LINK ANALYSIS

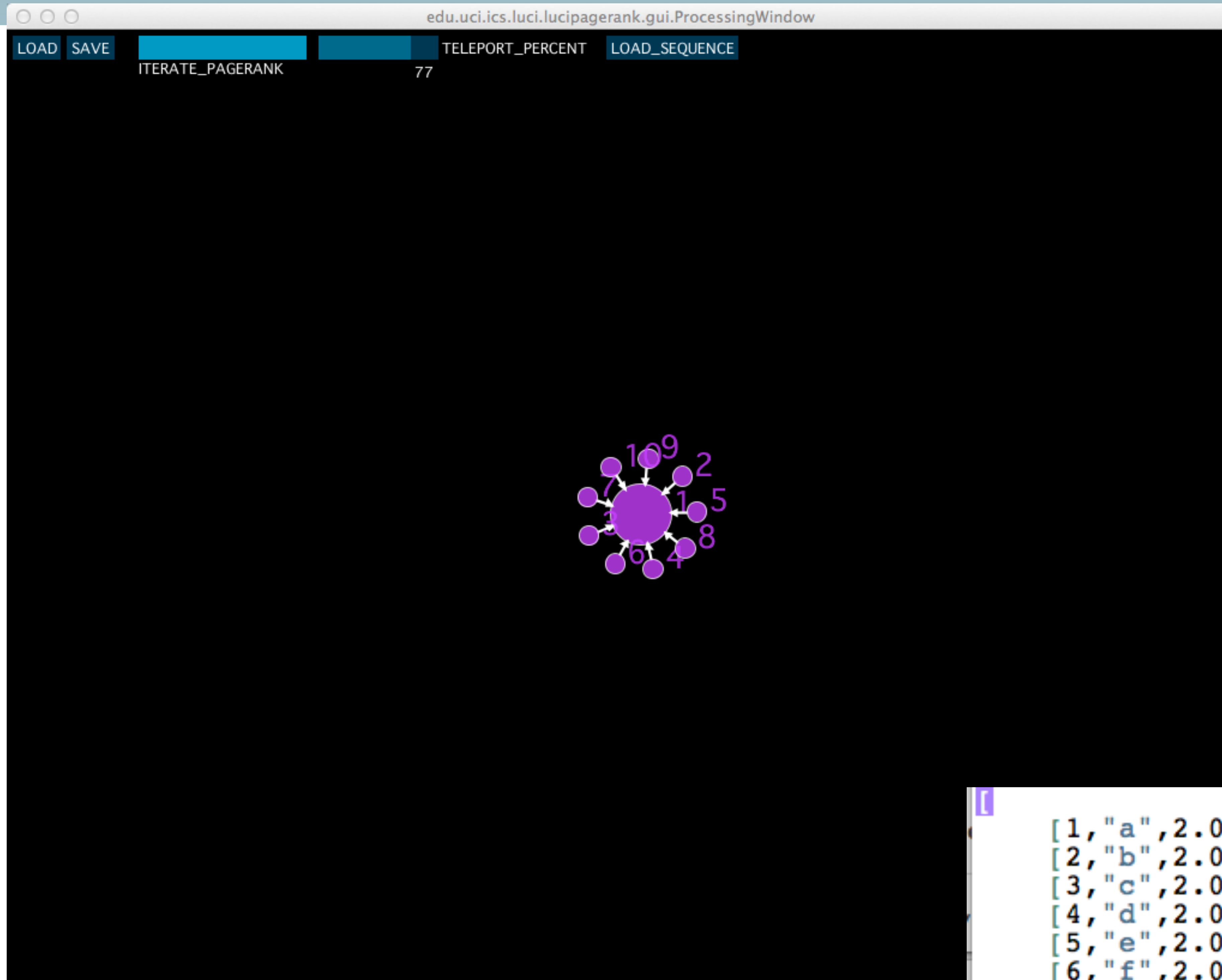
Introduction to Information Retrieval
CS 150
Donald J. Patterson

Content adapted from Essentials of Software Engineering 3rd edition by Tsui, Karam, Bernal Jones and Bartlett Learning

