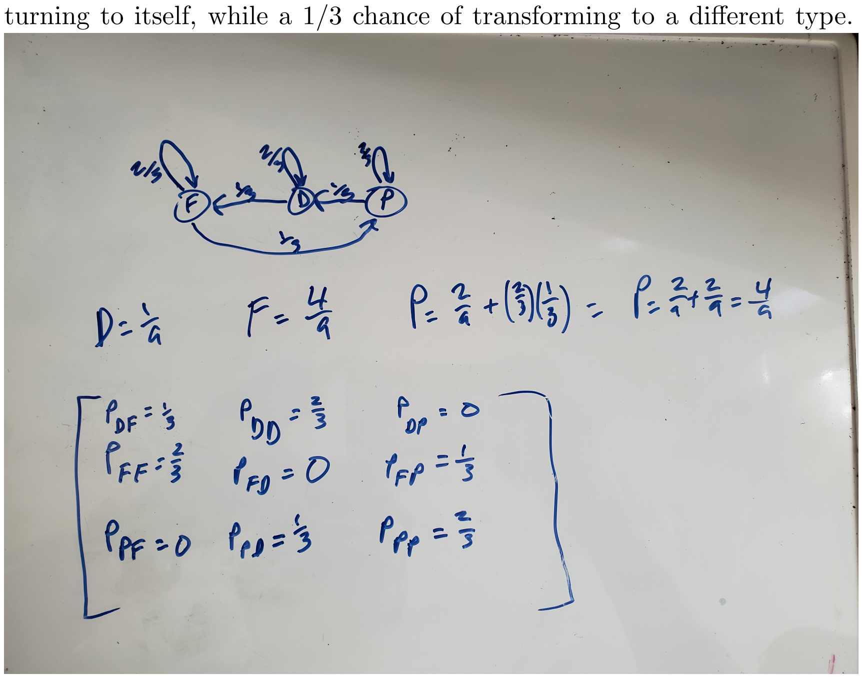
Homework 6

Kevin Chao

December 4 2019

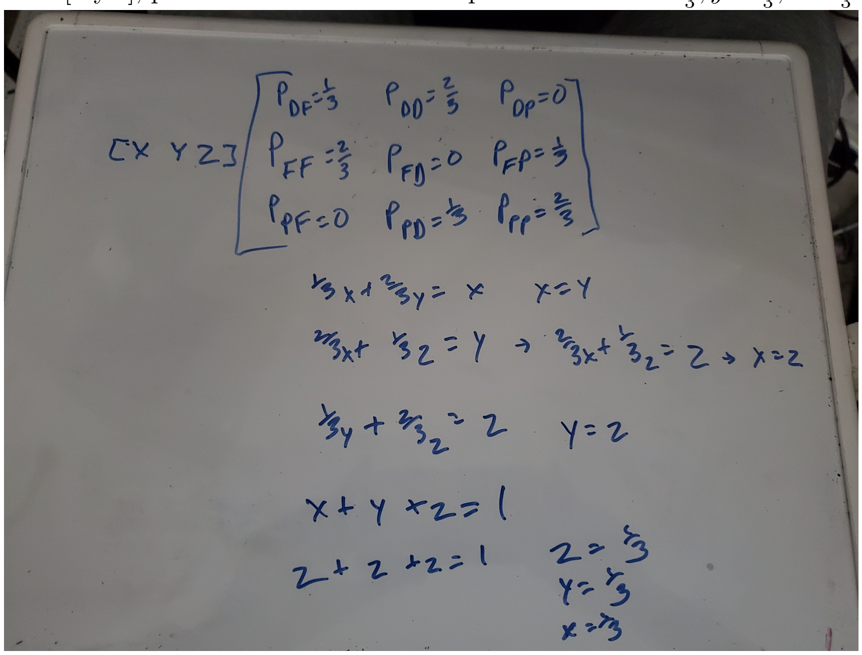
Question 1

a. Pictured below is the markov chain, the state transition matrix, and the probabilities of the states when n=2. This particular problem states that when Kecleon loses to a pokemon, it changes to that type. There is a "Rock Paper Scissors" Chain, where Fighting ¿ Dark, Dark ¿ Psychic, Psychic ¿ Fighting, and if Kecleon and the other pokemon are the same type, then it has a 50/50 chance of winning/losing. However, that means that if Kecleon is fighting the same type, no matter what it will stay that same type win or lose. Thus, each node has a 2/3 chance of returning to itself, while a 1/3 chance of transforming to a different type.

