- Previous Lecture:
  - Recursion
  - Efficiency

- Today's Lecture:
  - Models
    - Quantifying fairness
    - Quantifying importance
  - Simulation—Google "Page Rank"

Announcements

- P7 due today at 11pm
- Final exam: 12/14 (Mon) 7pm at Barton (indoor field) West
- Please fill out course evaluation on-line (Engineering will email about it, see "Exercise15")
- Regular office (consulting) hours end tomorrow (today). "Study Break" hours start next week.
- Review Session: 12/13 Sun 4:30-6pm, Location TBA
- Pick up any paper from consultants (prelim, regrade results) during consulting hours next week. Everything will be shredded afterwards.
- Read announcements on course website!

Proportional representation in the spirit of "one person, one vote"

Article I Section 2 of the US Constitution:

"Representatives... shall be apportioned among the several States, which may be included within this Union, according to their respective numbers..."

How do you quantify fairness?
There are different models of fairness.
(Were some models advanced for political reasons?)

The ratio of population to delegation size as a measurement of fairness

Distribute 435 Congressional seats among the 50 states so that the ratio of population to delegation size is roughly the same from state to state.

Sounds specific, but even with this "definition" of fairness there're different models that can be used as demonstrated throughout history… and in this lecture.

Related questions

How “close” is a state to losing a Congressional district because of population changes?

If Puerto Rico and/or Washington DC become states and the number of Congressional seats remain the same, which states would lose a seat?

- Reasoning about change is very important!
- How does the “answer” change if the data change or if the assumptions that underlie the computation change?

  → **Sensitivity analysis**

The apportionment problem

Distribute 435 Congressional seats among the 50 states so that the ratio of population to delegation size is roughly the same from state to state.

Number of states:	`n`
State populations:	`p(1),…,p(n)`
Total population:	`P`
State delegation size:	`d(1),…,d(n)`
Number of seats:	`D`

Ideal: Equal Representation

$$\frac{P}{D} = \frac{p(1)}{d(1)} = \ldots = \frac{p(n)}{d(n)}$$

Number of states:	**n**
State populations:	**p(1),…,p(n)**
Total population:	**P**
State delegation size:	**d(1),…,d(n)**
Number of seats:	**D**

Realistic situation

$$\frac{P}{D} \approx \frac{p(1)}{d(1)} \approx \ldots \approx \frac{p(n)}{d(n)}$$

Number of states:	**n**
State populations:	**p(1),…,p(n)**
Total population:	**P**
State delegation size:	**d(1),…,d(n)**
Number of seats:	**D**

Definition

An Apportionment Method determines delegation sizes d(1),…,d(n) that are whole numbers so that representation is approximately equal:

$$\frac{p(1)}{d(1)} \approx \ldots \approx \frac{p(n)}{d(n)}$$

Jefferson Method 1790-1830

Decide on a "common ratio," the ideal number of constituents per district.

In 1790: `r = 33000`

Delegation size for the i-th state is

```
d(i) = floor( p(i)/r )
```

Jefferson Method 1790-1830

Population and the chosen common ratio determine the size of Congress:

Year	P	r	D
1790	3615920	33000	105
1800	4889823	33000	141
1810	6584255	35000	181
1820	8969878	40000	213
1830	11931000	47700	240

Webster Method 1840

```
d(i) = round( p(i) / 70680 )
```

instead of **`floor`**

Common Ratio

Size of Congress also determined by common ratio

Hamilton Method (1850-1900)

This method fixes the size of Congress.

Allocations are based on the “ideal ratio”:

Total Population / Total Number of Seats

The 1850 Case (31 States)

$$P = 21{,}840{,}083$$
$$D = 234$$
$$\text{Ideal ratio, } r = P/D \approx 93334$$

```
% Round 1 allocation…
for i=1:31
    d(i)= floor( p(i)/r )
end
```

At this point, all but 14 of the 234 seats have been given out

`p(i)/r` where `r` is the ideal ratio

AL	6.798	KY	9.622	NC	8.074
AR	2.047	LA	4.498	OH	21.218
CA	1.768	ME	6.248	PA	24.769
CT	3.973	MD	5.859	RI	1.581
DE	0.971	MA	10.655	SC	5.513
FL	0.768	MI	4.261	TN	9.717
GA	8.073	MS	5.171	TX	2.028
IL	9.123	MO	6.933	VT	3.366
IN	10.590	NH	3.407	VI	13.207
IA	2.059	NJ	5.244	WI	3.272
		NY	33.186		

floor(p(i)/r)

AL	6.798	KY	9.622	NC	8.074
AR	2.047	LA	4.498	OH	21.218
CA	1.768	ME	6.248	PA	24.769
CT	3.973	MD	5.859	RI	1.581
DE	0.971	MA	10.655	SC	5.513
FL	0.768	MI	4.261	TN	9.717
GA	8.073	MS	5.171	TX	2.028
IL	9.123	MO	6.933	VT	3.366
IN	10.590	NH	3.407	VI	13.207
IA	2.059	NJ	5.244	WI	3.272
		NY	33.186		

