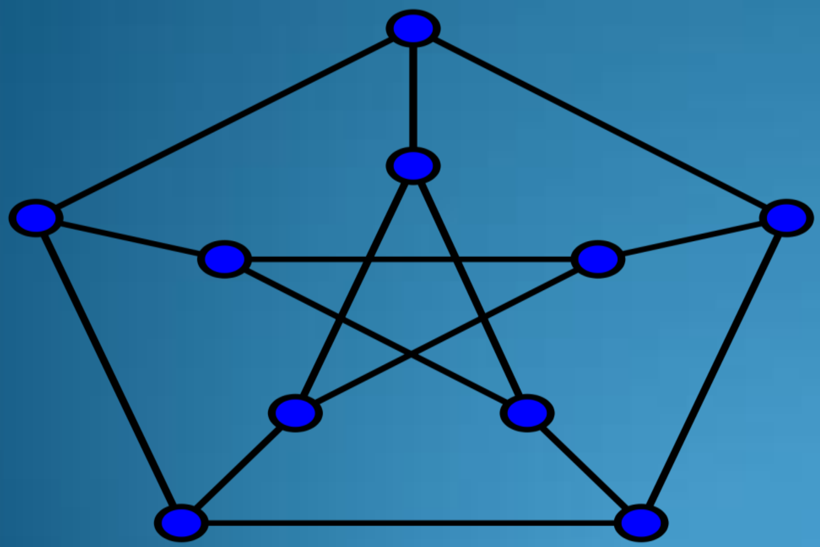
Permutation Graphs and Applications



ASAAD Y. GHAWI
Kent State University

In this Presentation:

- Idea of Permutation Graphs
- Applications
- Classical problem (Coloring)

Introduction

- **Let's consider the sequence of natural numbers:**

  N={1,2,3,4,5,6}

- **Then a permutation is a sequence as follows:**

  P1= {1,2,3,4,5,6}

  P2={6,2,4,3,1,5}

  P3={5,4,3,6,1,2}

  …………….

  …………….

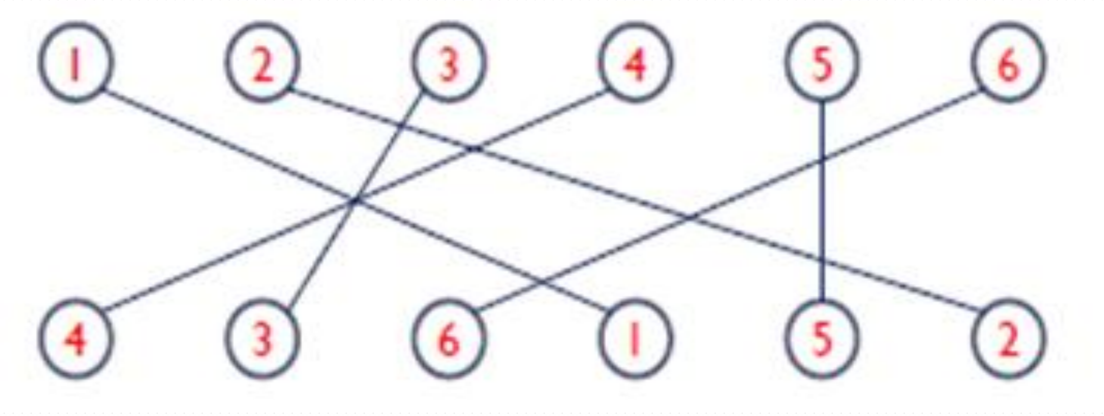
- **Notice there are N of permutations π**

Introduction

- **Given a permutation π, we can construct a graph G(π):**
- **Example:**

  N = [1, 2, 3, 4, 5, 6]

  π = [4, 3, 6, 1, 5, 2]