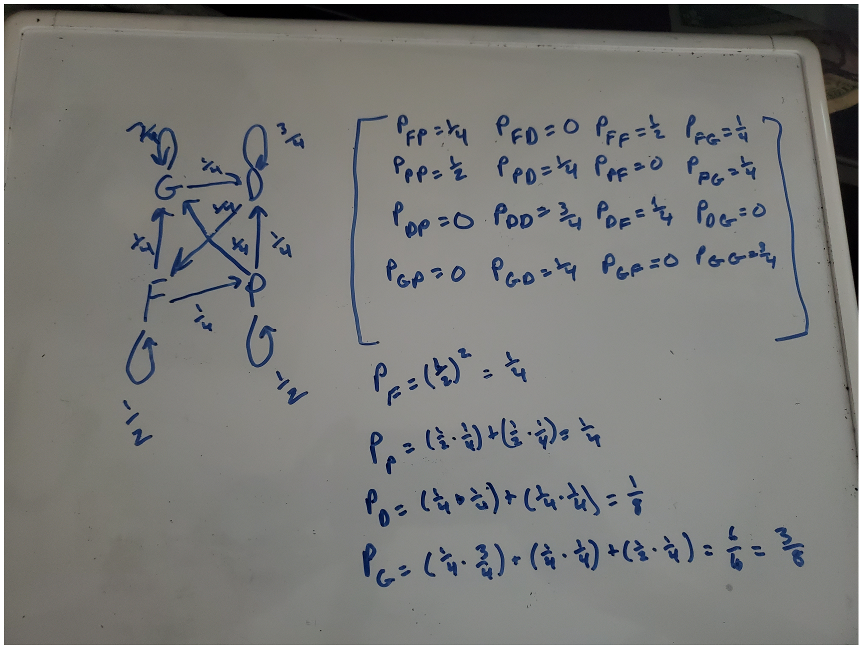


If the state starts with Fighting and can only go 2 steps, then it can only turn to Dark if it loses to a psychic and then loses to a dark, with both $\frac{1}{3}$ chance, making it a $\frac{1}{9}$. Fighting has a $\frac{2}{3}$ of staying with itself and the only way to keep it fighting is for this chance to occur twice, making it a $\frac{4}{9}$ chance. Finally, the probability that it turns into psychic has two possible situations: it turns to psychic 1 step then stays psychic the next, or it stays fighting 1 step and turns into psychic the next step, thus having $\frac{2}{9} + \frac{2}{9} = \frac{4}{9}$

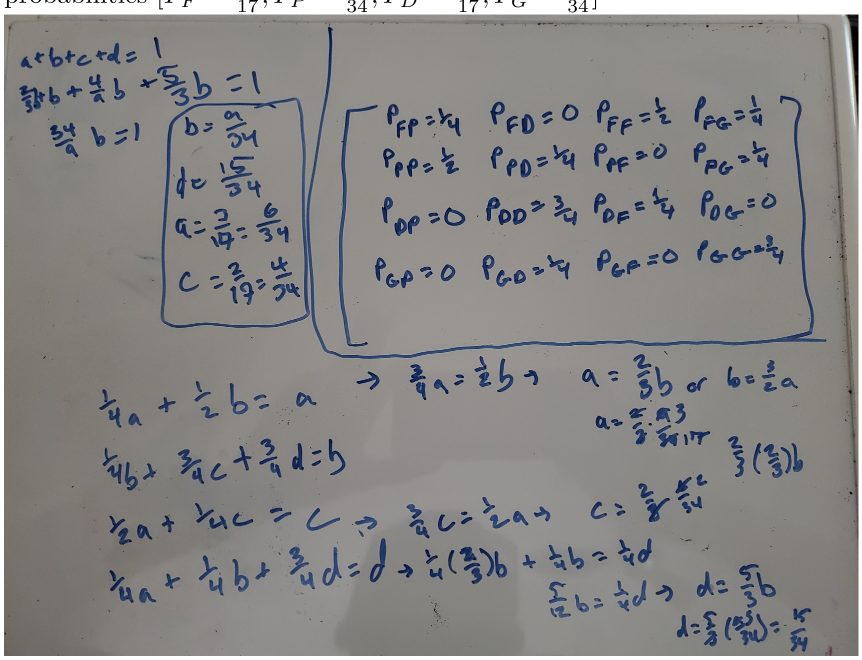
b. To find the steady-state distribution probabilities, you can do a system of equations with the state transition matrix. Using matrix multiplication with [x y z], pictured below is the three probabilities: $x = \frac{1}{3}, y = \frac{1}{3}, z = \frac{1}{3}$