These 14 states most deserve an extra seat

AL	6.798	KY	9.622	NC	8.074
AR	2.047	LA	4.498	OH	21.218
CA	1.768	ME	6.248	PA	24.769
CT	3.973	MD	5.859	RI	1.581
DE	0.971	MA	10.655	SC	5.513
FL	0.768	MI	4.261	TN	9.717
GA	8.073	MS	5.171	TX	2.028
IL	9.123	MO	6.933	VT	3.366
IN	10.590	NH	3.407	VI	13.207
IA	2.059	NJ	5.244	WI	3.272
		NY	33.186		

Alabama paradox:
AL would lose 1 seat if Congress increases by 1 seat (1880 census)

Method of Equal Proportions

This method has been in use since 1940.

For the 2000 apportionment:

```
n = 50
D = 435
```

Determine the delegation sizes `d(1:50)`

Given the state populations `p(1:50)`

% Every state gets a district

% “Deal out” remaining districts
% by ranking the states and each time giving a
% district to the “most deserving state”

```
% Every state gets a district
d= ones(50,1);
% “Deal out” remaining districts
% by ranking the states and each time giving a
% district to the “most deserving state”
```

```
% Every state gets a district
d= ones(50,1);
% "Deal out" remaining districts
for k= 51:435
    % Let i be the index of the state that
    % most deserves an additional district
    d(i)= d(i) + 1;
end
```

How to quantify “most deserving”?

The Method of Small Divisors

At this point in the “card game” deal a district to the state having the largest quotient

`p(i)/d(i)`

Tends to favor small states.

How to quantify “most deserving”?

The Method of Large Divisors

At this point in the “card game” deal a district to the state having the largest quotient

`p(i)/(d(i) + 1)`

Tends to favor large states

How to quantify "most deserving"?

The Method of Major Fractions

At this point in the "card game" deal a district to the state having the largest value of

`( p(i)/d(i) + p(i)/(d(i)+1) )/2`

Compromise via the arithmetic mean

How to quantify “most deserving”?

The Method of Equal Proportions

At this point in the “card game” deal a district to the state having the largest value of

```
sqrt( p(i)/d(i) * p(i)/(d(i)+1) )
```

<u>Compromise via the geometric mean</u>

A Sensitivity Analysis

- The 435th district was awarded to North Carolina
- Was that a "close call"? Was there another state that "almost" won this last district? Quantify the close call.

A Sensitivity Analysis

- The 435th district was awarded to North Carolina.
- Was that a "close call"? Was there another state that "almost" won this last district? Quantify the close call.
  - Look at the "most deserving" ranks for the last district handed out. Which state was second? (Utah) Was this 2nd highest rank "close" to the max?
  - How many people will need to move from NC to UT in order for the last district to go to UT (instead of NC)?

Other questions

If Puerto Rico and/or Washington DC become states and the total number of representatives remains at 435, then which states would lose a congressional seat?

If the population of New York remains fixed and all other states grow by 5% during the2000-10 decade, then how many seats will NY lose?

Quantifying Importance

How do you rank web pages for importance given that you know the link structure of the Web, i.e., the in-links and out-links for each web page?

A related question:

How does a deleted or added link on a webpage affect its "rank"?

Background

Index all the pages on the Web from 1 to n. (n is around ten billion.)