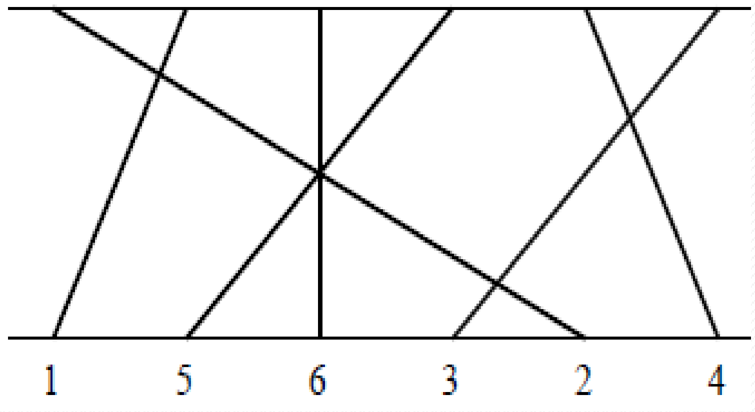


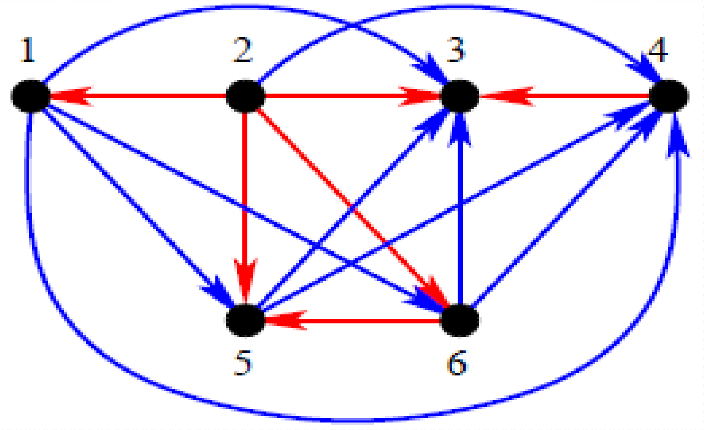
Introduction

Properties:

- **Reversing the order of π gives us the complement**
  - The complement is also a permutation graph

- **Transitively Orientable**
  - G(π) and G'(π) are transitively orientable





Introduction

Properties:

- **Corollary 7.4(Key point):**

  The chromatic number of G(π)=minimum number of queues required to sort π.

- **Also:**

  Theorem : A graph is a permutation graph IFF G and G' are comparability graphs.

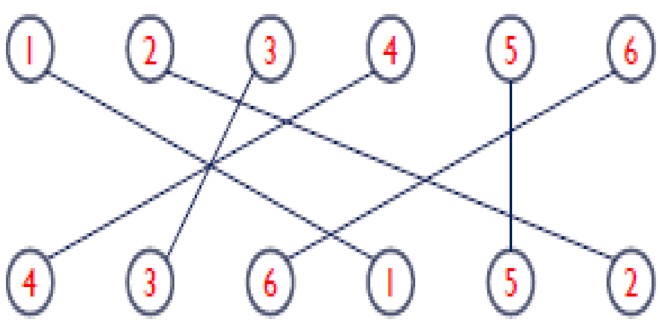
Application

- **Matching Diagram**

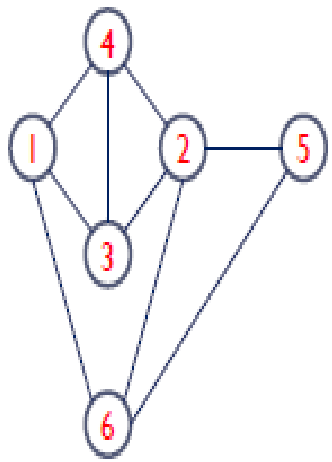
- is drawn by adjoining pairs of matching elements from the sets N and π.

- Intersections depict adjacency.

- characterizes Permutation Graphs.



