Web Algorithms – Sponsored Search

Eng. Fabio Persia, PhD

Matching Markets

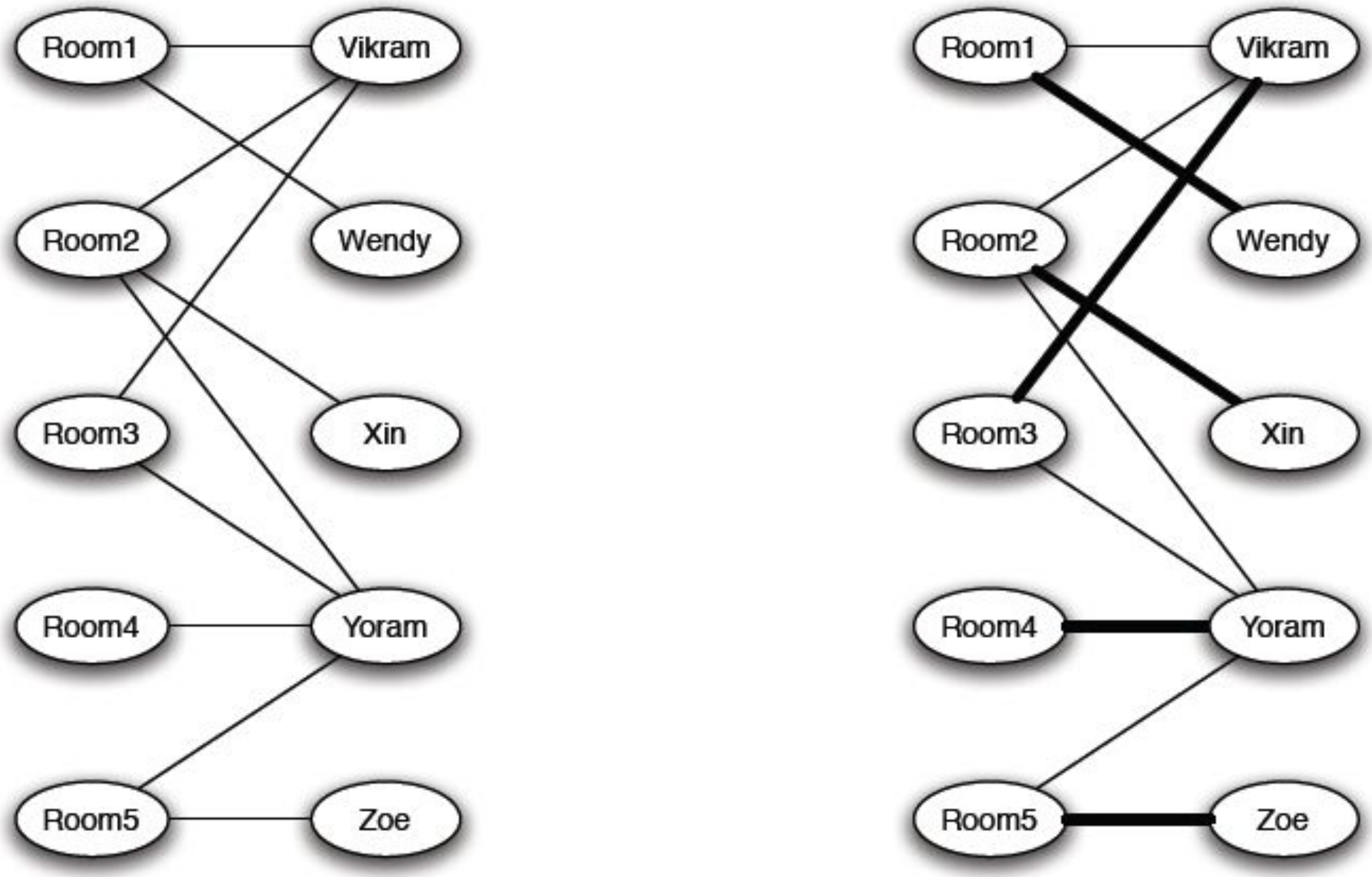
Prime example of network-structured interaction between many people/agents

Basic principles:

1. People may have different preferences for different kinds of goods
2. Prices can decentralize the allocation of goods to people
3. Prices can lead to allocations that are socially optimal

1st Scenario: Room Assigning

- Assigning rooms to students:
  - Each room is designed for a single student
  - Students may have different preferences over rooms