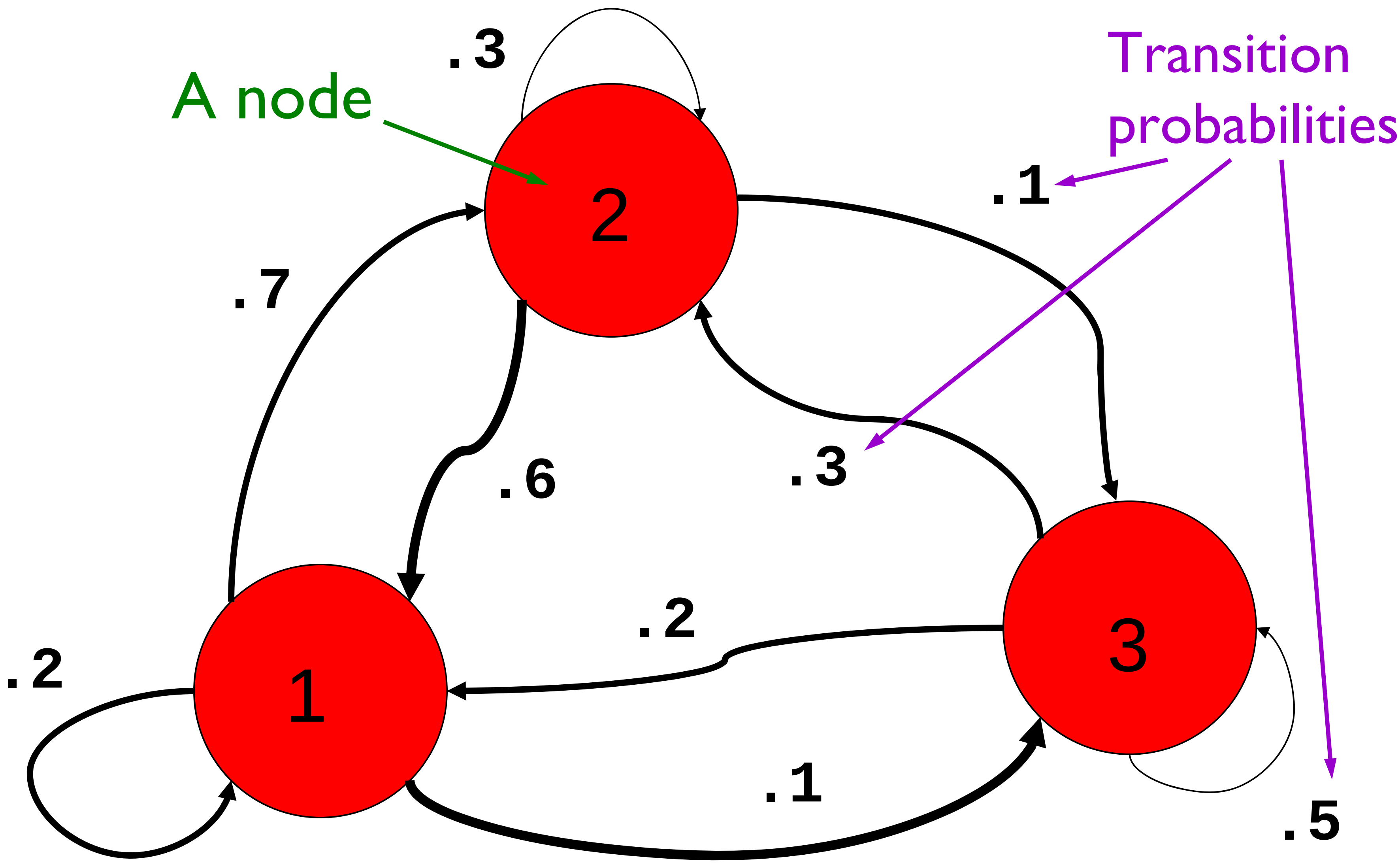
The PageRank algorithm orders these pages from "most important" to "least important."

It does this by analyzing links, not content.

Key ideas

- There is a random web surfer—a special random walk
- The surfer has some random "surfing" behavior—a transition probability matrix
- The transition probability matrix comes from the link structure of the web—a connectivity matrix
- Applying the transition probability matrix → Page Rank