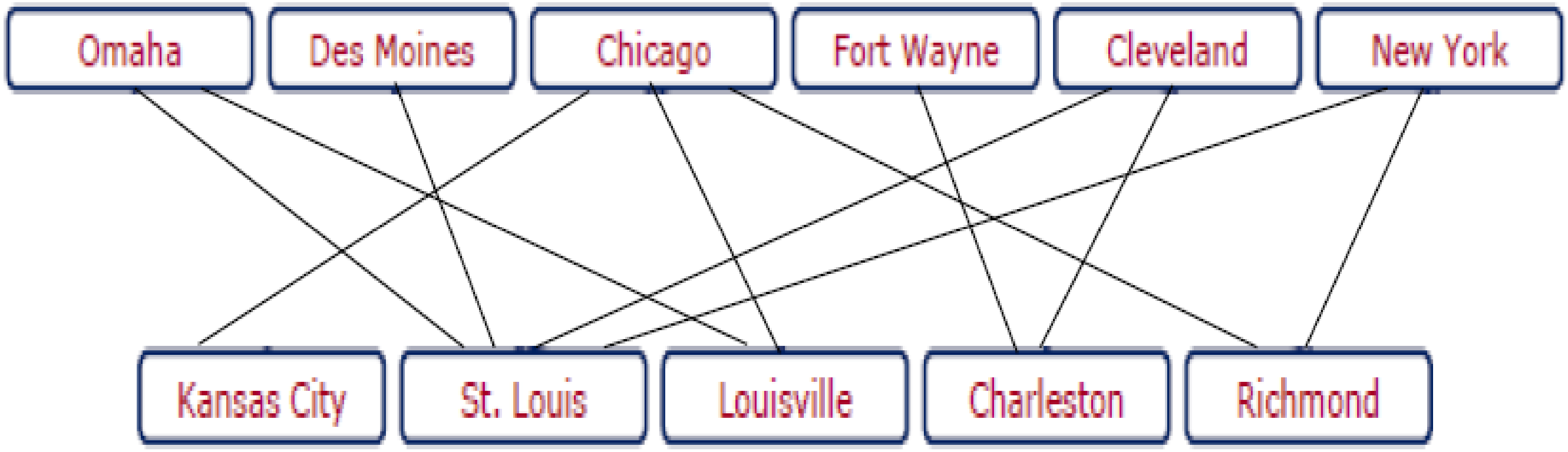
Matching Diagram of π = [4, 3, 6, 1, 5, 2]



Permutation Graph for π = [4, 3, 6, 1, 5, 2]

Application

- **Airline Routes** (our real-life problem)
  - Suppose two collections of cities -airports
- Assign flight altitudes to connecting cities
  - Prevent intersecting flights colliding mid-air
- Example:

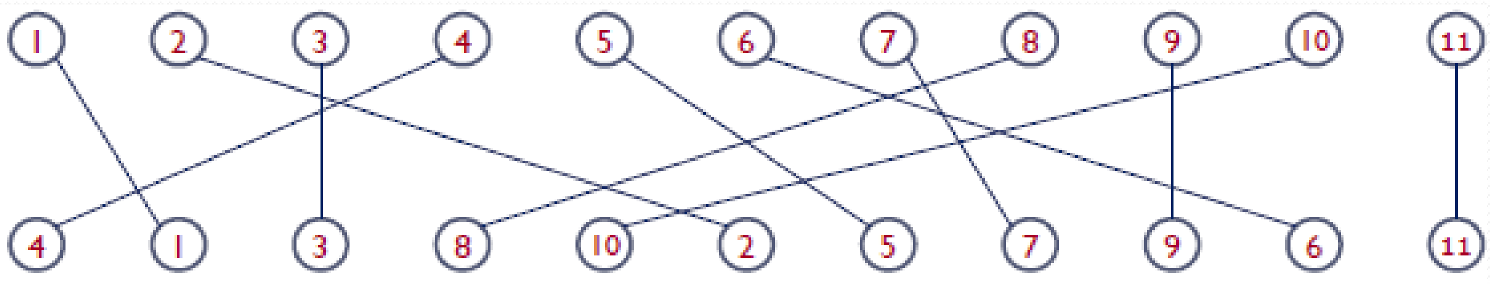


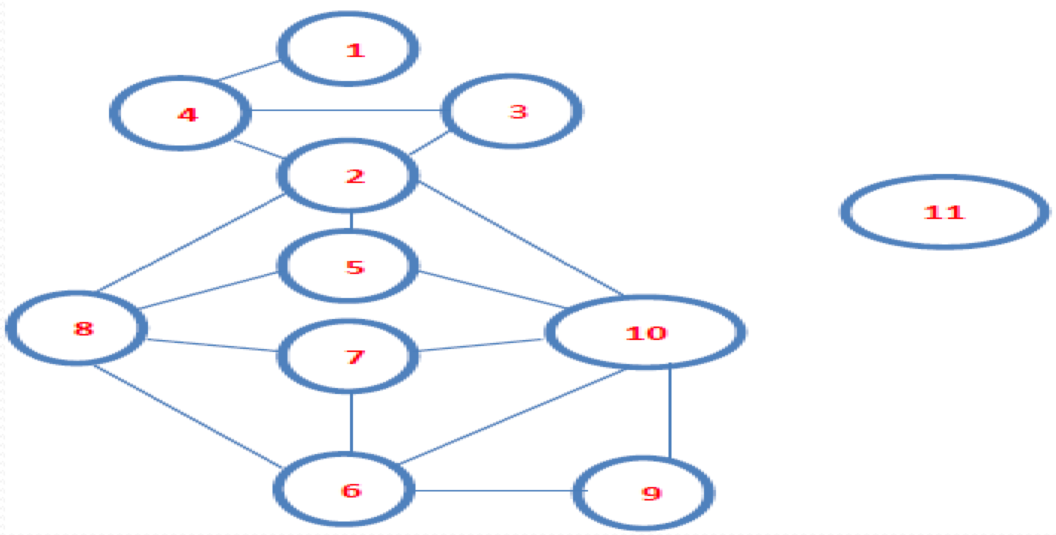
Application

- **Draw a matching diagram**

  - Each edge of the flight diagram is an "element"