Draw a graph with 10 nodes

1) such that 1 node clearly has the highest PageRank

Link Analysis - Exercises

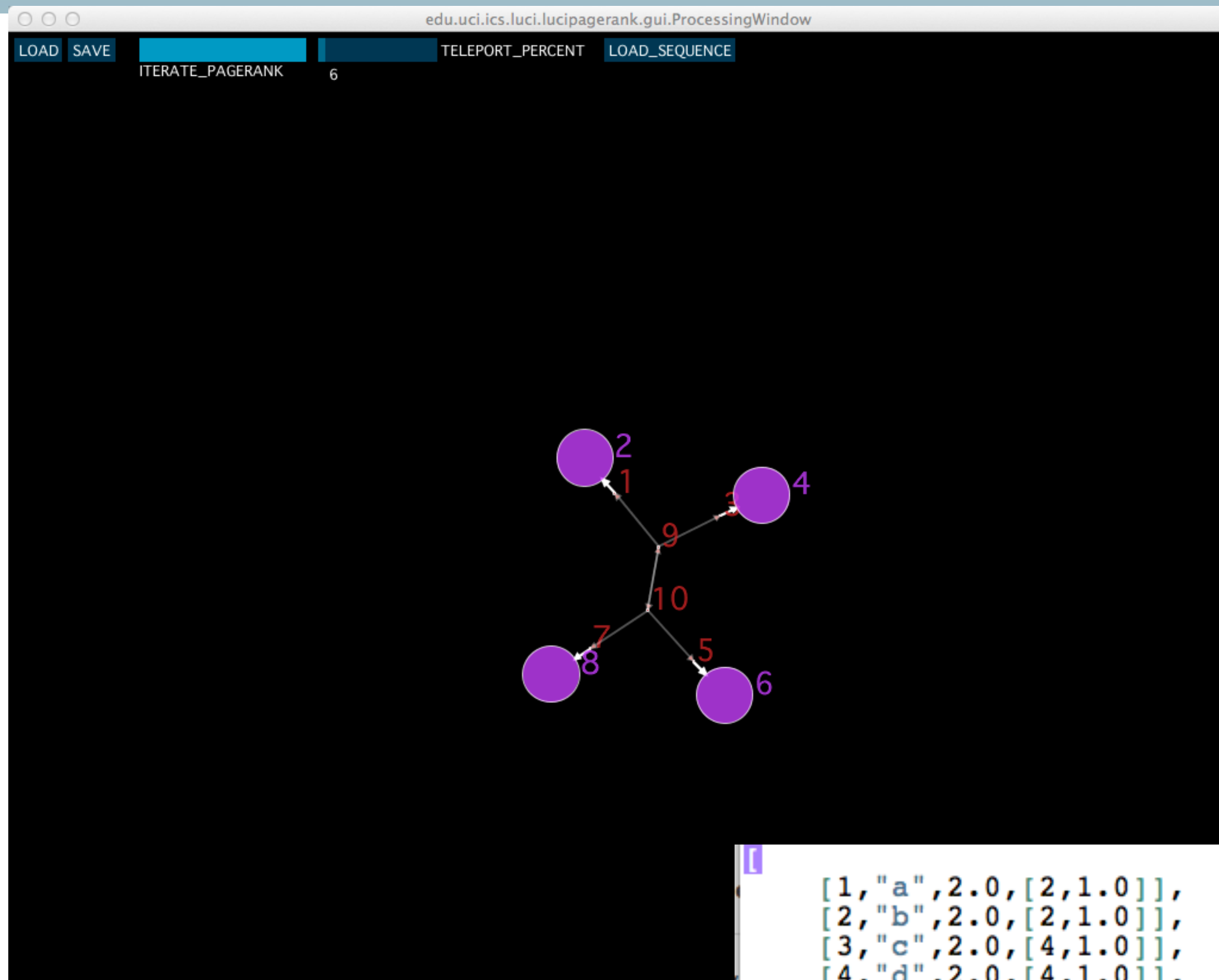



```
[
    [1,"a",2.0],
    [2,"b",2.0,[1,1.0]],
    [3,"c",2.0,[1,1.0]],
    [4,"d",2.0,[1,1.0]],
    [5,"e",2.0,[1,1.0]],
    [6,"f",2.0,[1,1.0]],
    [7,"g",2.0,[1,1.0]],
    [8,"h",2.0,[1,1.0]],
    [9,"i",2.0,[1,1.0]],
    [10,"j",2.0,[1,1.0]],
]
```