A 3-node network with specified transition probabilities

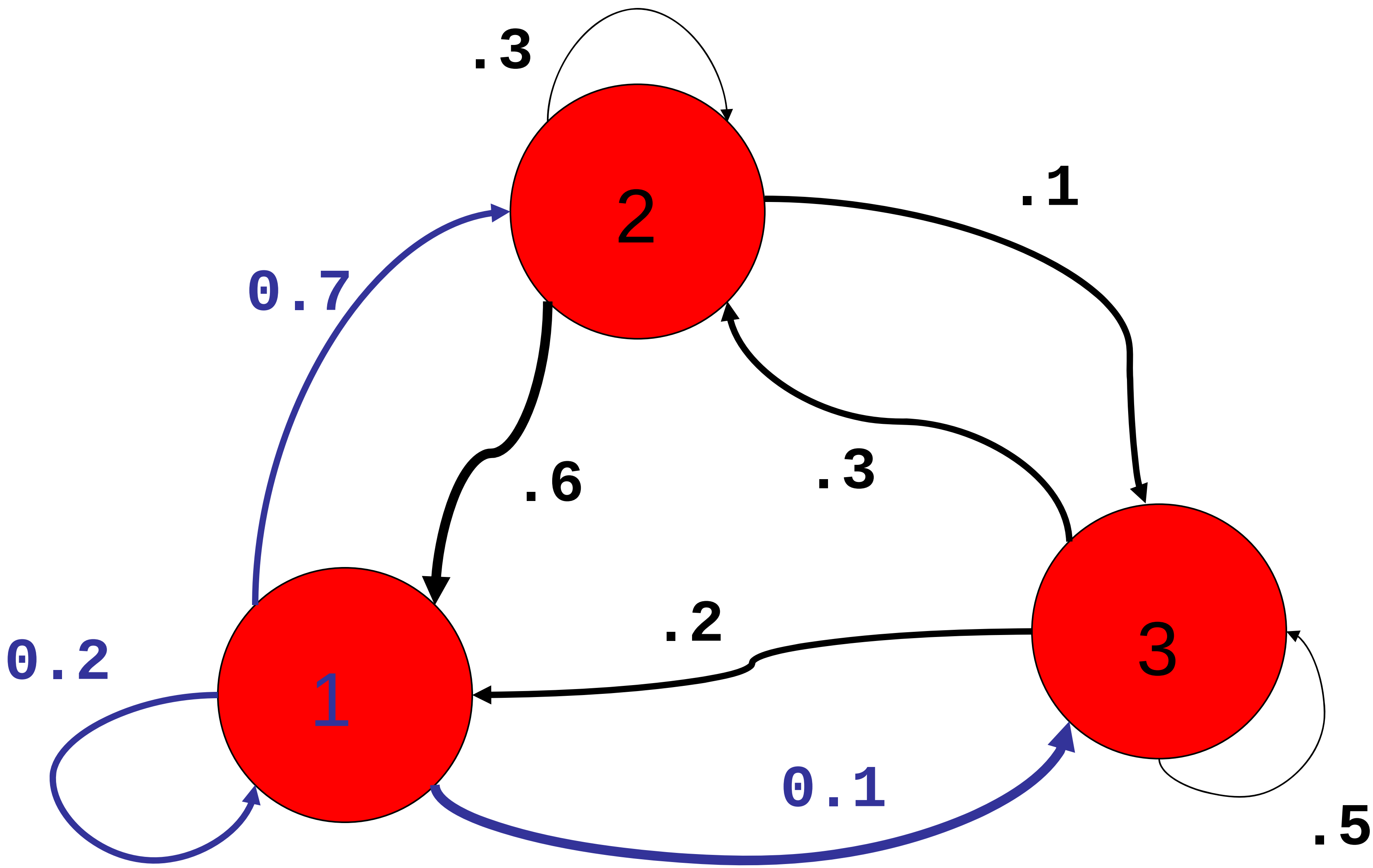


A special random walk

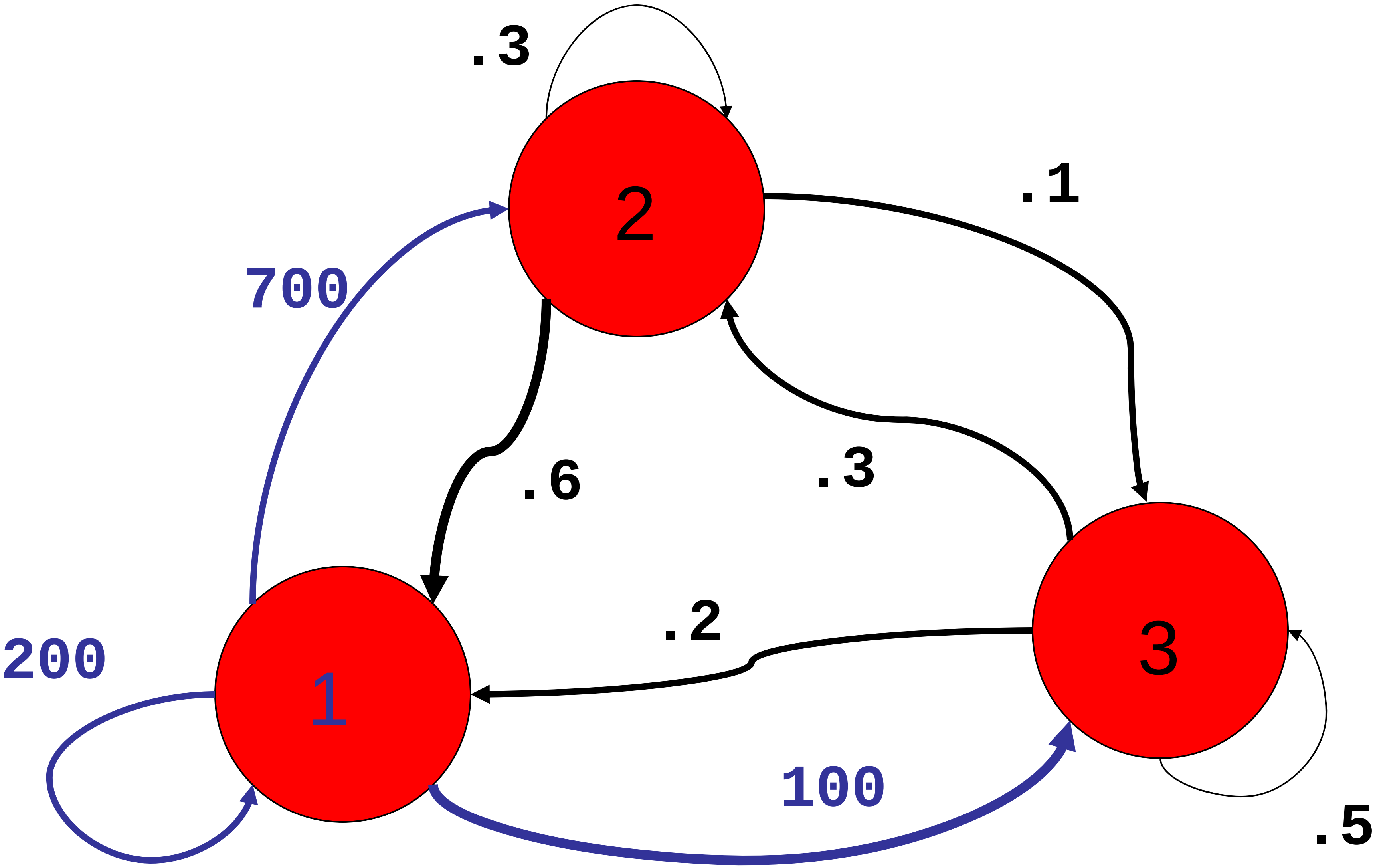
Suppose there are a 1000 people on each node.

