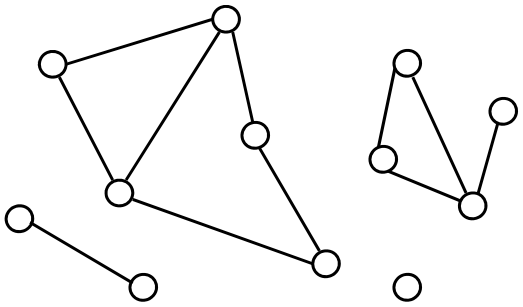
Finishing graph material
Dictionaries

- **Project 4**
  - Finding and evaluating clusters in an undirected connected graph
  - Provided file: earlier import_edge_list defined a directed graph; use an undirected graph
  - Gavin data set is large
- **Final Project: Continue 3 or 4**
  - Ising Spin Demon simulation
  - Using Gene Ontology Data to determine biological meaning of clusters

Clicker Question

What is the minimum number of edges that have to be added to make the graph connected?

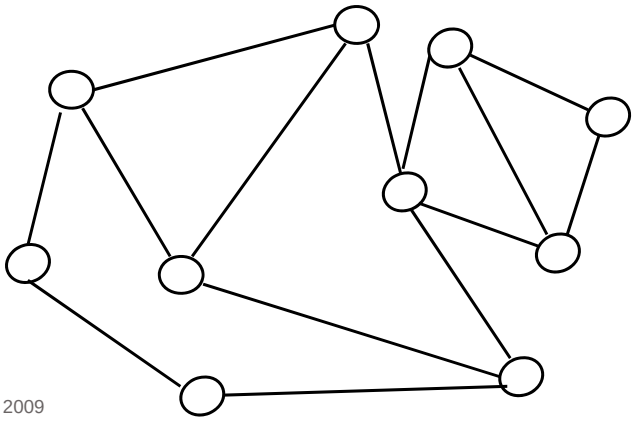
A. 2 edges
B. 3 edges
C. 4 edges
D. 5 edges



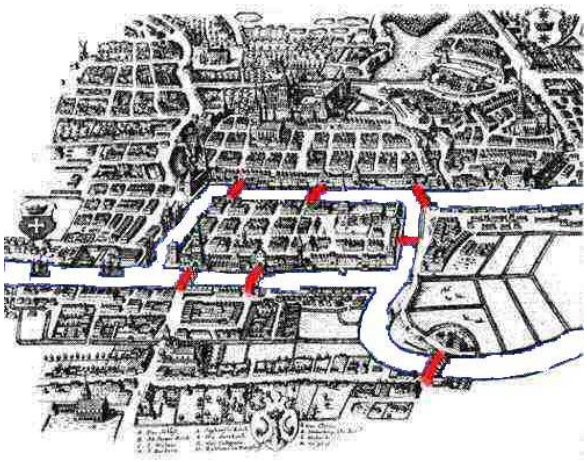
Clicker Question

What is the minimum number of nodes that have to be removed to create at least two connected components of size >1?

A. 1
B. 2
C. 3



Euler's Königsberg bridge problem

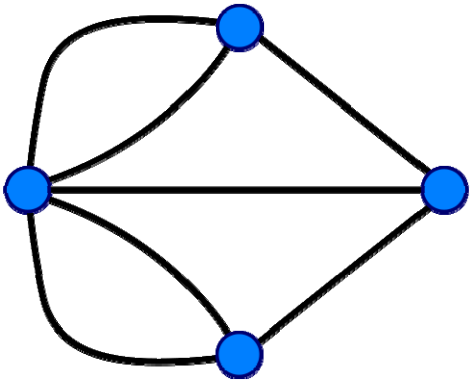


"Can I go for a walk and cross each bridge exactly once and return to where I started?"

Leonhard Euler (1707-1783)

Monday, April 13, 2009

Is it possible to construct a cycle which visits each edge exactly once?



Not for the graph representing the Königsberg bridges.

Euler figured this out in 1736.