c. It is nearly the same thing as (a) but now has an additional node. The picture below displays the drawing, the matrix, and the probabilities, which are $P_F = \frac{1}{4}, P_P = \frac{1}{4}, P_D = \frac{1}{8}, P_G = \frac{3}{8}$



d. It takes the same steps as (b) but is much more complicated. The picture below displays the matrix along with the process of how I arrived to the probabilities $[P_F = \frac{3}{17}, P_P = \frac{9}{34}, P_D = \frac{2}{17}, P_G = \frac{15}{34}]$



Question 2

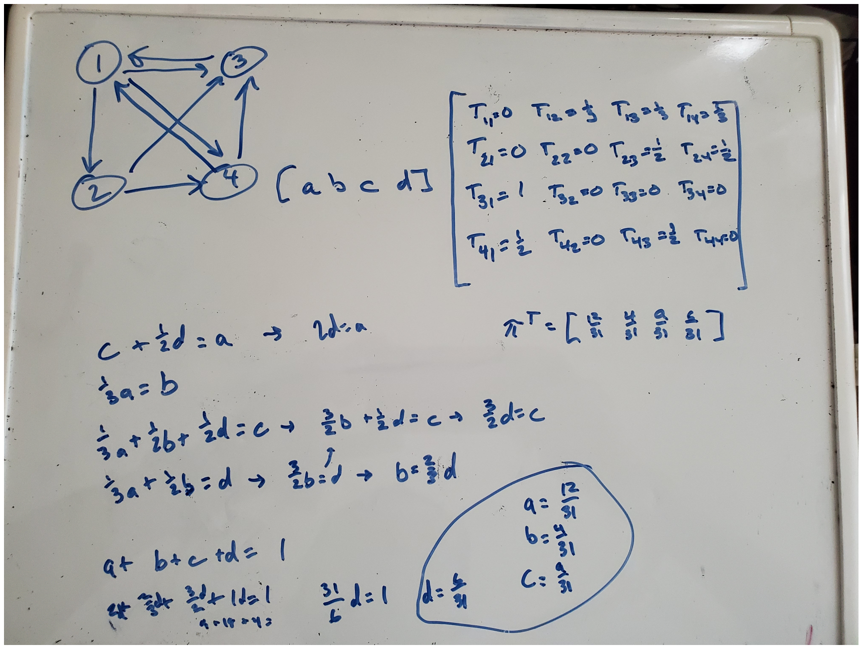
a. A king can move to any square at any path, there is no square to where the king is stuck making the same movements, and is thus irreducible.
The king can return to the same space it started on in 2 or more steps as it has the freedom to move. For example, lets say that it starts at (4,4), it can come back to (4,4) by doing
(4,4) - (4,5) - (4,4) OR (4,4) - (4,5) - (5,5) - (4,4).
And it could do it in many more steps if desired to do so, and is thus aperiodic.

b. A bishop moves diagonal only. However, that means that a bishop on the black squares can only move on the black squares and a bishop on the white square can only move on the white squares, and can never reach the other color. Treating the colors as recurrent classes in this case, then there are two recurrent classes and is thus not irreducible.
A bishop can come back to its original location in 2 or more steps if so desired, and is thus aperiodic.

c. A knight moves in an L shape. With that, each time it makes a movement, it has to move to a different color. Even if the path is complicated, it can still find a way to a specific coordinate (For example, if it starts at 2,2 and wants to be at 2,3, a different colored square, it can achieve that by going [2,2] - [4,3] - [5,1] - [2,3]). Thus, it is irreducible.
However, because of the fact that a knight changes from white to black or vice versa every legal steps, that would imply that at an odd number of steps, it would be impossible to return to its original state as it is a different color from. Thus, it is not aperiodic and has a period of 2.