At the sound of a whistle they hop to another node in accordance with the "outbound" probabilities.

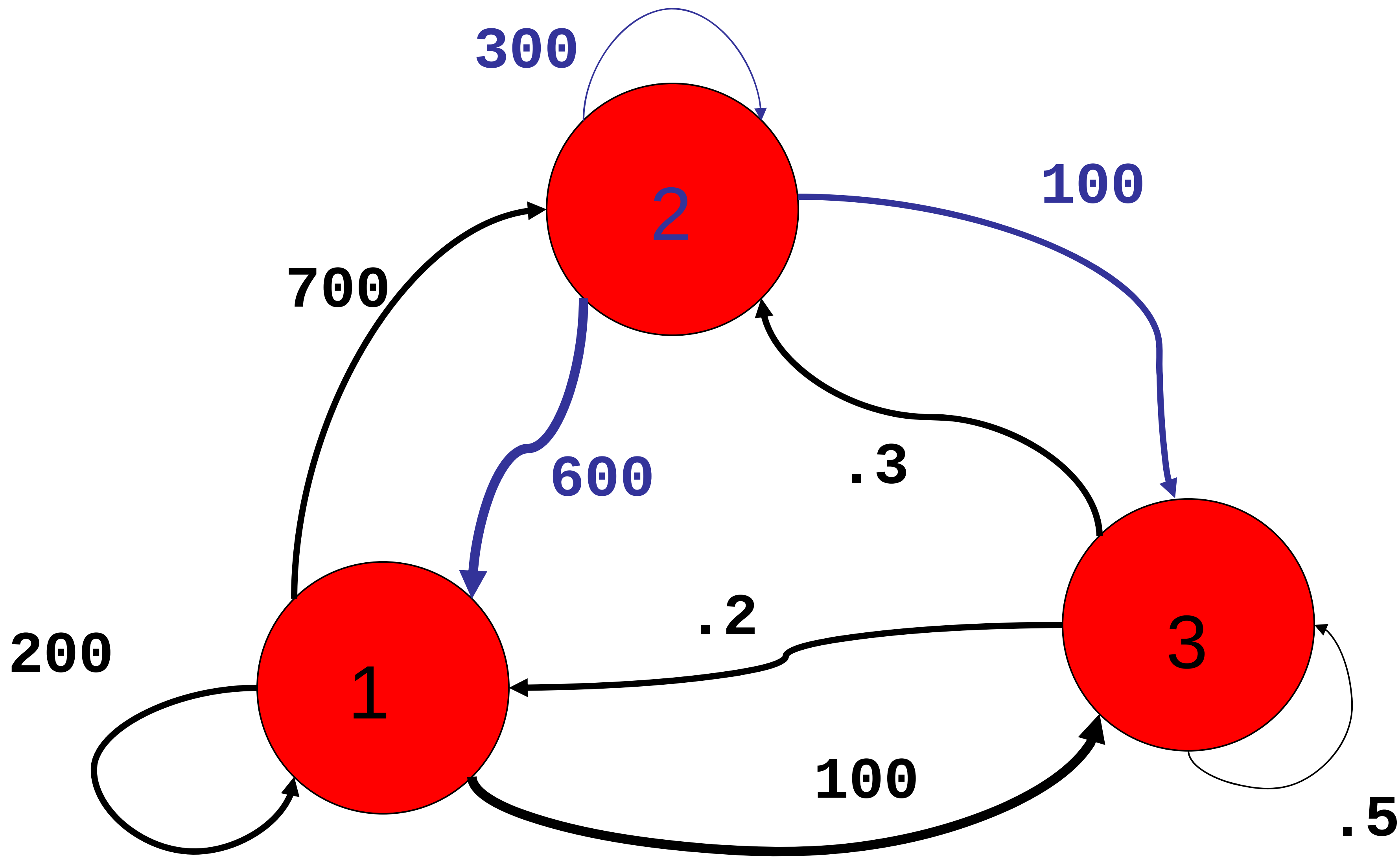
At Node 1



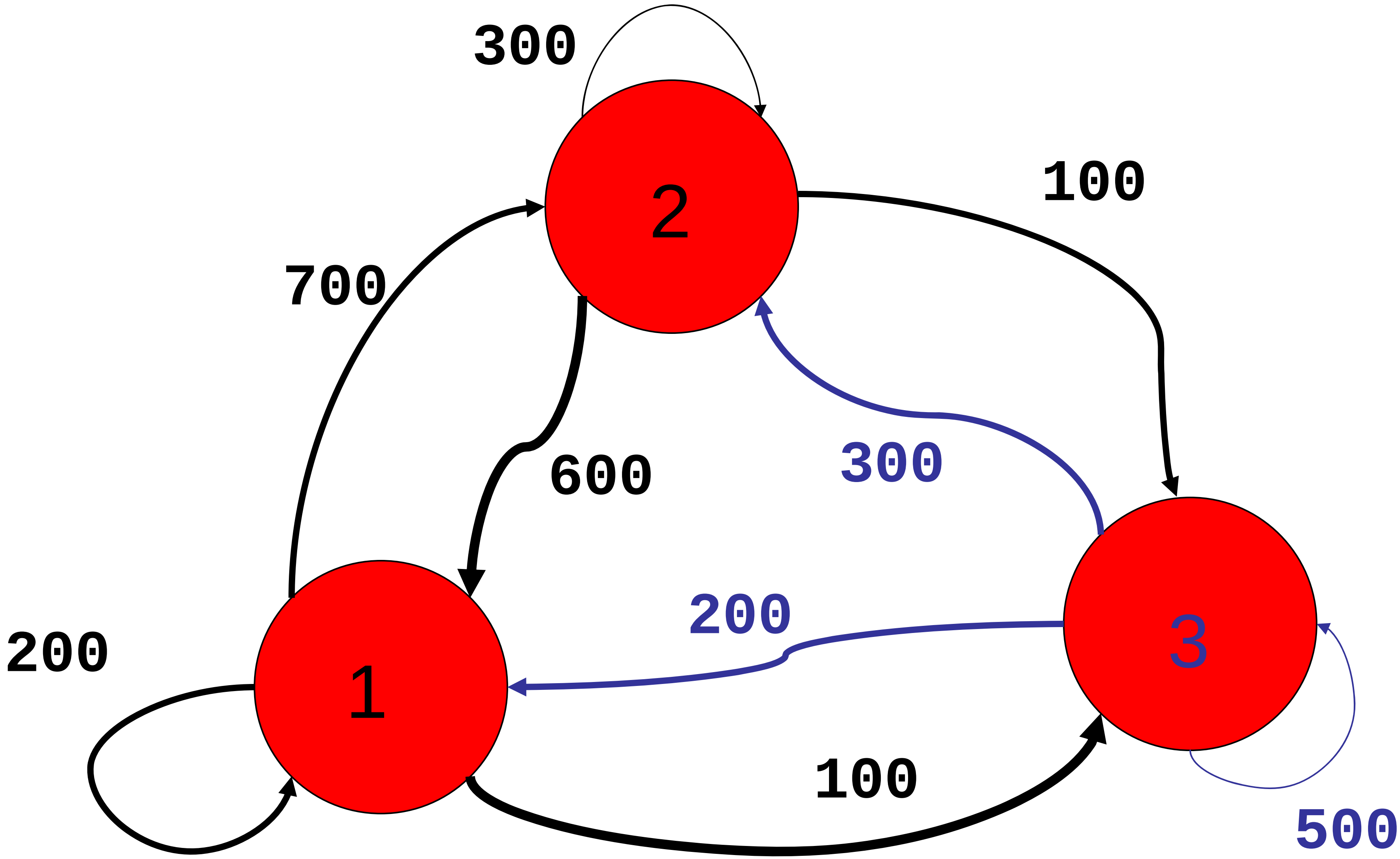
At Node 1



At Node 2



At Node 3



State Vector:
describes the state at each node at a specific time

T=0	T=1	T=2
1000	1000	1120
1000	1300	1300
1000	700	580

After 100 iterations

	Before	After
Node 1	1142.85	1142.85
Node 2	1357.14	1357.14
Node 3	500.00	500.00

Appears to reach a steady state

Call this the stationary vector

Transition Probability Matrix

$$
P \quad \begin{bmatrix} .2 & .6 & .2 \\ .7 & .3 & .3 \\ .1 & .1 & .5 \end{bmatrix}
$$