(a) *Bipartite Graph*

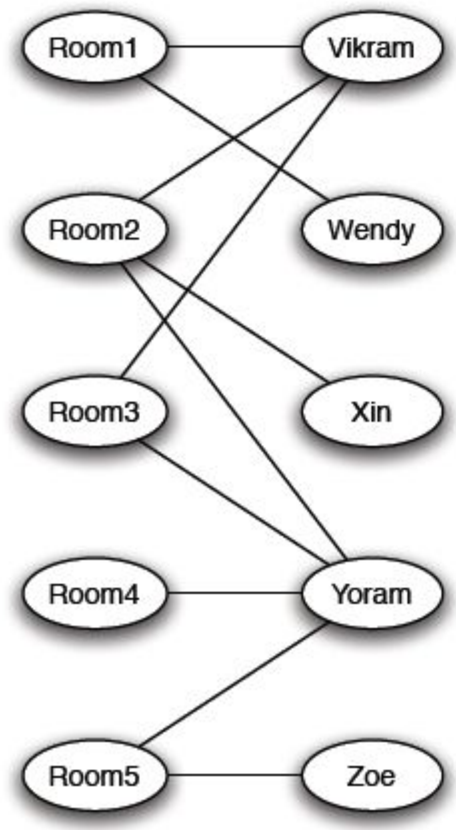
(b) *A Perfect Matching*

In order to provide a suitable formalization, let us recall some basic concepts:

- Bipartite graphs
- Perfect matchings
- Constricted sets
- The Hall Matching Theorem

Bipartite graph

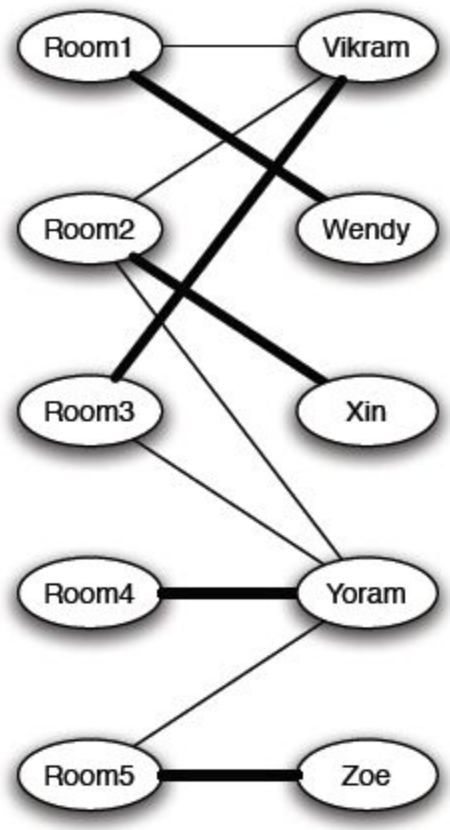
- **Nodes** are divided into two categories
- **Edges** connect nodes in one category to nodes in the other category



(a) *Bipartite Graph*

Perfect matching

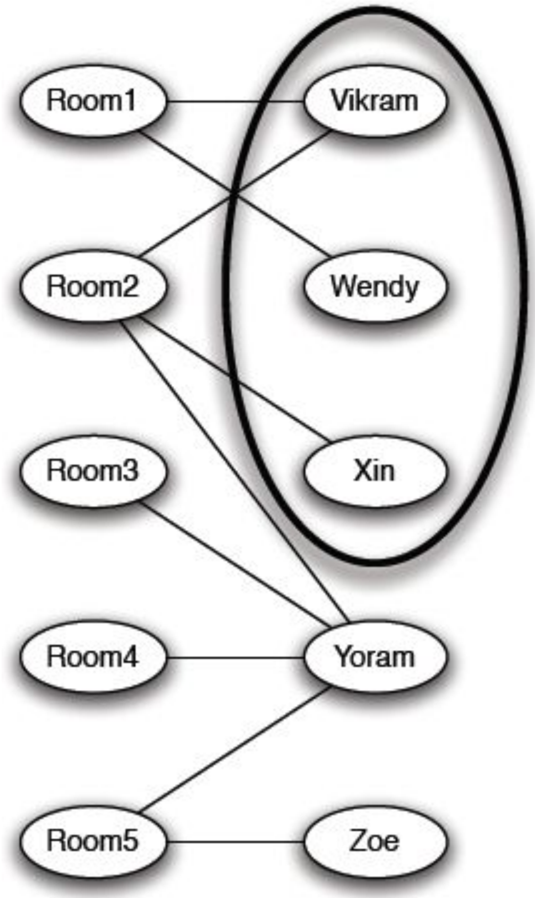
- A choice of edges in the bipartite graph so that each node is the endpoint of exactly one of the chosen edges
- In other words, a matching without isolated nodes