Draw a graph with 10 nodes

2) such that 4 nodes have very high and equal PageRank

Link Analysis - Exercises

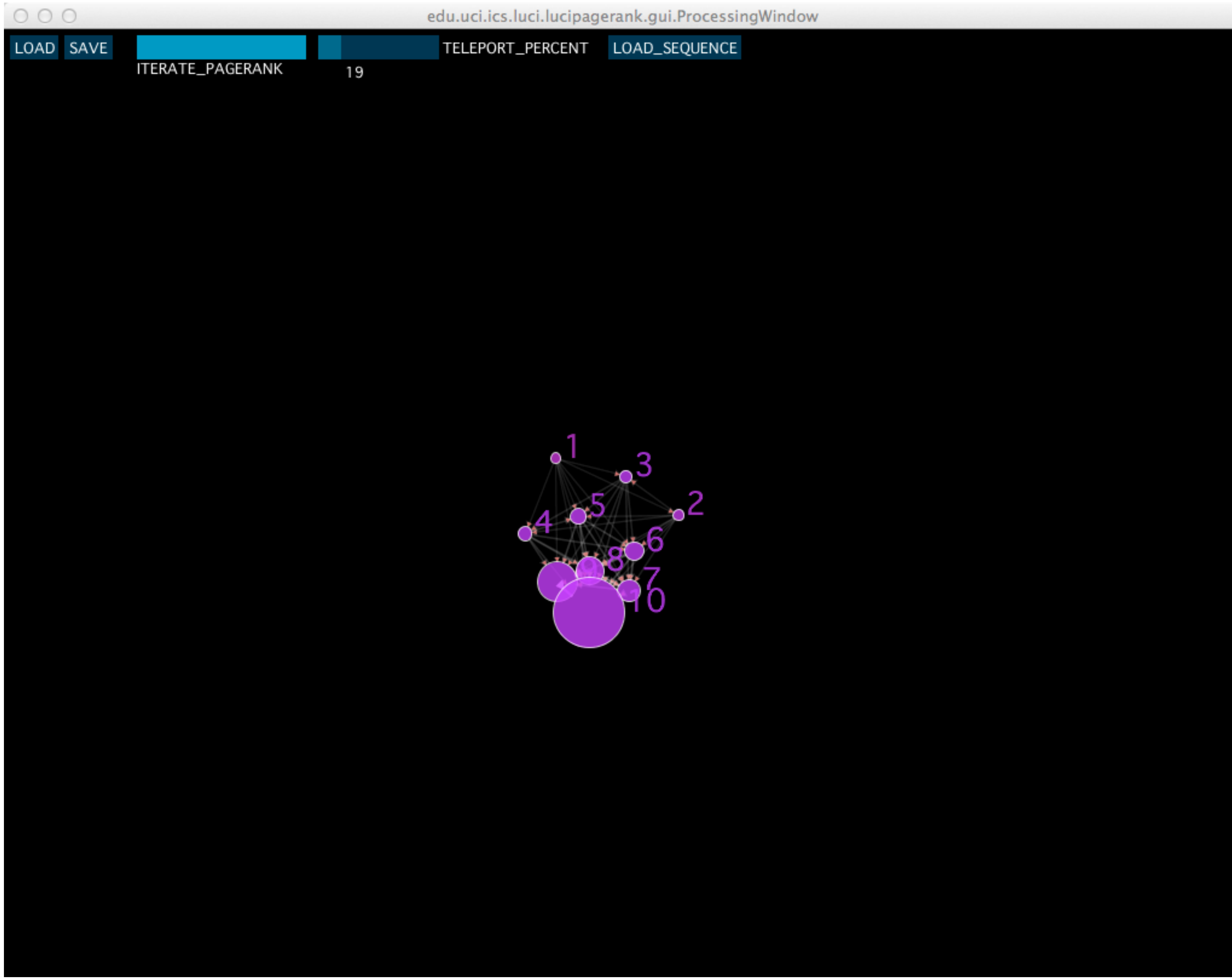



```
[
    [1,"a",2.0,[2,1.0]],
    [2,"b",2.0,[2,1.0]],
    [3,"c",2.0,[4,1.0]],
    [4,"d",2.0,[4,1.0]],
    [5,"e",2.0,[6,1.0]],
    [6,"f",2.0,[6,1.0]],
    [7,"g",2.0,[8,1.0]],
    [8,"h",2.0,[8,1.0]],
    [9,"i",2.0,[1,1.0],[3,1.0],[10,1.0]],
    [10,"j",2.0,[5,1.0],[7,1.0],[9,1.0]],
]
```

