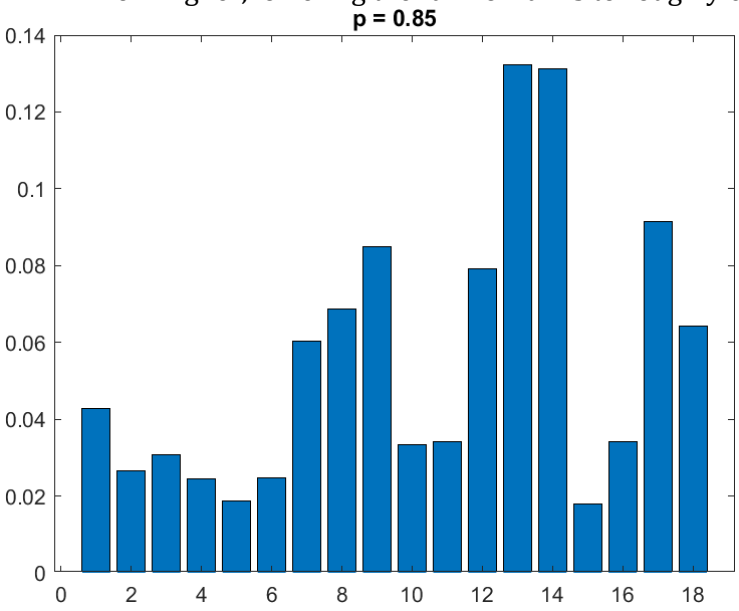
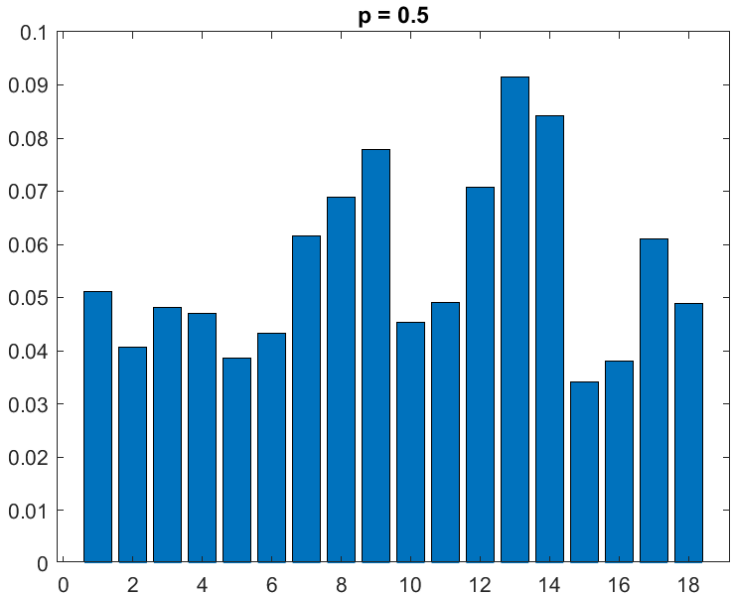
Rajan Grewal 301335629

MACM 316 – Assignment 6

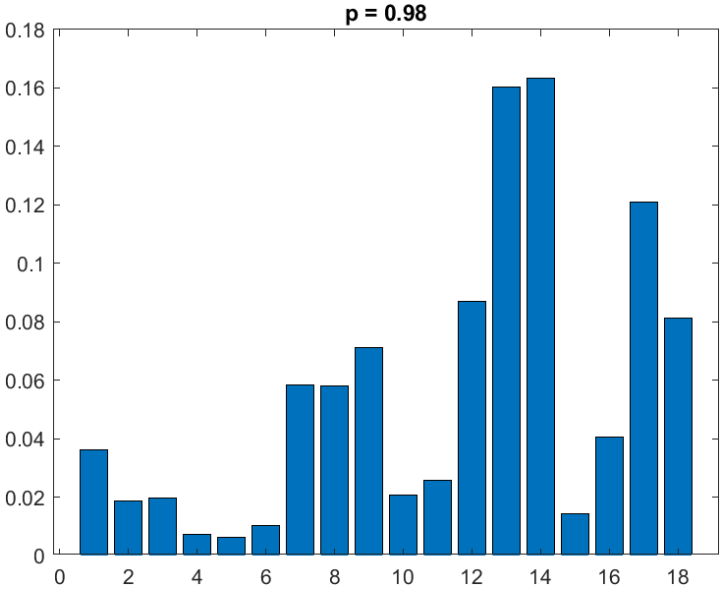
1. As p gets smaller, the power method converges faster and the websites start to have a similar rank. As p gets bigger, the power method takes more iterations to converge and the web pages get closer to their true rank, so a top page emerges. As p approaches 0, this means that we have a high chance of jumping to a random page since the probability of staying on a path is p, and the probability of randomly jumping is 1-p. When this value is low, we're practically always jumping to pages, so everything will have a rank of roughly 1/n since that is the probability of any given page being jumped on.

