forest and                            *)
(*    e >= p+4-Nc then p >= 9*g/4+2*Nc-5          *)
(*                                                *)
(**************************************************)
begin
  if (activerule[350]) and (max[nodes] < infinity)
     and (min[girth] < infinity) and (min[forest] = 0) then
    begin
      rule:='350/ ';
      z1:=(9*min[girth]+3) div 4+2*min[ncomp]-5;
      z:=min[edges]-3+min[ncomp];
```

```
      if z > z1 then z:=z1;
      if max[forest] = 0 then
         begin
           if z > min[nodes] then pushmin(nodes);
           z1:=(4*max[nodes]-9*min[girth]+20) div 8;
           z:=max[nodes]+3-min[edges];
           if z < z1 then z:=z1;
           if z < max[ncomp] then pushmax(ncomp);
           z:=max[nodes];
           if min[edges] >= z+4-min[ncomp] then
                  begin
                    z:=(4*z-8*min[ncomp]+20) div 9;
                    if z < max[girth] then pushmax(girth);
                  end
                else
                  if 4*z < 8*min[ncomp]+9*min[girth]-20 then
                     begin
                       z:=z+3-min[ncomp];
                       if z < max[edges] then pushmax(edges);
                     end;
           end
         else
           if max[nodes] < z then
              begin
                z:=1;
                pushmin(forest);
              end;
    end;
end;


end.
```