Draw a graph with 10 nodes

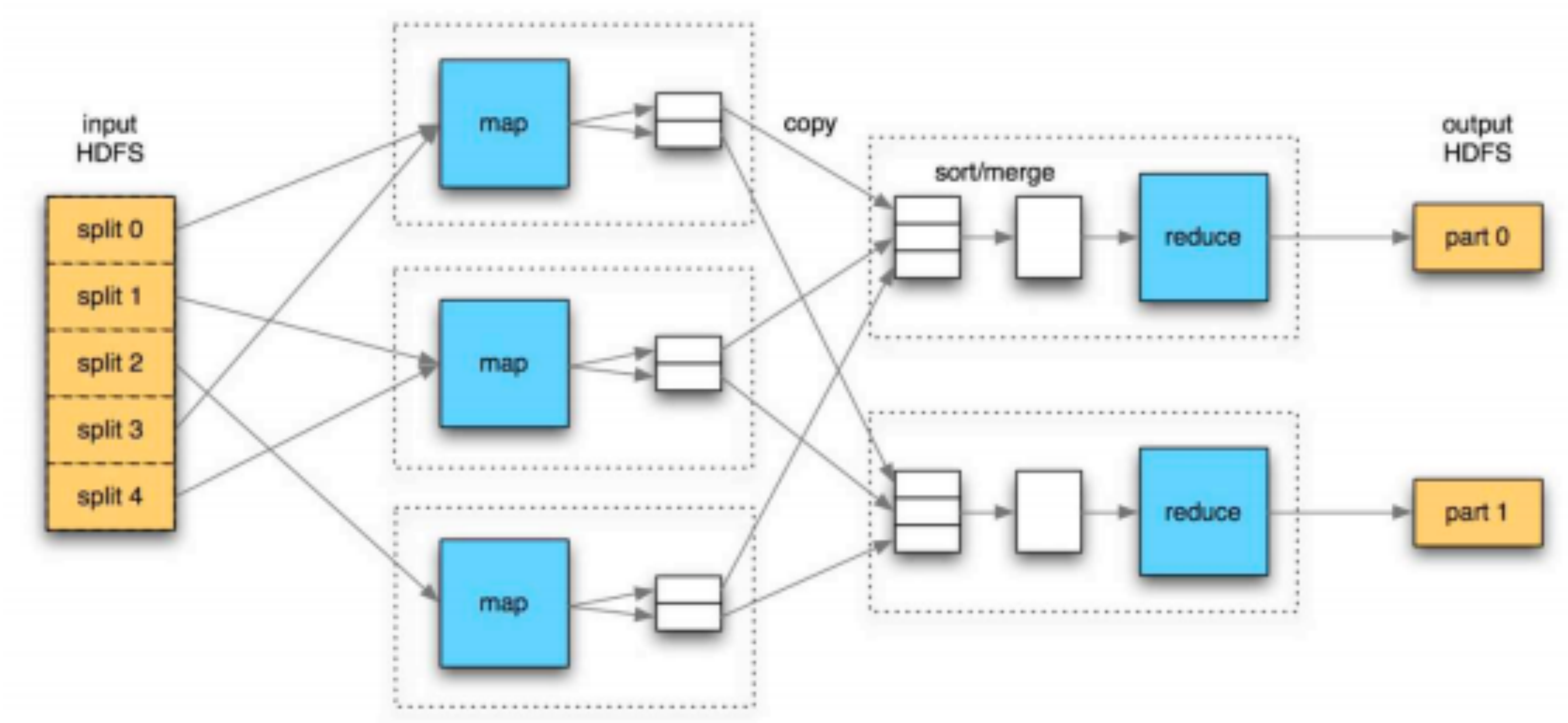
3) such that no node has the same PageRank

