Stable Matchings and Faster Bipartite Matching

Dr. Rajiv Misra
Associate Professor
Dept. of Computer Science & Engg.
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Preface

Recap of Previous Lecture:

- In the previous lecture, we have discussed Weighted Bipartite Matching, Transversal, Equality subgraph and Hungarian Algorithm.

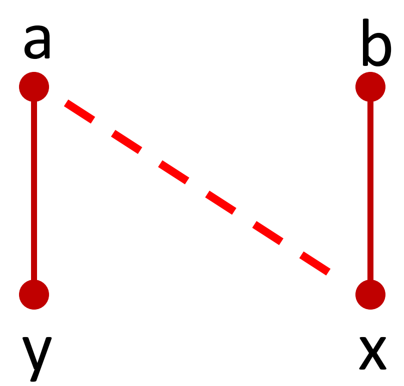
Content of this Lecture:

- In this lecture, we will discuss Stable Matchings, Gale-Shapley Algorithm and Faster Bipartite Matching *i.e.* Hopcroft-Karp algorithm

Stable Matchings

Stable Matchings

- Instead of optimizing total weight for a matching, we may try to optimize preferences. Given *n* men and *n* women; we want to establish *n* "stable" marriages.
- If man *x* and woman *a* are paired with other partners, but *x* prefers *a* to his current partner and a prefers *x* to her current partner, then they might leave their current partners and a prefers *x* to her current partner, then they might leave their current partners and switch to each other. In this situation we say that the unmatched pair (*x, a*) is an **unstable pair**.
- **Example:**

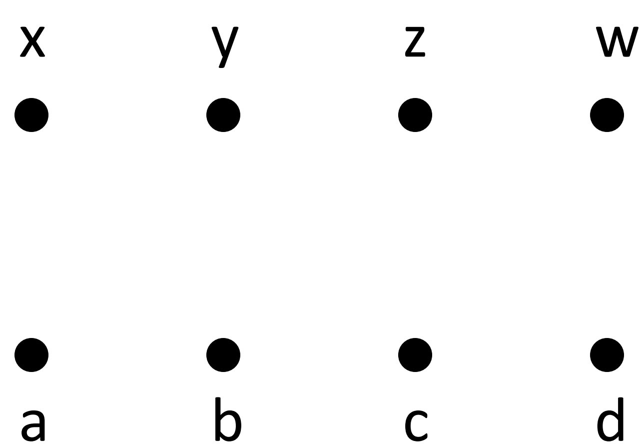


For a: x > y

For x: a > b

Definition

- A **perfect matching is a stable matching** if it yields **no unstable unmatched pair.**
- **Example:** Given men *x, y, z, w,* women *a, b, c, d,* and preferences listed below, the matching {*xa, yb, zd, wc*} is a stable matching.



Men {x, y, z, w}

x: a >b >c >d
y: a >c >b >d
z: c >d >a >b
w: c >b >a >d

Women {a, b, c, d}

a: z >x >y >w
b: y >w >x >z
c: w >x >y >z
d: x >y >z >w

Algorithm: Gale-Shapley Proposal Algorithm 3.2.17

Input: Preference rankings by each of n men and n women.

Idea: Produce a stable matching using proposals by maintaining information about who has proposed to whom and who has rejected whom.

Iteration: (i)Each man proposes to the highest woman on his preference list who has not previously rejected him.

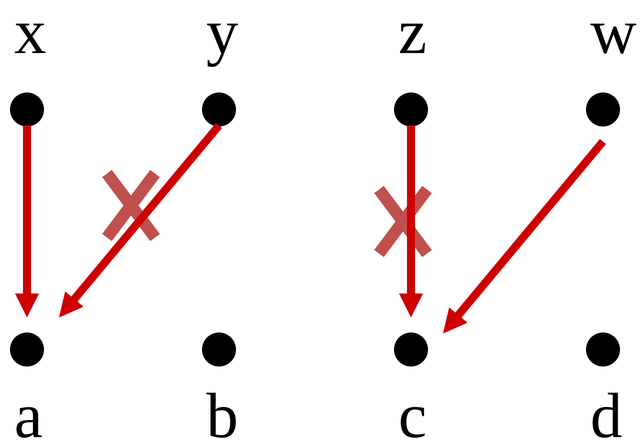
(ii) If each woman receives exactly one proposal, stop and use the resulting matching.

(iii) Otherwise, every woman receiving more than one proposal rejects all of them except the one that is highest on her preference list.

(iv) Every woman receiving a proposal says “maybe” to the most attractive proposal received.

