

```

unit rules500;
  interface
    uses
      globals,cmmnds1,pusherr,pushStack,ruleAtoF;

    procedure r451;procedure r452;procedure r453;procedure
r454;procedure r455;
    procedure r456;procedure r457;procedure r458;procedure
r459;procedure r460;

  implementation

  procedure r451;
  (*****
  (
  (
  (*   if mindeg <= p-2-MAX{nind,5} then
  (*       mindeg <= ((p-1)*(clique-1)-2)/clique
  (
  (
  (*****
begin
  if (activerule[451]) and (max[nodes] < infinity) then
    begin
      rule:='451/ ';
      z:=min[nodes]-max[nind]-2;
      if z > min[nodes]-7 then z:=min[nodes]-7;
      if max[mindeg] <= z then
        begin
          if max[clique] < infinity then
            begin
              z:=((max[nodes]-1)*(max[clique]-1)-2) div
max[clique];
              if z < max[mindeg] then pushmax(mindeg);
              z:= (min[mindeg]*max[clique]+1) div
(max[clique]-1)+2;
              if z > min[nodes] then pushmin(nodes);
            end;
            z:=max[nodes] div (max[nodes]-1-min[mindeg])+1;
            if z > min[clique] then pushmin(clique);
          end
        else
          if max[clique] < infinity then
            begin
              z:=((max[nodes]-1)*(max[clique]-1)-2) div
max[clique];
              if min[mindeg] > z then
                begin
                  z:=min[nodes]-max[nind]-1;
                  if z > min[nodes]-6 then z:=min[nodes]-6;
                  if z > min[mindeg] then pushmin(mindeg);

```

```

        z:=max[mindeg]+max[nind]+1;
        if z < max[mindeg]+6 then z:=max[mindeg]+6;
        if z < max[nodes] then pushmax(nodes);
        if max[mindeg] <= min[nodes]-7 then
            begin
                z:=min[nodes]-max[mindeg]-1;
                if z > min[nind] then pushmin(nind);
            end;
        end;
    end;
end;

end;
end;

procedure r452;
(*****
(*)
(*) if nind = 2 then clique >= p(xln(x)-x+1)/(x-1)**2 (*)
(*) where x = p-1-2e/p (*)
(*) and ln(x) is the natural log. of x (*)
(*)
(*****)
begin
    if (activerule[452]) and (min[nind] <= 2) then
        begin
            rule:='452/ ';
            rz:=min[nodes]-1-2*max[edges]/min[nodes];
            if rz > 1 then
                begin
                    rhb:=0.693143*log2(rz);
                    z:=round(min[nodes]*(rz*rhb-rz+1)/((rz-1)*(rz-1))+hf);
                    if min[nind] = max[nind] then
                        begin
                            if z > min[clique] then pushmin(clique);
                        end
                    else
                        if max[clique] < z then
                            if min[nind] = 2 then
                                begin
                                    z:=3;
                                    pushmin(nind);
                                end
                            else
                                if max[nind] = 2 then
                                    begin
                                        z:=1;
                                        pushmax(nind);
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

procedure r453;

```

```

(*****
*)
(*)
(*)
(*)  e <= p(p-1)/2-m(p-nccov)-nccov(nccov-1)/2+(clique-1)m(m+1)/2
*)
(*)  where  m = (p-nccov) div (clique-1)
*)
(*)
*)
(*****
*)
var z2:longint;
begin
  if (activerule[453]) and (max[nodes] < infinity) and
    (min[nccov] = max[nccov]) and (min[clique] = max[clique])
  then
    begin
      rule:='453/ ';
      z:=min[nccov];
      z:=(z*(z-1)) div 2;
      z2:=max[nodes];
      z1:=(z2-min[nccov]) div (min[clique]-1);
      z2:=(z2*(z2-1)) div 2;
      z:= z2-z1*(max[nodes]-max[nccov])-z+(max[clique]-1)*((z1
*(z1+1)) div 2);
      if z < max[edges] then pushmax(edges);
    end;
  end;

procedure r454;
(*****
*)
(*)
(*)  if nind <= 2 and p <= mindeg+4 then
(*)
(*)  e <= p(p-14)/2+14*clique
(*)
(*)
(*****
*)
begin
  if (activerule[454]) and (min[nind] <= 2) and (max[nodes] >=
14) then
    begin
      rule:='454/ ';
      z:=max[nodes];
      if max[nodes] < infinity then
        begin
          z1:=(z*(z-14)) div 2;
          if (max[nind] <= 2) and (min[mindeg] >= max[nodes]-4)
then
            begin
              z:=(min[edges]-z1+13) div 14;
              if z > min[clique] then pushmin(clique);
            end;