Link Analysis - Exercises




```
[
    [1,"a",2.0,[2,1.0],[3,1.0],[4,1.0],[5,1.0],[6,1.0],[7,1.0],[8,1.0],[9,1.0],[10,1.0]],
    [2,"b",2.0,[3,1.0],[4,1.0],[5,1.0],[6,1.0],[7,1.0],[8,1.0],[9,1.0],[10,1.0]],
    [3,"c",2.0,[4,1.0],[5,1.0],[6,1.0],[7,1.0],[8,1.0],[9,1.0],[10,1.0]],
    [4,"d",2.0,[5,1.0],[6,1.0],[7,1.0],[8,1.0],[9,1.0],[10,1.0]],
    [5,"e",2.0,[6,1.0],[7,1.0],[8,1.0],[9,1.0],[10,1.0]],
    [6,"f",2.0,[7,1.0],[8,1.0],[9,1.0],[10,1.0]],
    [7,"g",2.0,[8,1.0],[9,1.0],[10,1.0]],
    [8,"h",2.0,[9,1.0],[10,1.0]],
    [9,"i",2.0,[10,1.0]],
    [10,"j",2.0],
]
```

How could PageRank be calculated in Hadoop?



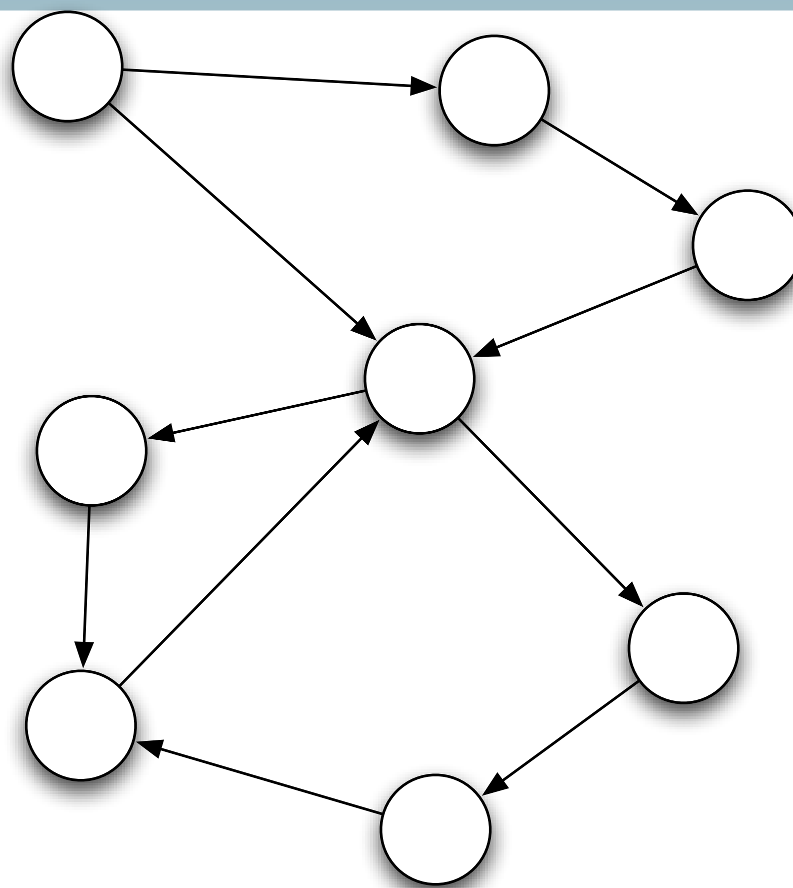
PageRank with MapReduce

- PageRank is iterative
- MapReduce is not
- This solution describes how to do one iteration of PageRank using MapReduce
- Multiple iterations would be required to converge

PageRank with MapReduce

- Quick review of PageRank
  - PageRank determines which pages are well-connected
  - A connection is a social signal that a web page is important
  - A connection is a vote for importance
  - Connections take time to form
    - Not so good for real-time data
  - Mathematically this is a Markov Chain