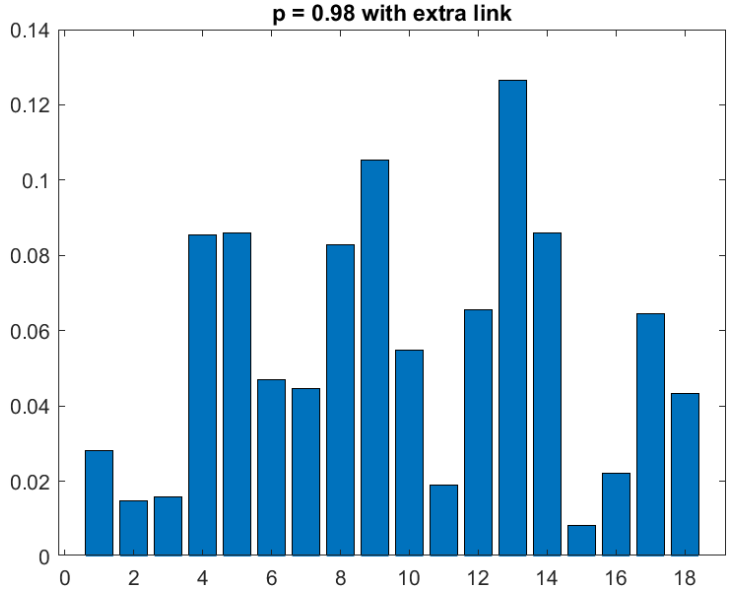
2. a) With p = 0.85, we get the following distribution, with a fairly large range of 0.12
   b) With p = 0.5, the previously high-ranked pages are now lower, and the previously low-ranked pages are now higher, lowering the rank of ranks to roughly 0.06.





c) The few low ranking pages are such due to their low number of links from other pages, and also the low ranks of the pages that point to these few low ranking pages. If I were to add a link to increase the rank of the lowest pages, I would add a link between the busiest page, page 13, and the quitest page, page 5. Since page 5 points to the second lowest ranked page, page 4, if will also significantly boost page 4's rank. This holds true judging from the bar graph on the bottom right.





Rajan Grewal 301335629

```
% Estimate the PageRank for a small web connectivity
% matrix using the power method.

p = 0.98;  % "teleport" probability

% Set up the connectivity matrix using index
% vectors that define the from-to connections
% between pages numbered 1 to n.
n = 18;     % number of pages

ii = [5 11 2 3 1 7 9 6 10 8 4 5 14 1 8 9 8 10 12 13 13 9 6 12 8 7 7 13 14 18 17 11 12 15 12 13 16 13 12 13 17]; % which page jj is connecting to
jj = [13 1 1 2 3 3 3 4 4 4 5 6 6 7 7 8 9 9 9 9 10 10 10 11 11 11 12 12 13 14 14 15 15 16 16 16 17 17 17 18 18]; % #outlinks
G = sparse(ii, jj, 1, n, n);

c = full(sum(G));  % column sums
e = ones(n,1);
k = find(c~=0);    % indices of zero columns
D = sparse(k, k, 1./c(k), n, n);
z = ((1-p)*(c~=0) + (c==0)) / n;

A = p*G*D + e*z;   % PageRank matrix

x = e/n;         % initial guess
xold = zeros(n,1);
niter = 0;
while norm(x-xold) > 0.0001,
  niter = niter + 1;
  xold = x;
  x = A*x;
end
x, niter
bar(x), shg    % plot ranks as a bar plot
title("p = 0.98 with extra link")
```