Question 3

a. Below is the state transition matrix along with π^T. The highest pagerank would be webpage 1.



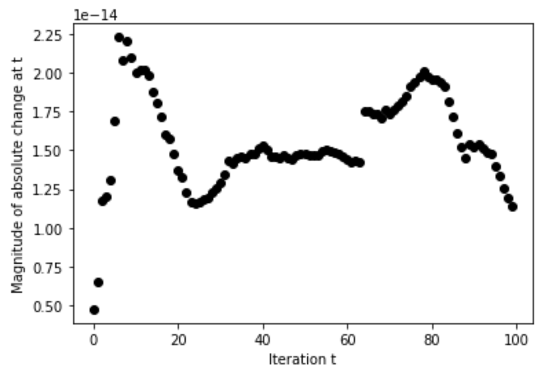
b.

```
def load_data():
    L = np.load('large_network.npz')['L']
    names = np.load('large_network.npz')['names'].tolist()
    return L, names

L, names = load_data()
m = np.shape(L)[0]   # number of websites (nodes)
T = np.array(L, np.float)
for i in range(m):
    s = np.sum(T[i])
    if s > 0:
        T[i] /= s
    else:
        T[i, i] = 1

p0 = np.ones(m) / m
G = (T*.85) + (p0 * .15)
```

c. Below is the results of the program, displaying the plot of magnitude vs iteration, top 25 sites, and steady state probability of the 25 sites.




```
pagerank        in      out     url:
0.00843         109     0       b'http://home.netscape.com/comprod/mirror/index.html'
0.00792         134     0       b'http://www.linkexchange.com/'
0.00577         199     0       b'http://www.yahoo.com/'
0.00534         5       0       b'http://eureka.ca.gov'
0.00503         20      0       b'http://www.rsac.org/'
0.00496         54      0       b'http://www.leginfo.ca.gov/bilinfo.html'
0.00396         53      0       b'http://www.adobe.com/prodindex/acrobat/readstep.html'
0.00393         29      0       b'http://home.netscape.com/comprod/mirror/
client_download.html'
0.00387         155     0       b'http://www.leginfo.ca.gov/calaw.html'
0.00355         42      0       b'http://www.weather.com/weather/us/states/
California.html'
0.00297         52      0       b'http://www.eff.org/blueribbon.html'
0.00291         42      0       b'http://www.dot.ca.gov/hq/roadinfo/i80'
0.00274         38      0       b'http://www.nws.noaa.gov/'
0.00268         35      0       b'http://www.csuchico.edu/~sbarker/scott.html'
0.00255         64      0       b'http://lcweb.loc.gov/homepage/lchp.html'
0.00249         29      0       b'http://www.floydsordeal.com/ring.htm'
0.00240         34      0       b'http://www.netscape.com'
0.00238         7       0       b'http://www.aaa.com'
0.00231         4       0       b'http://www.eli.net'
0.00229         35      0       b'http://cdip.ucsd.edu/models/wave.model.html'
0.00228         40      0       b'http://www.nasa.gov/'
0.00223         39      0       b'http://www.mip.berkeley.edu/cilc/brochure/
brochure.html'
0.00222         95      0       b'http://www.epa.gov/'
0.00219         57      0       b'http://www.stpt.com/'
0.00219         23      0       b'http://water.usgs.gov/'
Steady-state Probability is: 0.08957183181651172
```

It does appear that π_t does converge to a limit, as the magnitude of change seems to be going down even more after 100. Most of the webpages with the highest pagerank seems to be either .gov sites or fairly common sites for 2002 (i.e. yahoo.com and netscape). A lot of the webpages are either the main home page or a page where it displays instructions/basic information (index.html, readstep.html, etc.).