```

```

        end;
    if max[clique] < infinity then
        begin
            z:=2*min[edges]+49-28*max[clique];
            if z >= 0 then
                begin
                    z:=7+round(sqrt(z)+hf);
                    if z > min[mindeg]+5 then z:=min[mindeg]+5;
                    if (max[nind] <= 2) and (z > min[nodes]) then
pushmin(nodes);
                end;
            if max[nodes] < infinity then
                if max[nind] <= 2 then
                    begin
                        if min[mindeg] >= max[nodes]-4 then
                            begin
                                z:=z1+14*max[clique];
                                if z < max[edges] then pushmax(edges);
                            end
                        else
                            if min[edges] > z1+14*max[clique] then
                                begin
                                    z:=max[nodes]-5;
                                    if z < max[mindeg] then
pushmax(mindeg);
                                end;
                            end
                        else
                            if (max[nodes] <= min[mindeg]+4)
                                and (min[edges] > z1+14*max[clique]) then
                                begin
                                    z:=3;
                                    pushmin(nind);
                                end;
                            end;
                    end;
                end;
            end;
        end;

procedure r455;
(*****
(*)
(*) if nind <= 2 and mindeg >= p-3 then (*)
(*) e <= p(p-15)/2+15*clique (*)
(*)
(*****)
begin
    if (activerule[455]) and (min[nind] <= 2) and (max[nodes] >=
15) then
        begin
            rule:='455/ ';
            z:=max[nodes];
            if max[nodes] < infinity then

```

```

begin
  z1:=(z*(z-15)) div 2;
  if (max[nind] <= 2) and (min[mindeg] >= max[nodes]-3)
then
  begin
    z:=(min[edges]-z1+14) div 15;
    if z > min[clique] then pushmin(clique);
  end;
end;
if max[clique] < infinity then
begin
  z:=8*min[edges]+225-120*max[clique];
  if z >= 0 then
    begin
      z:=round((15+sqrt(z)/2)+hf);
      if z > min[mindeg]+4 then z:=min[mindeg]+4;
      if (max[nind] <= 2) and (z > min[nodes]) then
pushmin(nodes);
    end;
    if max[nodes] < infinity then
      if max[nind] <= 2 then
        begin
          if min[mindeg] >= max[nodes]-3 then
            begin
              z:=z1+15*max[clique];
              if z < max[edges] then pushmax(edges);
            end
          else
            if min[edges] > z1+15*max[clique] then
              begin
                z:=max[nodes]-4;
                if z < max[mindeg] then
pushmax(mindeg);
              end;
            end
          else
            if (max[nodes] <= min[mindeg]+3)
              and (min[edges] > z1+15*max[clique]) then
              begin
                z:=3;
                pushmin(nind);
              end;
            end;
          end;
        end;
      end;
    end;

procedure r456;
( ***** )
( * )
( * if clique = 2 then chr <= (3ncov+52)/16 )
( * )
( ***** )

```

```

begin
  if (activerule[456]) and (min[clique] <= 2) then
    begin
      rule:='456/ ';
      if max[clique] = 2 then
        begin
          if max[ncov] < infinity then
            begin
              z:=(3*max[ncov]+52) div 16;
              if z < max[chr] then pushmax(chr);
            end;
            z:=(16*min[chr]-50) div 3;
            if z > min[ncov] then pushmin(ncov);
          end
        else
          if max[ncov] < infinity then
            if min[chr] > (3*max[ncov]+52) div 16 then
              begin
                z:=3;
                pushmin(clique);
              end;
            end;
          end;
        end;
      end;
    end;
  end;

procedure r457;
(*****)
(*                                           *)
(* if clique = 2 then chr <= (3p-3nind+52)/16 *)
(*                                           *)
(*****)
begin
  if (activerule[457]) and (min[clique] <= 2) then
    begin
      rule:='457/ ';
      if max[clique] = 2 then
        begin
          if max[nodes] < infinity then
            begin
              z:=(3*max[nodes]-3*min[nind]+52) div 16;
              if z < max[chr] then pushmax(chr);
              z:=(3*max[nodes]-16*min[chr]+52) div 3;
              if z < max[nind] then pushmax(nind);
            end;
            z:=(16*min[chr]+3*min[nind]-50) div 3;
            if z > min[nodes] then pushmin(nodes);
          end
        else
          if max[nodes] < infinity then
            if min[chr] > (3*max[nodes]-3*min[nind]+52) div
16 then
              begin
                z:=3;

```

```

        pushmin(clique);
    end;
end;

procedure r458;
(*****)
(*
(* if nind = 2 then nccov <= (3p-3clique+52)/16 *)
(*
(*****)
begin
    if activerule[458] then
        begin
            rule:='458/ ';
            if max[nind] = 2 then
                begin
                    if max[nodes] < infinity then
                        begin
                            z:=(3*max[nodes]-3*min[clique]+52) div 16;
                            if z < max[nccov] then pushmax(nccov);
                            z:=(3*max[nodes]-16*min[nccov]+52) div 3;
                            if z < max[clique] then pushmax(clique);
                        end;
                        z:=(16*min[nccov]+3*min[clique]-50) div 3;
                        if z > min[nodes] then pushmin(nodes);
                    end
                end
            else
                if max[nodes] < infinity then
                    if min[nccov] > (3*max[nodes]-3*min[clique]+52)
div 16 then
                        begin
                            z:=3;
                            pushmin(nind);
                        end;
                    end;
                end;
            end;
        end;

procedure r459;
(*****)
(*
(*
(*
(*****)
begin
    if activerule[459] then
        begin
            rule:='459/ ';
        end;
    end;

procedure r460;
(*****)

```

```

( * * )
( * * )
( ***** )
begin
  if activerule[460] then
    begin
      rule:='460/ ';
    end;
  end;
end;

end.

```