`P(i,j)` is the probability of hopping to node `i` from node `j`

Formula for the new state vector

P

.2	.6	.2
.7	.3	.3
.1	.1	.5

`P(i,j)` is probability of hopping to node `i` from node `j`

```
W(1) = P(1,1)*v(1) + P(1,2)*v(2) + P(1.3)*v(3)
W(2) = P(2,1)*v(1) + P(2,2)*v(2) + P(2,3)*v(3)
W(3) = P(3,1)*v(1) + P(3,2)*v(2) + P(3,3)*v(3)
```

`v` is the old state vector
`W` is the updated state vector

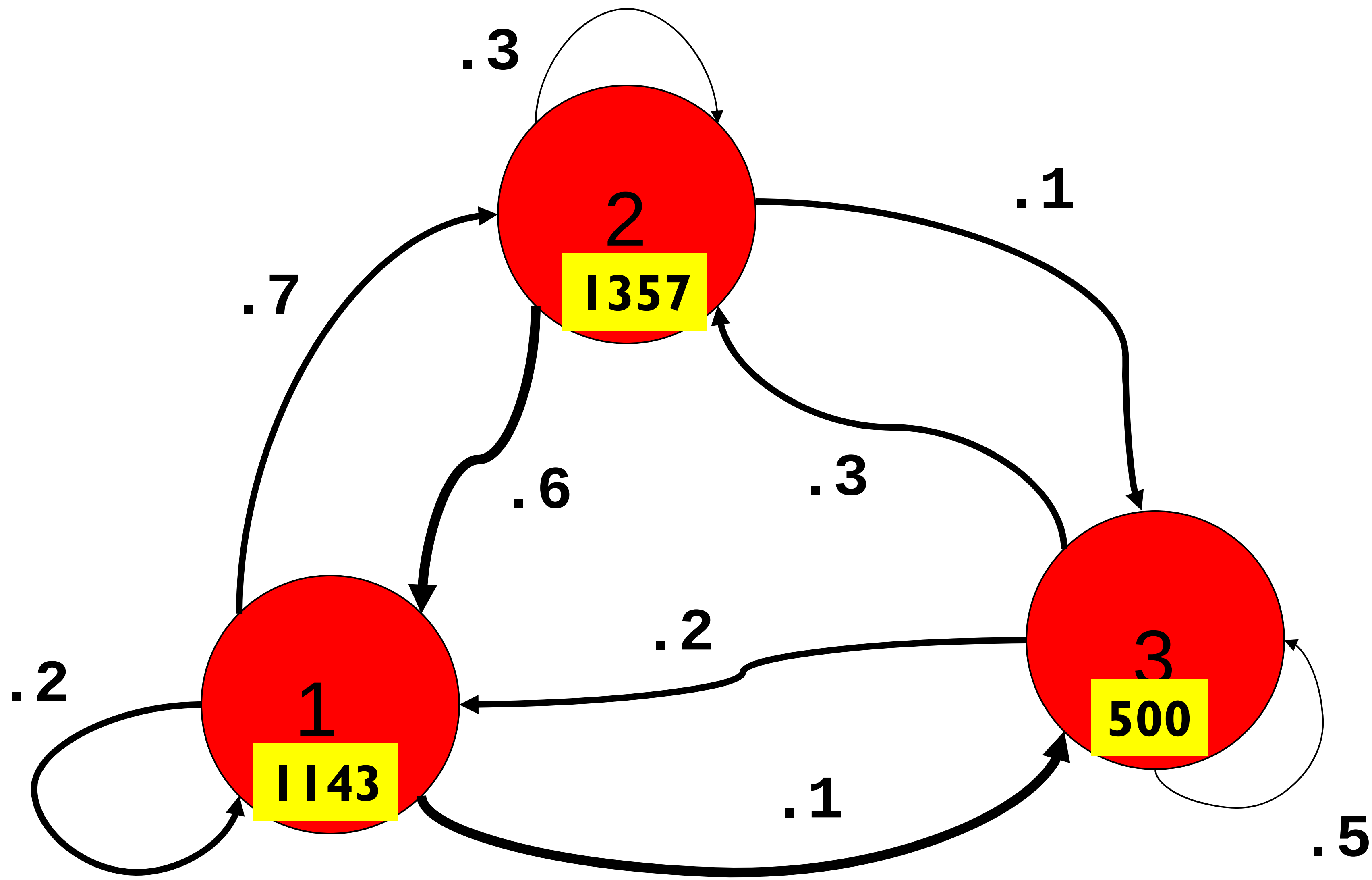
The general case

```
function w = Update(P,v)
% Update state vector v based on transition
% probability matrix P to give state vector w
n = length(v);
w = zeros(n,1);
for i=1:n
   for j=1:n
      w(i) = w(i) + P(i,j)*v(j);
   end
end
```

To obtain the stationary vector…

```
function [w,err]= StatVec(P,v,tol,kMax)
% Iterate to get stationary vector w
w = Update(P,v);
err = max(abs(w-v));
k = 1;
while k<kMax && err>tol
      v = w;
      w = Update(P,v);
      err = max(abs(w-v));
      k = k+1;
end
```

Stationary vector indicates importance: 2 1 3



A random walk on the Web

Repeat:

You are on a webpage.