Monday, April 13, 2009

Graph Theory & Graph Algorithms

- **Definition:** An Euler tour is a cycle which visits every edge of the graph exactly once.
- **Problem:** Given an undirected graph G, does G contain an Euler tour?
- **Theorem:** A connected undirected graph contains an Euler tour exist if and only if ever node has even degree.
- **Algorithm**
  - The test if an Euler tour exists, check all node degrees.
  - An Euler tour can be generated by a simple traversal of the graph, marking edges already visited (not a NetworkX function)

Monday, April 13, 2009

Representing Graphs

There are two main representations for graphs

- 2-dimensional matrix (adjacency matrix)
  - A[a][b] =1 if there exists an edge from node a to b
  - can also store the edge weight; node names mapped to integers
- Lists of edges ("adjacency lists")
  - i-th list contains the edges node i is incident to (generally represented as the list of adjacent nodes)

Monday, April 13, 2009

Revisit Web Crawler Simulation

- See notes and code from February16
  - Graph could have multiple edges between to nodes
- Graph was represented as a matrix
  - File matrix_surfer.py
  - Function get_next_page scanned a row of the matrix
- Now: creating a graph object of the classes in NetworkX
  - We don't know how graphs are represented
  - File graph_surfer.py
  - Function get_next_page uses the ability to access all edges

Monday, April 13, 2009

```
# Calculate the transition matrix given the 90/10 rule
RANDOM_PROB = .1
transition = zeros((N,N))
for i in range(N):
   for j in range(N):
      transition[i][j] =
            ((1-RANDOM_PROB) * link_counts[i][j] / out_degrees[i]
                  + RANDOM_PROB/N)
return transition, N
```

Main difference is in function get_next_page …

choosing the next node to visit is easier in the graph represenation

Monday, April 13, 2009

```
r = random.uniform(0,1)
    total = 0
    for k in range(N):
        total += trans_mat[current_site][k]
        if total >= r:  return k
```

```
r = random.uniform(0,1)
if r > RANDOM_PROB:                    # set to 0.1
    next_site = random.choice(G.successors(current_site))
else:
    next_site = random.choice(G.nodes())
return next_site
```


Dictionaries in Python

Zelle, 11.6

Learning Python, 8.3, 8.4

Data types you have used

- Numbers
- Strings
- Lists
- Arrays
- Sets (used in Lab 8)

Covered in Python texts/manuals, but not yet used

- **Dictionaries**

Example: Bird Watching

- Read bird watching observations from a file
  - one line of the file records the observation of one bird by giving the bird name
  - assume perfect spelling
- Create a structure that records how often a bird was seen
- Allow queries on the structure
  - Report how often a given bird was observed
  - Insert, delete a bird and its observation
  - Record another observation of a bird (increase count)
  - Print observations in various formats

In input file:

canada goose
speckle throated woodpecker
black nunbird
canada goose
california quail
black nunbird
banded stilt
canada goose

canada goose, 3
speckle throated woodpecker, 1
black nunbird, 2
california quail, 1
banded stilt, 1

This information can be represented in a list Bird_list where
Bird_list[i][0] stores a name
Bird_list[i][1] stores the count

Monday, April 13, 2009

Basic list solution

```
bird_count = [ ]
filename = "bird_observations.txt"
infile = open(filename, 'r')

for line in infile:
    name = string.lower(line.strip())
    found = False
    for entry in bird_count:
        if entry[0] == name:
                entry[1] += 1
                found = True
 if not found:
        bird_count.append([name, 1])
```

Monday, April 13, 2009

Using lists for the bird problem

- Lists can be used, but it can get clumsy and slow
- Lists represent a sequential collection storing ordered objects
  - L[i] refers to the i-th element in some order
- Is order important in the bird example?
  - Not really
  - Maybe we want to print the names of all birds in sorted order
  - Maybe print the names sorted by occurrence

What we really want

- The name of the bird should be the key
- The number of sightings is a value associated with the bird
- We must be able to change values
- A bird can appear at most once in the structure
- We want to dynamically change the size of the structure (insert and delete, no assumption on maximum size)

Such operations are supported by **dictionaries**

Dictionaries

- non-sequential data collection
- very flexible built-in data type
  - not provided as a built-in by all languages
- List = ordered collection of objects
  Dictionary = unordered collection of objects
- A dictionary
  - stores a collection of entries
  - retrieves an entry based on a key-value (not based on the position/index)
  - every key has associated values

Monday, April 13, 2009

```
>>> birds1 = {'canada goose' : 3, 'northern fulmar' : 10}
>>> birds1
{'canada goose' : 3, 'northern fulmar' : 1}
>>> birds1['northern fulmar' ]
10
>>> birds2 = {}                    # create an empty dictionary
 >>> birds2['banded stilt'] = 2
>>> birds2['black nunbird'] = 14
>>> birds2
{'banded stilt': 2, 'black nunbird': 14}
 >>> birds2.keys()
['banded stilt', 'black nunbird']
>>> birds2.items()
[('banded stilt', 2), ('black nunbird', 14)]
>>> birds2.values()
[2, 1]
```

Monday, April 13, 2009