  - Gives us a permutation and a graph

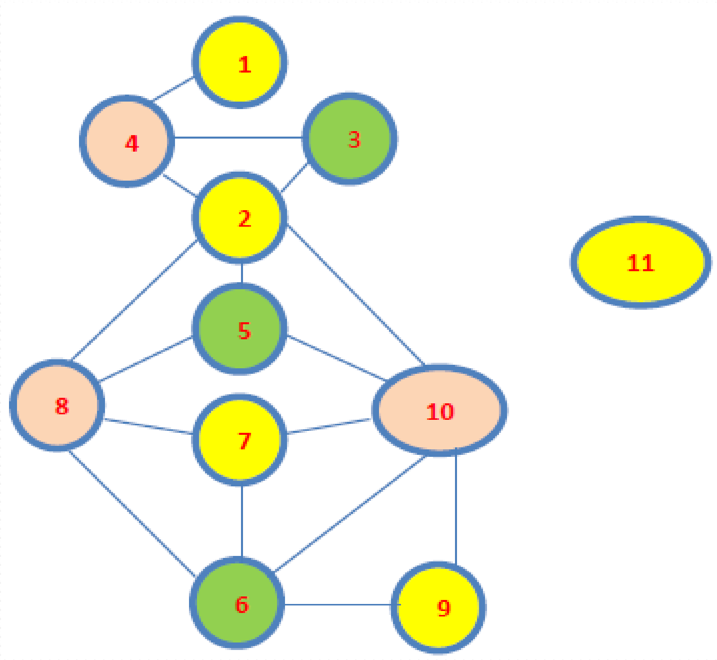




Classical Problem

- **The Coloring problem**

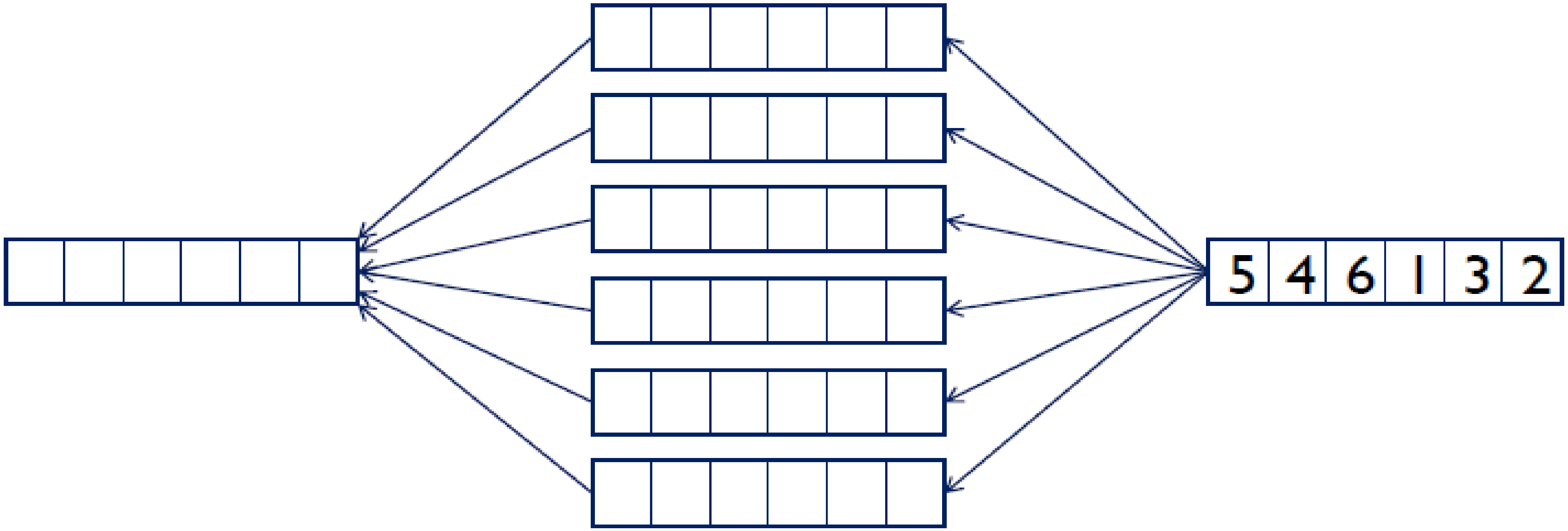
  - Distinct colors need distinct altitudes



- **Algorithm:**

Sorting Permutation

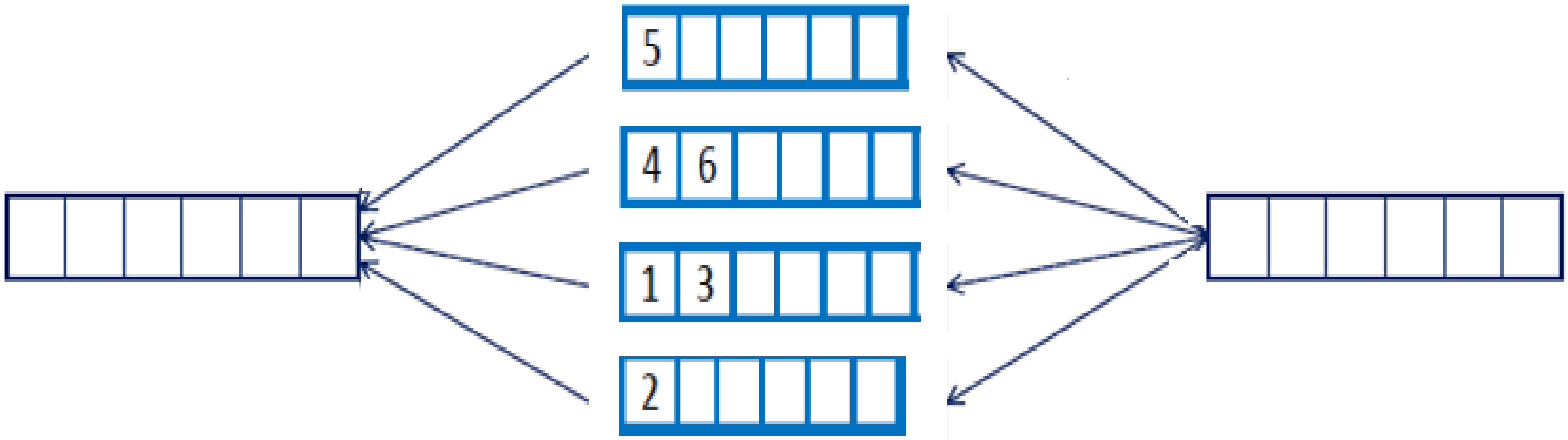
- **We use n queues in parallel to demonstrate:**