(b) *A Perfect Matching*

Constricted sets

- A set of nodes such that their edges on the other side of the bipartite graph “constrict” the formation of a perfect matching
- It implies the nonexistence of a perfect matching



- If *G* has a constricted set, then it does not admit a perfect matching

- Question: is the reverse also true? If *G* does not admit a perfect matching then *G* has a constricted set?

- Answer: yes, a famous theorem states that it in fact the *if and only* if holds

- Hall's Matching Theorem

  A bipartite graph *G* (with equal number of nodes on the left and right) has not a perfect matching if and only if it contains a constricted set

- Actually, the standard (equivalent) formulation of the theorem is:

  A bipartite graph *G* (with equal number of nodes on the left and right) has a perfect matching if and only if it does not contain a constricted set

Hall’s Matching Theorem

A bipartite graph *G* (with equal number of nodes on the left and right) has a not perfect matching if and only if it contains a constricted set

Proof.

Clearly if *G* contains a constricted set, then it cannot have a perfect matching

It remains to show that if there is not a perfect matching, then there is a constricted set

How can we identify a constricted set in a bipartite graph, knowing only that it does not have a perfect matching?

Idea:

1. Start from any (non perfect) matching *M*
2. Try to enlarge
3. **If** SUCCESS

    switch to enlarged matching and iterate

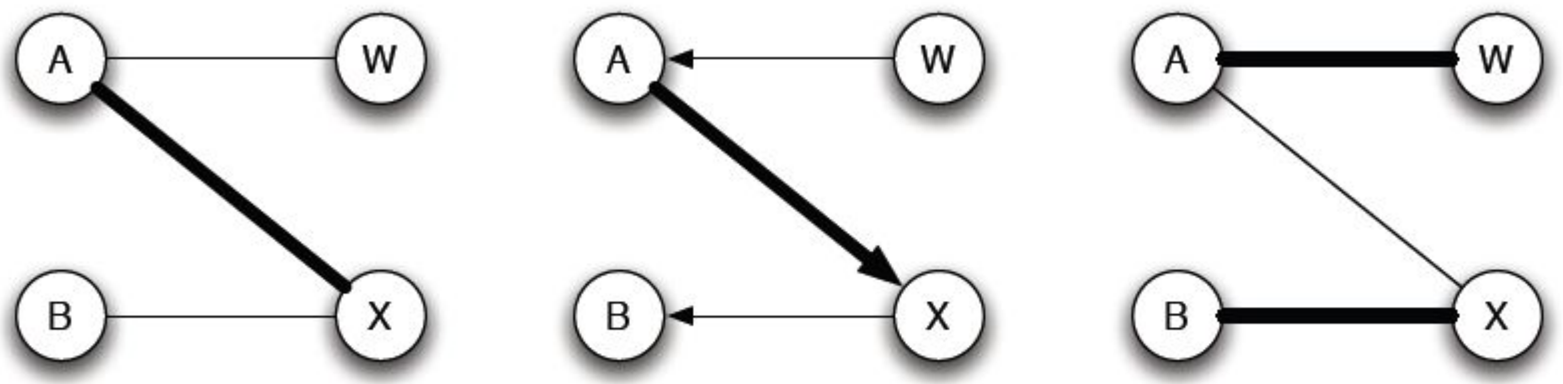
    else identify a constricted set

Clearly, by the hypothesis, we must finally arrive to a non perfect matching *M* which is not enlargeable, and thus determine a constricted set

Definitions:

1. **Matching edges:** edges used in given matching *M*
2. **Non-matching edges:** the other edges.
3. **Alternating path:** a simple path that alternates between non-matching and matching edges

4. **Augmenting path:** alternating path whose endpoints are unmatched nodes → matching *M* can be enlarged