Stable matching: Example

Step 1:



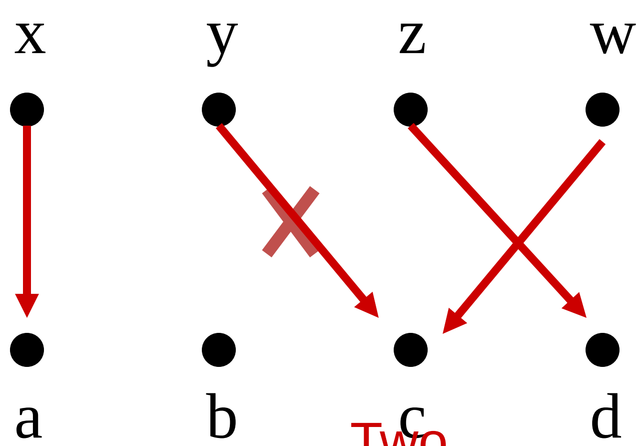
Two proposer, refuse the less like one

x: a b c d
y: ~~a~~ c b d
z: ~~c~~ d a b
w: c b a d

a: z x y w
b: y w x z
c: w x y z
d: x y z w

Stable matching: Example

Step 2:



Two proposer, refuse the less like one

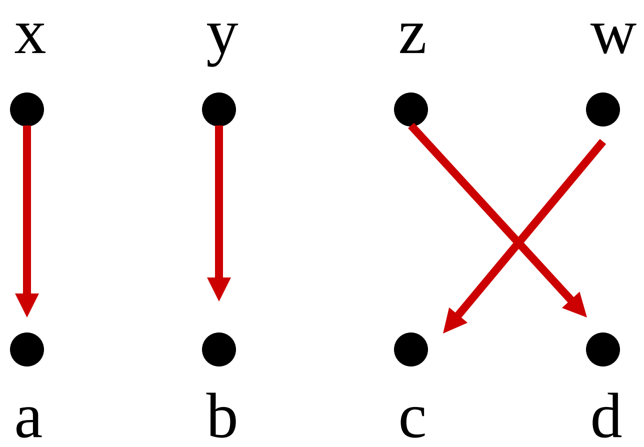
x: a b c d
y: ~~a~~ ~~c~~ b d
z: ~~c~~ d a b
w: c b a d

a: z x y w
b: y w x z
c: w x y z
d: x y z w

Observation: X moves right and O moves left.

Stable matching: Example

Step 3:



x: a b c d
y: ~~a~~ ~~c~~ b d
z: ~~c~~ d a b
w: c b a d

a: z x y w
b: y w x z
c: w x y z
d: x y z w

Every woman gets only one proposal. Agree and stop.

Example Stable Matching

$M = \{(1,4), (2,3), (3,2), (4,1)\}$

1:	2	4	1	3	1:	2	1	4	3
2:	3	1	4	2	2:	4	3	1	2
3:	2	3	1	4	3:	1	4	3	2
4:	4	1	3	2	4:	2	1	4	3

Men's preferences Women's preferences

Example Stable Marriage Instance

1:	2	4	1	3	1:	2	1	4	3
2:	3	1	4	2	2:	4	3	1	2
3:	2	3	1	4	3:	1	4	3	2
4:	4	1	3	2	4:	2	1	4	3

Men's preferences | Women's preferences

Analysis of the Gale-Shapley Algorithm

- The algorithm terminates after no more than ***n^2* iterations with a stable marriage output.**

- The stable matching produced by the algorithm is always ***gender-optimal***: each man gets the highest rank woman on his list under any stable marriage. One can obtain the ***woman-optimal*** matching by making women propose to men
- A man (woman) optimal matching is unique for a given set of participant preferences
- The stable marriage problem has practical applications such as matching medical-school graduates with hospitals for residency training

Stable Marriage Formalisation

- Set n men $S_M = \{m_1, m_2, ..., m_n\}$
- Set n women $S_W = \{w_1, w_2, ..., w_n\}$

- Each man ranks the women in S_W in strict order of preference.
- Each woman ranks the men in S_M in strict order of preference.

- A matching M is a bijection between the men and women.
- We say a (man, woman) pair (m,w) **blocks *M*** if:
  - m prefers w to his partner in M, and
  - w prefers m to her partner **in** M.
- A matching that admits no blocking pair is said to be **stable**
  - Can't improve by making an arrangement outside the matching.
- Stable Marriage was first formalised by **David Gale** and **Lloyd Shapley** in 1962.

Structure of Stable Marriage

- **The Gale-Shapley (GS) algorithm has two possible orientations:**
  - **man-oriented** GS (MEGS) algorithm : where the men propose to the women.
  - **women-oriented** GS (WEGS) algorithm : where the women propose to the men.