- **Each element goes into one of the n-queues**
  - But cannot go "behind" a larger element

Sorting Permutation

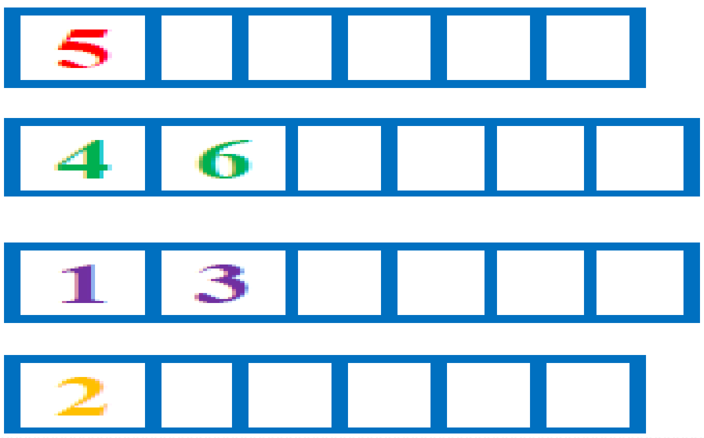
- **The "unpacking/withdrawing" stage is done**
  - By pulling out elements in proper order



Conclusion: We only allow non-inverted pairs in same queue

Sorting Permutation

- **Assigning unique color to each queue**



Corollary 7.4:

chromatic number = minimum number of queues

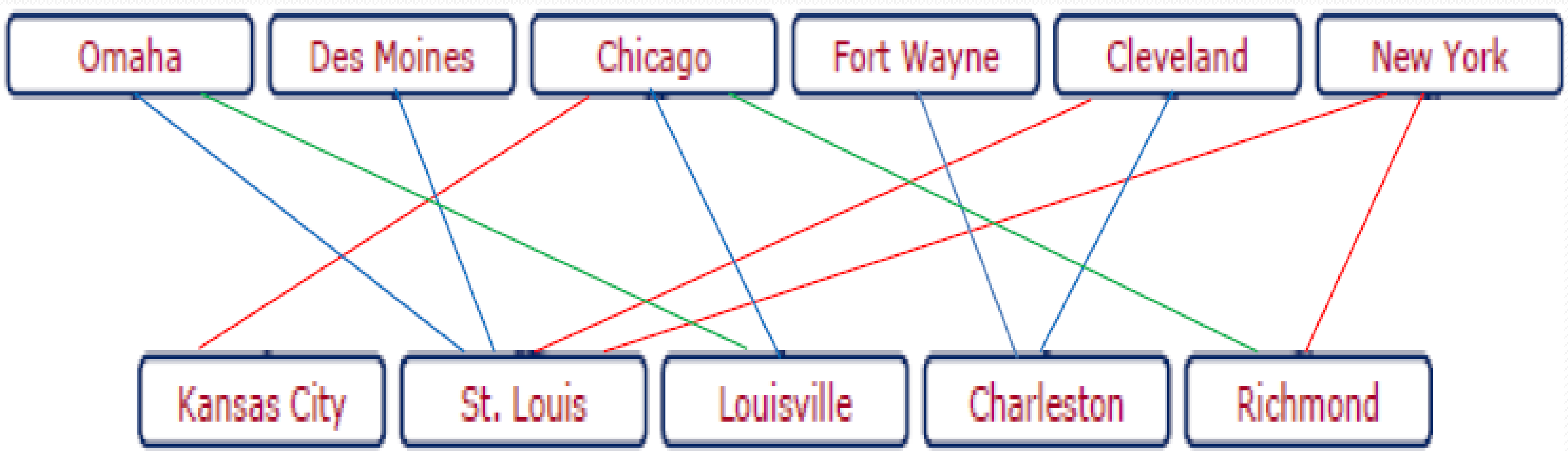
Coloring of Airline Routes

4	8	10	11

1	3	5	7	9

2	6

4	1	3	8	10	2	5	7	9	6	11



Thank you!


Colourbox