(a) *A matching that is not of maximum size*

(b) *An augmenting path*

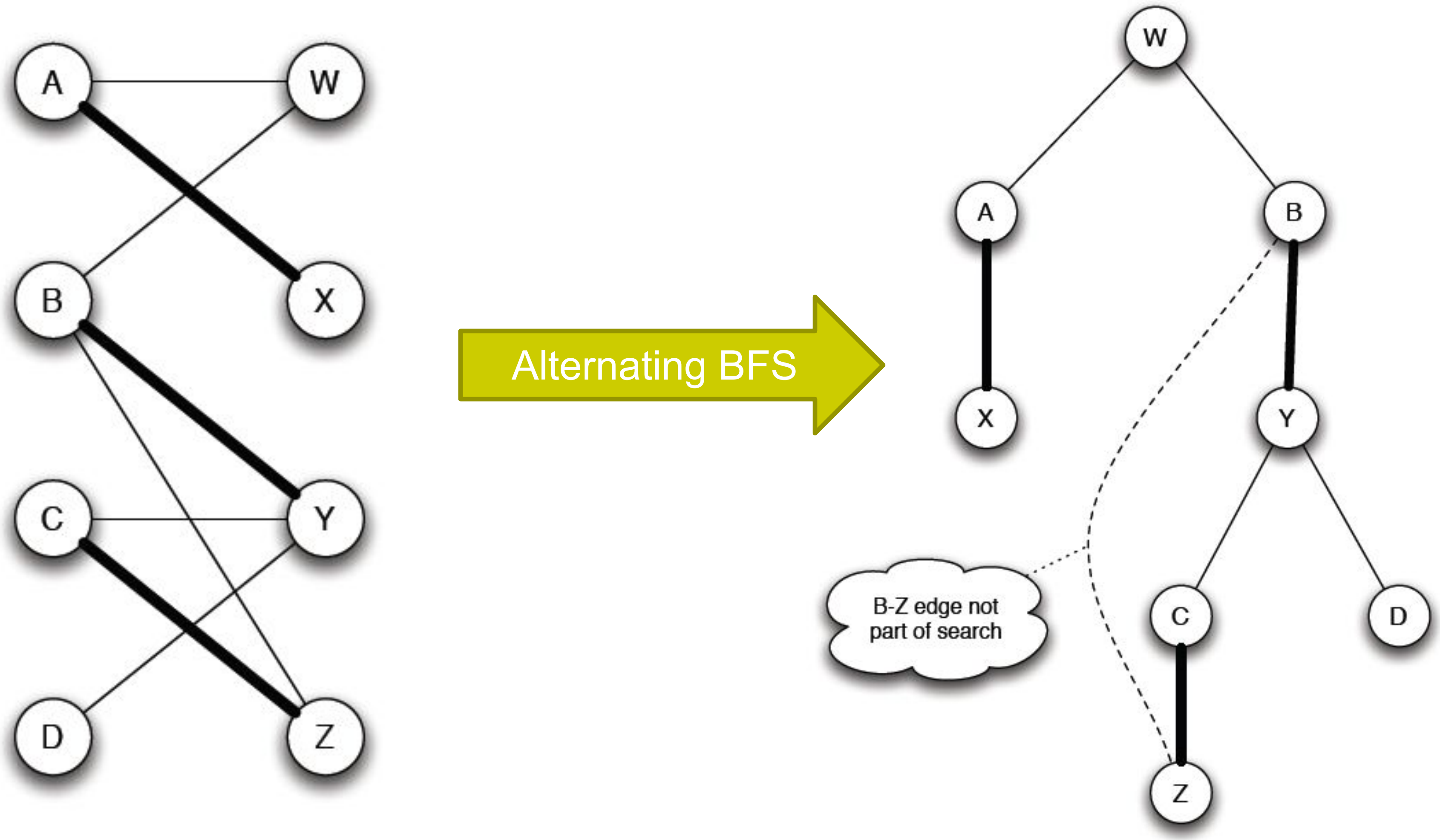
(c) *A larger (perfect) matching*

Searching for an augmenting path (starting from a non perfect matching *M*)