The code to find all of these is given below

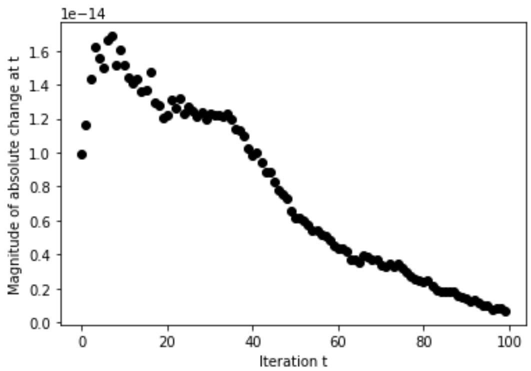
```
magnitudes = np.array([])
iterations = np.array([])
for i in range(100):
    iterations = np.append(iterations, i)
```

```
        if i == 0:
            pt = np.dot(p0,G)
            magnitudes = np.append(magnitudes,abs(np.sum(pt - p0)))
        else:
            prev = pt
            pt = np.dot(pt,G)
            magnitudes = np.append(magnitudes,abs(np.sum(pt - prev)))

    plt.plot(iterations, magnitudes, 'ok')
    plt.xlabel('Iteration t')
    plt.ylabel("Magnitude of absolute change at t")
    plt.show()

    rank_inds = np.argsort(pt)[::-1]
    rank_value = pt[rank_inds]
    steady = 0
    print('pagerank\tin\tout\turl:')
    for i in range(25):
        cur_ind = rank_inds[i]
        links_in  = np.sum(L[:, cur_ind])
        links_out = np.sum(L[cur_ind, :])
        print('%.5f\t\t%d\t%d\t%s' % (rank_value[i], links_in,
                links_out, names[rank_inds[i]]))
        steady += rank_value[i]
    print('Steady-state Probability total is: ' + str(steady))
```

d. Below is the result with the top 25 webpages, and their steady-state probabilities. In order to properly change from outgoing to incoming links, I applied transpose to the L matrix using np.transpose(L) and then created T matrix using the new L.




```
pagerank        in      out     url:
0.00182         0       43      b'http://www.geo.berkeley.edu/seismo/'
0.00136         0       19      b'http://www.cahwnet.gov/hwdc/governmt.htm'
0.00124         0       1       b'http://www.houseboat.net/aus.html'
0.00123         0       3       b'http://www.mother.com/SacHomePage/sachome.htm'
0.00122         0       16      b'http://www.findlaw.com/07cle/cle/3mcle_cal.html'
0.00121         0       2       b'http://www.clay.net/statag.html/'
0.00118         0       15      b'http://lana.lanminds.com/~wilworks/ehnglinx.htm'
0.00108         0       37      b'http://www.rahul.net/lpc/'
0.00082         0       12      b'http://www.pacific.net/~lawlib/topics.htm'
0.00082         0       11      b'http://www.tegnetcorporation.com/'
0.00078         4       164     b'http://www.water.ca.gov/www.gov.sites.html'
0.00073         0       1       b'http://sports.monterey.edu/'
0.00072         0       5       b'http://jobsmart.org/socal/adjobs/orgweb.htm'
0.00072         0       43      b'http://www.water.ca.gov/Env.html'
0.00065         0       8       b'http://www.politicalaccess.com/medialinks.htm'
0.00062         0       5       b'http://it.ucdavis.edu/service.html'
0.00060         0       3       b'http://www.washingtonpost.com/wp-srv/politics/
campaigns/keyraces98/cagov/parties.htm'
0.00058         0       1       b'http://www.weatherpost.com/wp-srv/weather/cities/
591.htm'
0.00057         0       11      b'http://www.pacificnet.net/~cooperag/'
0.00056         0       9       b'http://www.healthlaw.org/california.html'
0.00056         0       19      b'http://www.scec.gps.caltech.edu/'
0.00055         0       10      b'http://www.yahoo.com/Government/Politics/Elections/
Regional/U_S__States/California/Complete_Listing/index.html'
0.00055         0       10      b'http://www.pe.net/storm/html/links.html'
0.00052         0       3       b'http://www.la.yahoo.com/Transportation/Mass_Transit'
0.00049         0       1       b'http://lmc.tradewave.com/galaxy/Community/World-
Communities/North-America/United-States/California.html'
Steady-state Probability is: 0.021196837157045765
```

A big difference when it comes to the site is how specific the webpages are, to which it probably needs to have many links for sources and going back to the main webpage (for example, the webpage for www.water.ca.gov/www.gov.site.html would contain a list of webpages). The overall probabilities of the 25 top webpages is also significantly lower, going from .09 to .02 . To me, the ranking from part (c) seems much more sensible, as for 2002 I would probably be more likely to go onto the yahoo main page rather than a specific subsection of mass transit in LA.