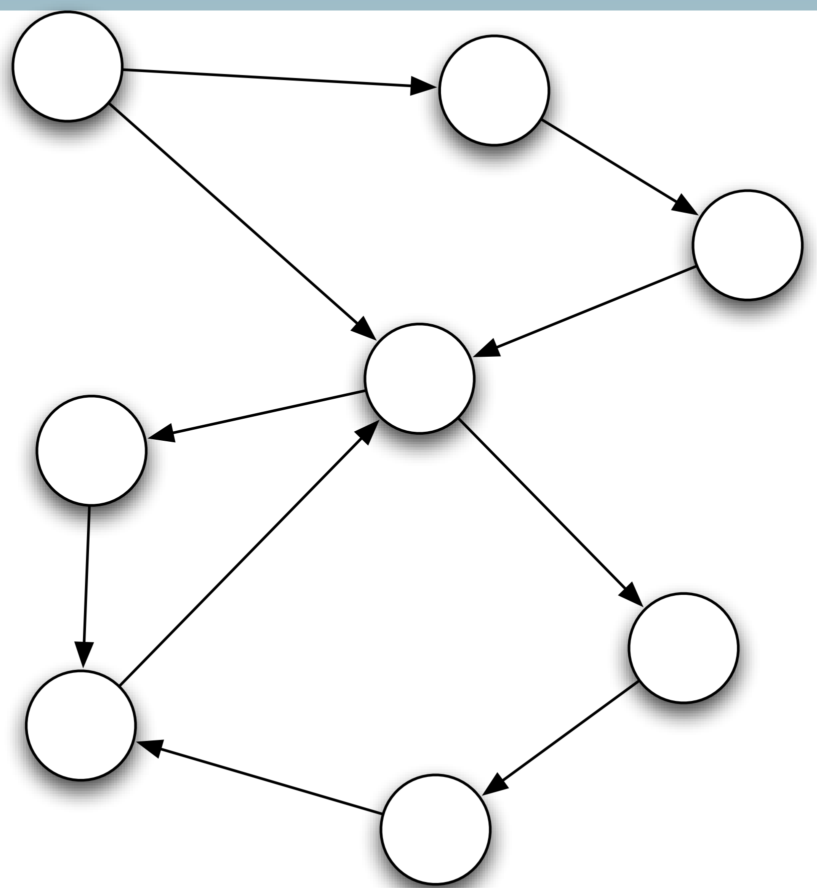
PageRank with MapReduce



- Quick review of PageRank
  - A Markov Chain
    - Has a starting probability
    - Has a set of states
    - Has transition probabilities
  - The web forms a graph which can be treated like a Markov Chain
  - If the Markov Chain is ergodic, then PageRank converges

PageRank with MapReduce

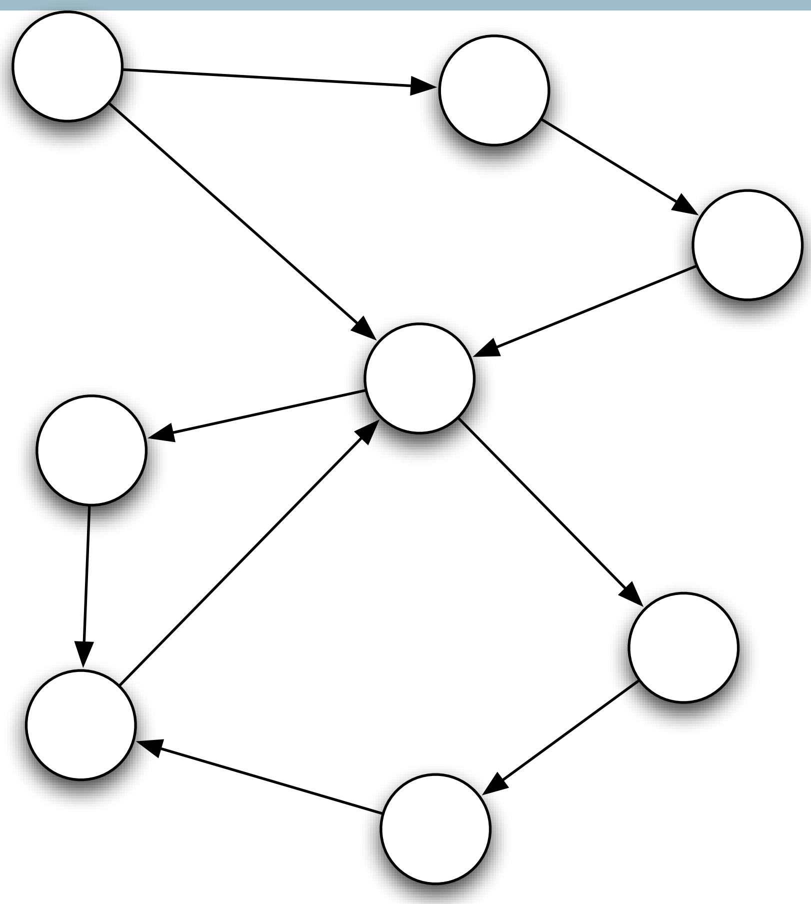
- Quick review of PageRank
  - A Markov Chain
    - Has a starting probability P_0
    - Has a set of states N
    - Has transition probabilities A_{ij}
  - The web forms a graph which can be treated like a Markov Chain
  - If the Markov Chain is ergodic, then PageRank converges