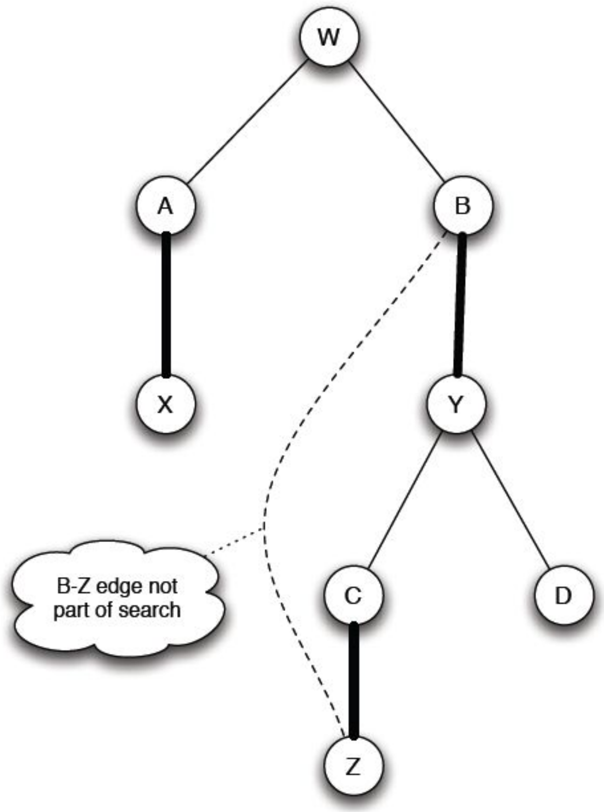
Alternating Breadth-First Search

- Start at any unmatched node *W* on the right (layer *0*)
- Explore the rest of the graph layer by layer, using non-matching edges at odd steps and matching edges at even steps
- At every step add new nodes to the next layer if there are corresponding edges