- **Optimality properties of each algorithm:**
  - MEGS algorithm - **man-optimal** stable matching M_0 - simultaneously the best possible stable matching for all men.
  - WEGS algorithm **-** **woman-optimal** stable matching M_z- simultaneously the best possible stable matching for all women.

- Deletions that occur during an execution of either algorithm result in a set of reduced preference lists on termination of each algorithm:
  - The **MGS-lists** for the MEGS algorithm.
  - The **WGS-lists** for the WEGS algorithm.

- The intersection of the MGS-lists and the WGS-lists is called the **GS-lists.**
- The GS-lists have many important structural properties.

Faster Bipartite Matching

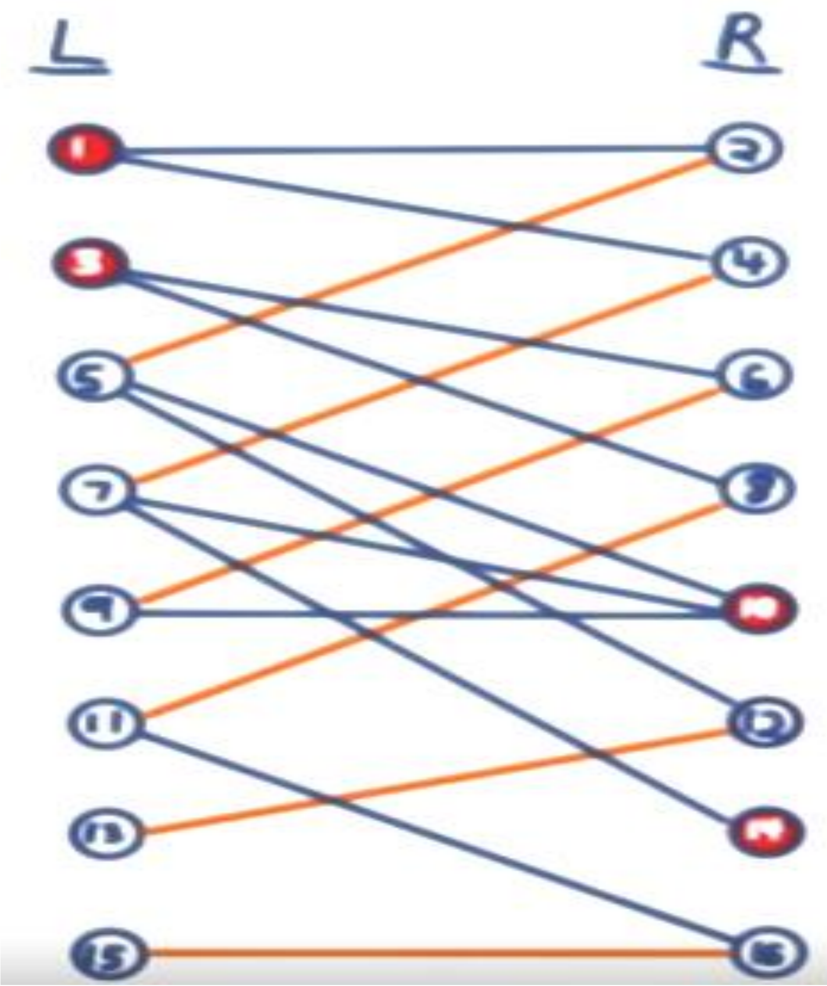
Faster Bipartite Matching: Hopcroft and Karp [1973]

- The running time of an algorithm for finding maximum matchings in bipartite graphs can be improved by seeking augmenting paths in a clever order; when short augmenting paths are available, we needn't explore many edges to find one.
- Using a **Breadth-First Search** simultaneously from all the unsaturated vertices of X, we can find many paths of the same length with one examination of the edge set.
- **Hopcroft and Karp [1973]** proved that subsequent augmentations must use longer paths, so the searches can be grouped in phases finding paths of the same lengths. They combined these ideas to show that few phases are needed, enabling maximum matchings in n-vertex bipartite graphs to be found in **$O(n^{2.5})$ time.**

Hopcroft-Karp Algorithm

1. Initialize M = Ø //M is matching
2. Repeat
   -Build alternating level graph rooted at unmatched vertices in L using breadth first search (BFS) // L is the left part of the partition and R is the right part