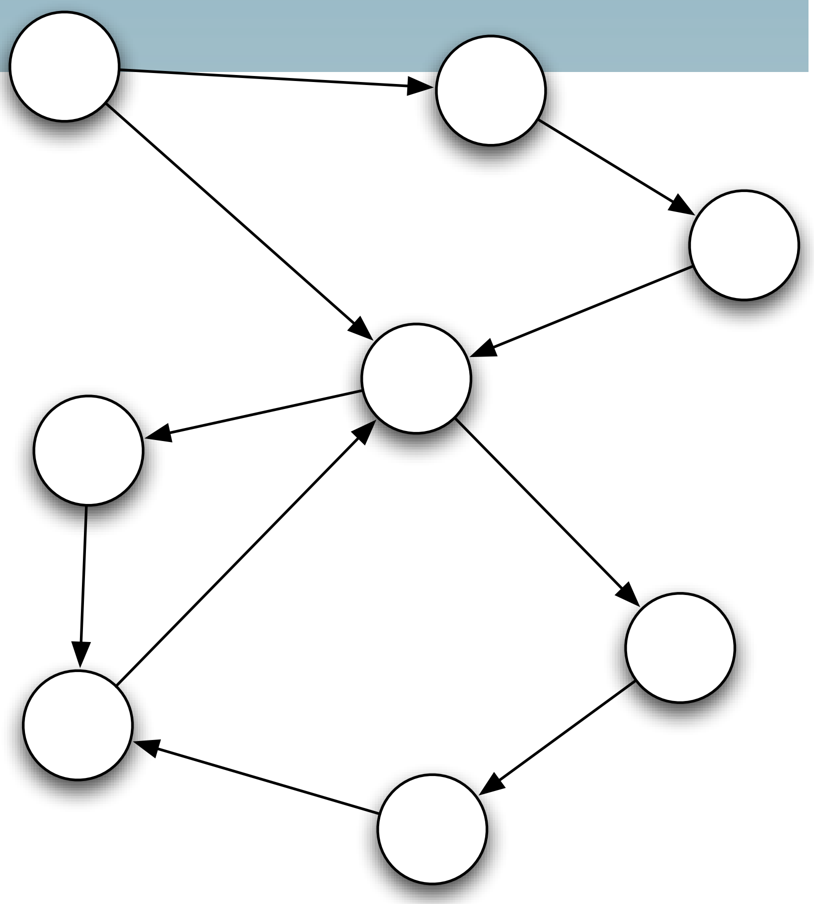
PageRank with MapReduce

$$P_1 = P_0 A$$

$$PageRank = \lim_{n \to \infty} (P_n)$$



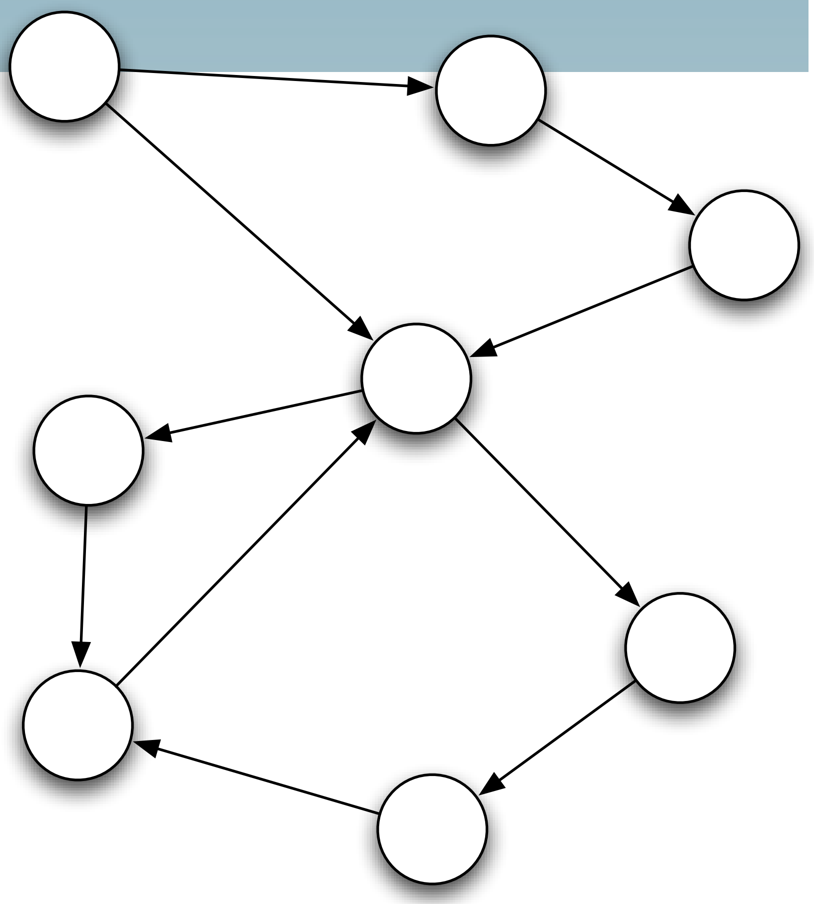
PageRank with MapReduce



- Assumptions
  - Initial probability is uniform
  - A transition is made up of
    - outlinks O
    - deadend teleports D
    - random teleports T
    - a mixing constant $0 <= \alpha <= 1$

$$A_{ij} = \alpha O + \alpha D + (1 - \alpha) T$$

PageRank with MapReduce



- Assumptions
  - Initial probability is uniform
  - A transition is made up of
    - outlinks O
    - deadend teleports D
    - random teleports T
    - a mixing constant $0 <= \alpha <= 1$

$$A_{ij} = \alpha O + \alpha D + (1 - \alpha) T$$

PageRank with MapReduce

- Map
  - Input is
    - key: page id, i
    - value: [p_i, set of outlinked pages O_i]
  - One output for every page $j \in (1..n)$
    - key: page id, j
    - value:
      - if $(O_i == \{\})$ $(\alpha f_D(i,j) + (1-\alpha) f_T(i,j)) p_i$
      - if $(j \in O_i)$ $(\alpha f_O(i,j) + (1-\alpha) f_T(i,j)) p_i$
      - if $(j \notin O_i)$ $(\alpha(0) + (1-\alpha) f_T(i,j)) p_i$

$$p_i(\alpha \frac{1}{|O_i|} + (1-\alpha)\frac{1}{n})$$