Let’s see an example of execution in the following figure …



A
W
B
X
C
Y
D
Z
Alternating BFS
W
A
B
X
Y
C
D
Z
B-Z edge not part of search

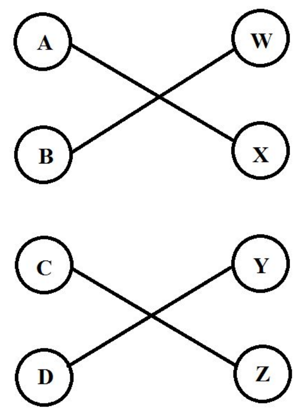


Augmenting path

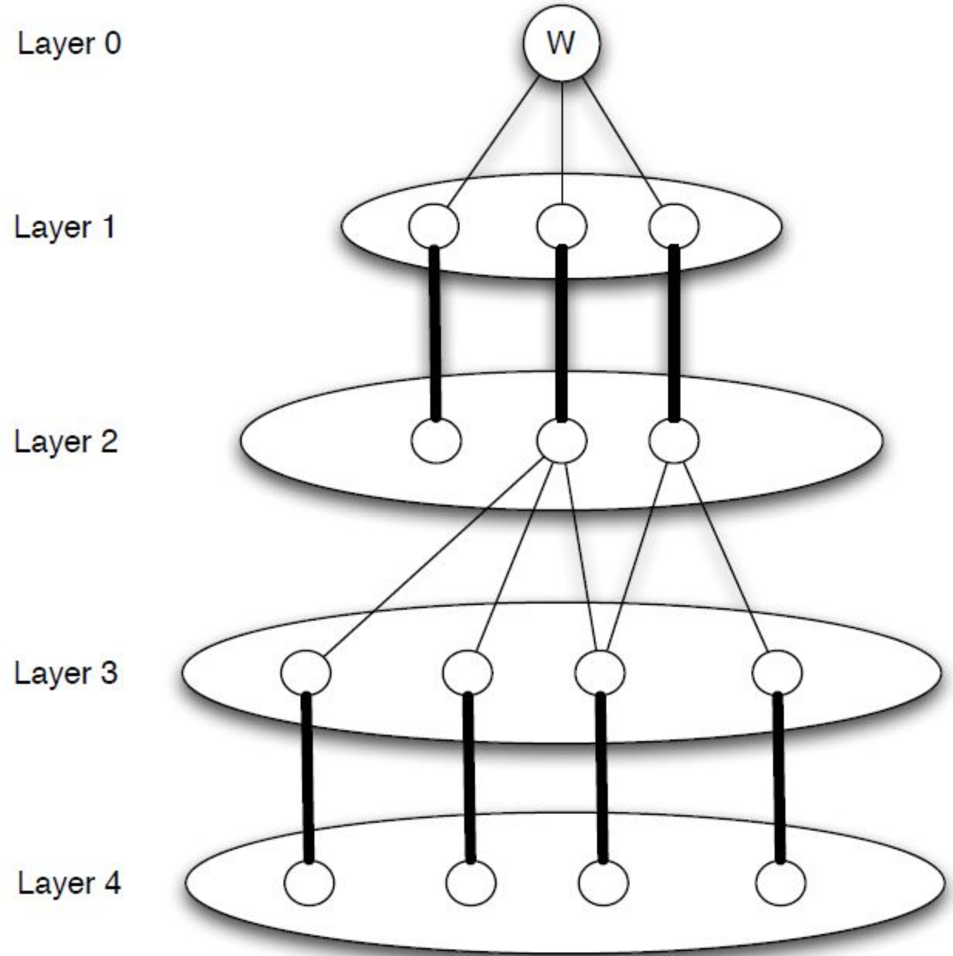
- W -> B -> Y -> D

Augmented matching

- Swapping matching and non-matching edges in the augmenting path



The BFS tree ….



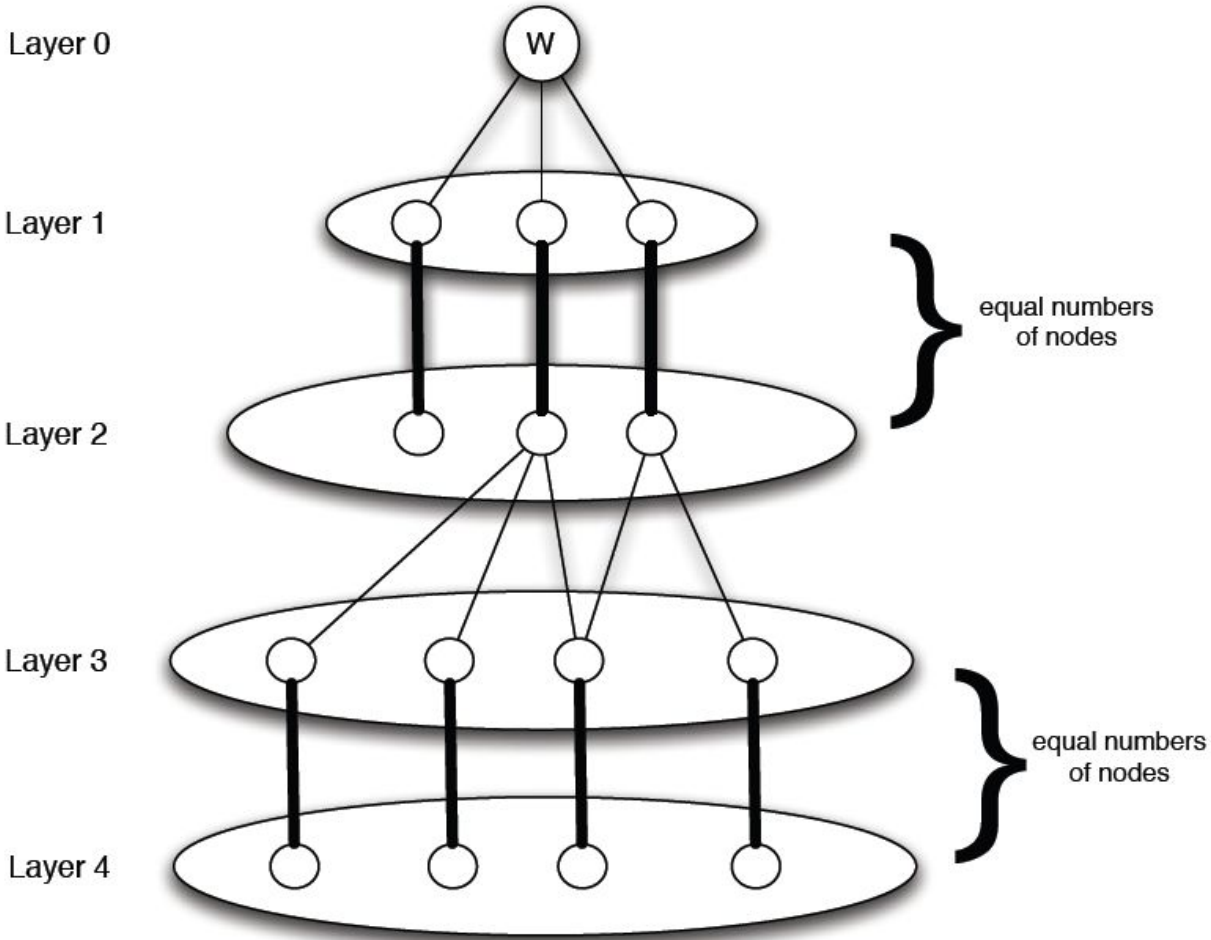
Properties of BFS tree:

1. Even layers contain nodes on the right-hand side of the bipartite graph (buyers)
2. Odd layers contain nodes on the left-hand side of the bipartite graph (sellers)
3. If there is an unmatched node *Z* in an odd layer, then the path from *W* to *Z* in BFS tree is an augmenting path

Thus, if there is an unmatched node *Z* in an odd layer, the current matching is enlargeable

We can then switch to the enlarged matching and iterate

Assume then that in the BFS tree there is not any unmatched node in an odd layer



Under the assumption of no unmatched nodes in odd layers:

1. Number of nodes in any odd layer i = number of nodes in even layer $i+1$ (since we follow matching edges)
2. Thus, not counting node W in layer 0, number of nodes in all odd layers = number of nodes in all even layers
3. Every node in an even layer has all its neighbors in the graph occurring in the odd layers of the BFS tree:
   - Its matched neighbor in previous layer
   - All the remaining neighbors not occurring in previous layers are in next layer
4. Constricted set: W and all the nodes in even layers (one more than ones in odd layers)

So we have finally proven the following claim (hence the Hall's Matching theorem)

Claim: Consider any bipartite graph *G* (with equal number of nodes on the left and right) and a non perfect matching *M*, and let *W* be any unmatched node on the right-hand side.

Then either there is an augmenting path beginning at *W*, or there is a constricted set containing *W*.

Computing a Perfect Matching

Remark: The previous proof also provides an algorithm for determining an eventual perfect matching in a bipartite graph:

1. Start with an empty matching
2. Look for an unmatched node *W* on the right
3. Use alternating BFS to search for an augmenting path beginning at *W*
4. If found, use this path to enlarge the matching and iterate, else indicate the constricted set

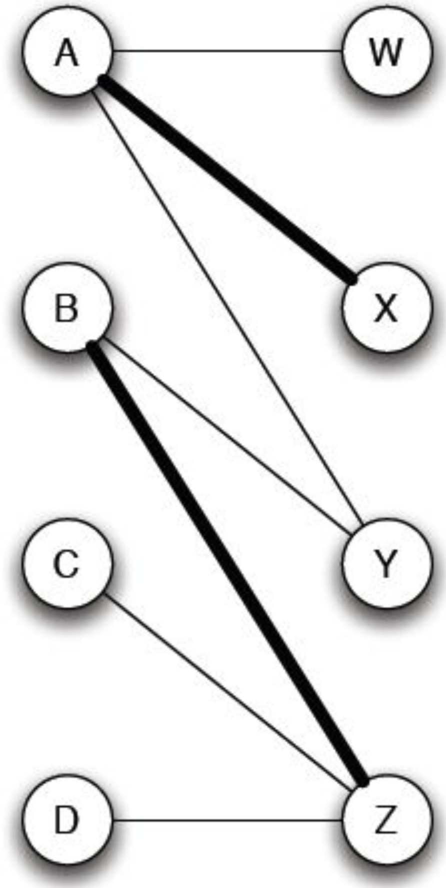
So, either we end up with a perfect matching, or we can provide a constricted set

Computing a Maximum Matching

Consider graph in figure

- Starting from W, we fail to find augmenting path.

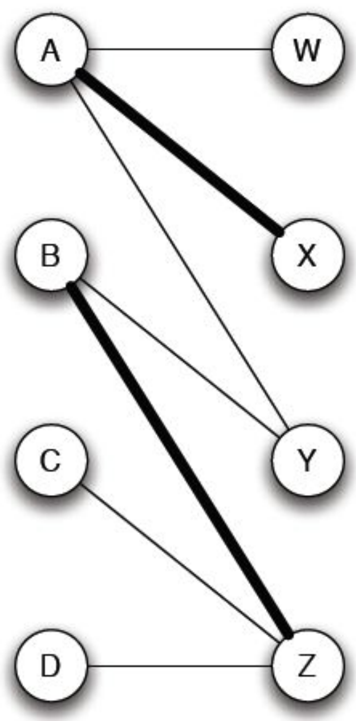
- Starting from Y, we can produce the path *Y – B – Z – D.*