There are m outlinks.

Choose one at random.

Click on the link.

What if there are no outlinks?
We'll deal with dead ends later.

A random walk on the Web

Repeat:

You are on a webpage.

There are m outlinks.

Choose one at random.

Click on the link.

- How to get the transition probability matrix to start the random walk?
- Use the link structure of the web!

Connectivity Matrix

G

0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1
0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0

G(i,j) is **1** if there is a link on page **j** to page **i**.

(I.e., you can get to **i** from **j**.)

Connectivity Matrix

G

0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1
0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0

Transition Probability Matrix derived from Connectivity Matrix

P

0	0	0	0	0	0	?	?
?	0	0	?	0	0	0	0
?	0	?	0	0	?	0	?
0	0	0	0	?	0	0	0
?	0	?	0	0	0	0	?
0	0	?	0	0	0	0	?
0	0	?	0	0	0	0	0
0	?	0	?	0	0	0	0

Connectivity Matrix

G

0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1
0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0

Transition Probability Matrix

derived from Connectivity Matrix

P

0	0	0	0	0	0	1	.25
.33	0	0	.50	0	0	0	0
.33	0	.25	0	0	1	0	.25
0	0	0	0	1	0	0	0
.33	0	.25	0	0	0	0	.25
0	0	.25	0	0	0	0	.25
0	0	.25	0	0	0	0	0
0	1	0	.50	0	0	0	0

Connectivity (G) → Transition Probability (P)

```
[n,n] = size(G);
P = zeros(n,n);
for j=1:n
   P(:,j) = G(:,j)/sum(G(:,j));
end
```

To obtain the stationary vector…

```
function [w,err]= StatVec(P,v,tol,kMax)
% Iterate to get stationary vector w
w = Update(P,v);
err = max(abs(w-v));
k = 1;
while k<kMax && err>tol
      v = w;
      w = Update(P,v);
      err = max(abs(w-v));
      k = k+1;
end
```

Stationary vector represents how “popular” the pages are → PageRank

statVec	sorted	idx	pR
0.5723	0.8911	6	4
0.8206	0.8206	2	2
0.7876	0.7876	3	3
0.2609	0.5723	1	6
0.2064	0.4100	8	8
0.8911	0.2609	4	1
0.2429	0.2429	7	7
0.4100	0.2064	5	5

```
[sorted, idx] = sort(-statVec);
for k= 1:length(statVec)
   j = idx(k);  % index of kth largest
   pR(j) = k;
end
```

statVec	sorted	idx	pR
0.5723	-0.8911	6	4
0.8206	-0.8206	2	2
0.7876	-0.7876	3	3
0.2609	-0.5723	1	6
0.2064	-0.4100	8	8
0.8911	-0.2609	4	1
0.2429	-0.2429	7	7
0.4100	-0.2064	5	5

The random walk idea gets the transitional probabilities from connectivity. So how to deal with dead ends?

Repeat:

You are on a webpage.

There are m outlinks.

Choose one at random.

Click on the link.

What if there are no outlinks?

The random walk idea gets transitional probabilities from connectivity. Can modify the random walk to deal with dead ends.

```
Repeat:
    You are on a webpage.
    If there are no outlinks
        Pick a random page and  go there.
    else
        Flip an unfair coin.
        if heads
            Click on a random outlink and go there.
        else
            Pick a random page and go there.
        end
    end
```

In practice, an unfair coin with prob .85 heads works well.

This results in a different transitional probability matrix.

What we learned…

- Develop/implement algorithms for problems
- Develop programming skills
  - Design, implement, document, test, and debug
- Programming "tool bag"
  - Control flow (if-else; loops)
  - Functions for reducing redundancy
  - Data structures
  - Graphics
  - File handling

What we learned… (cont'd)

- Applications and concepts
  - Image and sound
  - Sorting and searching—you should know the algorithms covered
  - Divide-and-conquer strategies
  - Approximation and error
  - Simulation
  - Computational effort and efficiency

Final Exam

- Mon 12/14, 7-9:30pm, Barton West
- Covers entire course, but emphasizes material after Prelim 2
- Closed-book exam, no calculators
- Bring student ID card

- Check for announcements on webpage:
  - Study break office/consulting hours
  - Review session time and location
  - Review questions
  - List of potentially useful functions