PageRank with MapReduce

- Outlink probability
  - uniform

$$f_O(i,j) = \frac{1}{|O_i|}$$

- When you hit a deadend
  - jump to a random page uniformly

$$f_D(i,j) = \frac{1}{n}$$

- When you teleport
  - teleport to a random page uniformly

$$f_T(i,j) = \frac{1}{n}$$

- More sophisticated extensions are imaginable

PageRank with MapReduce

- Reduce collects the probabilities and adds them
  - Input is
    - key: page id, i
    - value: probability of $j \rightarrow i$
  - Output is
    - key: page id, i
    - value: sum of all input probabilities

$$p_i = \sum_j p_j A_{ji}$$

PageRank with MapReduce

- Summary
  - Each step of PageRank computes one iteration of

$$P_{n+1} = P_n A$$

  - Each Map job handles the probability mass of one page being split across many pages
  - Each Reduce job collects the probabilities of one page coming from many pages

Link Analysis - Exercises

input: node_a:[P(node_a), [node_b,node_c]]

map out: [node_b, P(node_a)/2]
[node_c, P(node_a)/2]
[node_a,[node_b,node_c]]

reduce in:
node_x: [P(in1),...,P(in3)....[node_y,node_z]]

reduce out:
node_x: [sum(P(in1)...P(in3)),[node_y,node_z]]