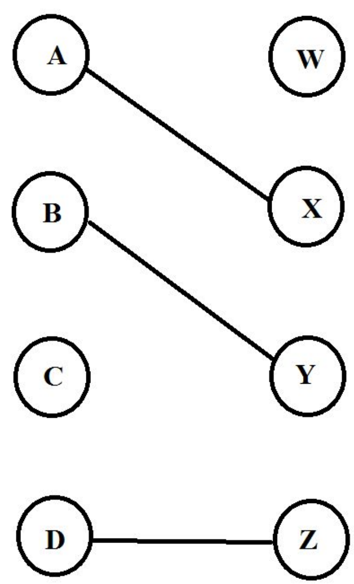
Computing a Maximum Matching

- Starting from Y, we can produce the path *Y – B – Z – D.*

OLD MATCHING



NEW MATCHING



- Remark: if there is not an augmenting path from a specific unmatched node *W* in the right, it doesn't mean that the matching is *maximum*

- However, if there is no augmenting path beginning from any unmatched node on the right, the current matching has maximum size

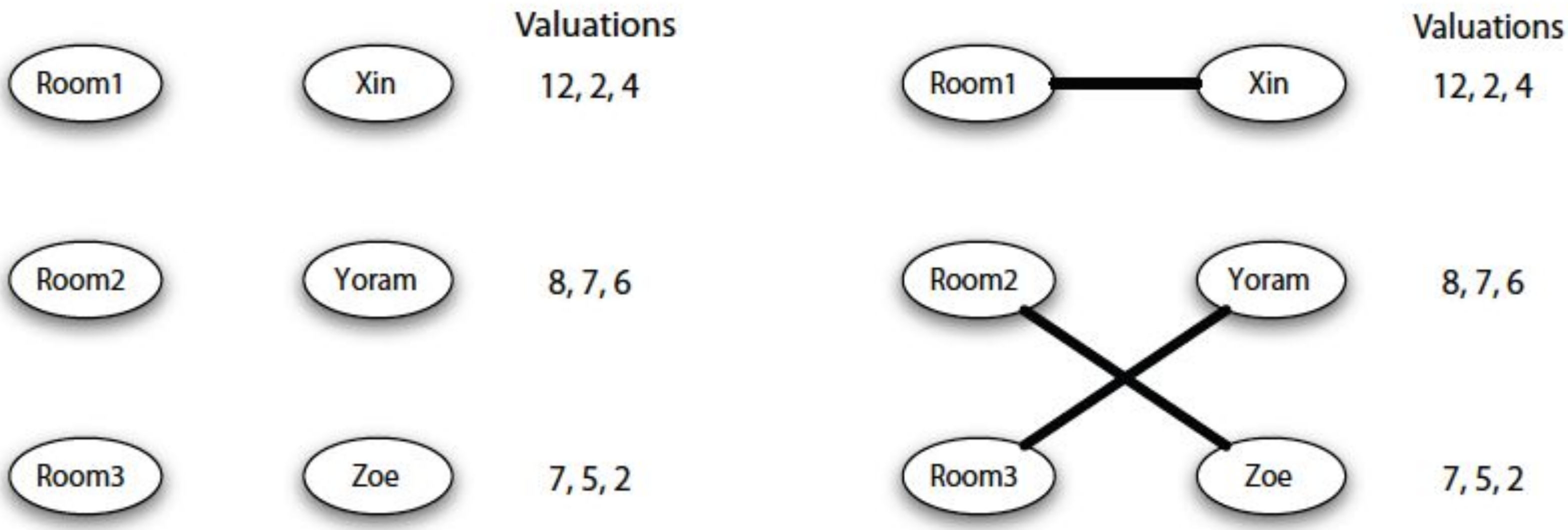
- The previous procedure can then be adapted to compute also a maximum matching: modify the alternating BFS by putting all the unmatched nodes on the right to layer 0

Back to Room Assigning: Refinement with Valuations

- More than a binary choice "accept-or-not"
- Valuations:
  - individuals evaluating a objects
  - numbers expressing degree of preference

Quality of an assignment of objects to individuals $\equiv$ Sum of each individual's valuation for what he gets

- Optimal assignment: an assignment that maximizes the total happiness of everyone



(a) *A set of valuations* (b) *An optimal assignment*

- Quality of the assignment: 12+6+5 = 23

Market-Clearing Prices

The previous scenario assumed a "central administrator" determining a perfect matching or an optimal assignment

A typical market instead consists of individuals making independent or autonomous free choices based on prices and valuations

Question: how can we decentralize markets using prices?