```
function [Q,Z,E,A,mcur,ncur,s,t] = Staircase(E,A,Q,Z,r,tol)
%
% function [Q,Z,E,A,df,s,t] = Staircase(E,A,Q,Z,r,tol)
%
% transforms a pencil [A1 A2]-s[E1 E2] where the second block has
% r columns, to a strict equivalent block triangular pencil
%
% [A11 A12 A13 A14]  [E11 E12 E13 E14]
% [ 0  A22 A23 A24]-s[ 0  E22 E23 E24]
% [ 0   0  A33 A34]  [ 0   0  E33 E34]
%
% where A22-sE22 contains the df zeros of the pencil A1-sE1 and
% the pencil [A11]-s[E11] contains its right minimal indices.
% The tolerance for the rank checks is tol.
%
% The routine returns the unitary transformations Q and Z
% and the sets of dimensions s and t of the diagonal blocks.
%
mn=size(E);mcur=mn(1);ncur=mn(2)-r;
s=zeros(1,0);t=zeros(1,0);
while 0 < mcur,
% Reduce the leading mcur x ncur pencil by one block
% First compress the rows of leading block E
[Qleft,Er,rE] = RowCompT(E(1:mcur,1:ncur),tol);
if mcur == rE, return, s=[s 0];t=[t 0]; end
E(1:mcur,1:ncur)=Er;
E(1:mcur,ncur+1:mn(2))=Qleft'*E(1:mcur,ncur+1:mn(2));
A(1:mcur,:)=Qleft'*A(1:mcur,:);
Q(:,1:mcur)=Q(:,1:mcur)*Qleft;
% Then compress the columns of the new leading block
[Qright,Ar,rA] = ColCompR(A(rE+1:mcur,1:ncur),tol);
A(rE+1:mcur,1:ncur)=Ar;
A(1:rE,1:ncur)=A(1:rE,1:ncur)*Qright;
E(1:rE,1:ncur)=E(1:rE,1:ncur)*Qright;
Z(:,1:ncur)=Z(:,1:ncur)*Qright;
% Now update the dimensions
s=[s, mcur-rE];t=[t rA];
mcur=rE;ncur=ncur-rA;
end
```