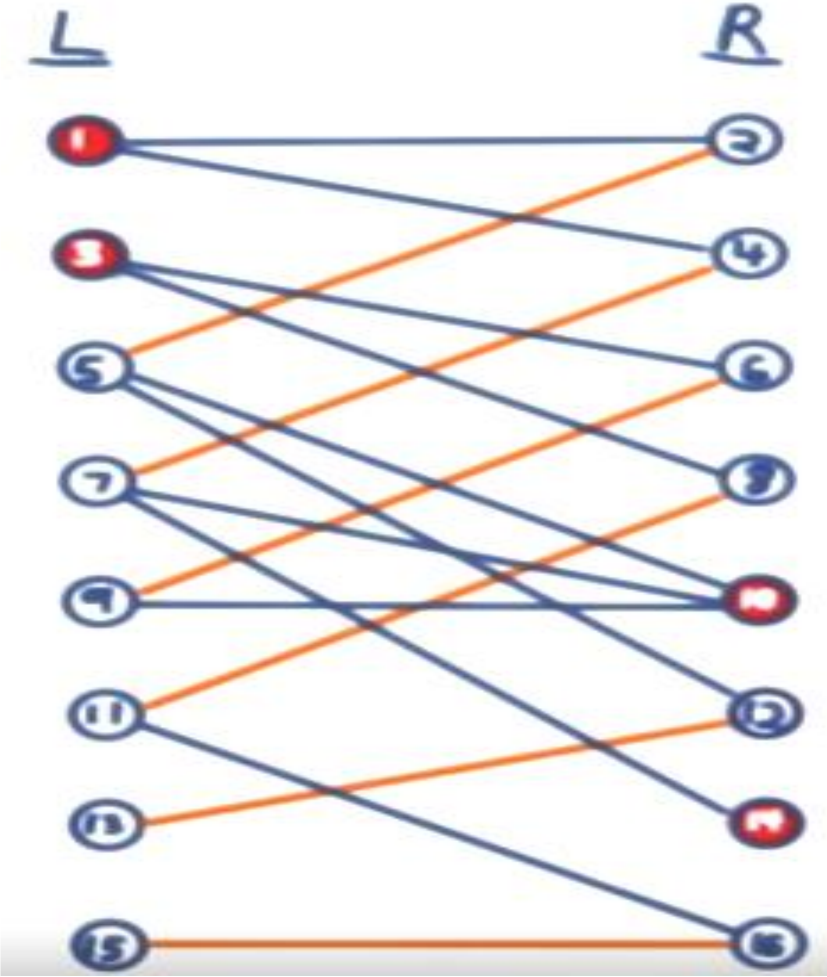
Example:



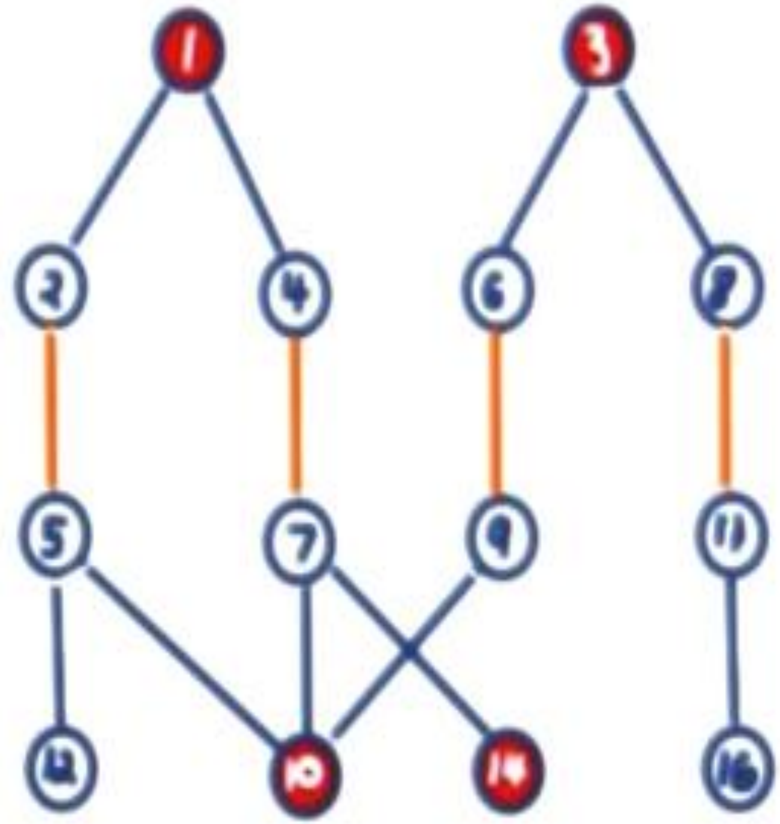
Orange Edge: Existing Matchings

Note: L is the left part and R is the right part of the partition

Example: Alternative level graph using BFS



Orange Edge: Existing Matchings



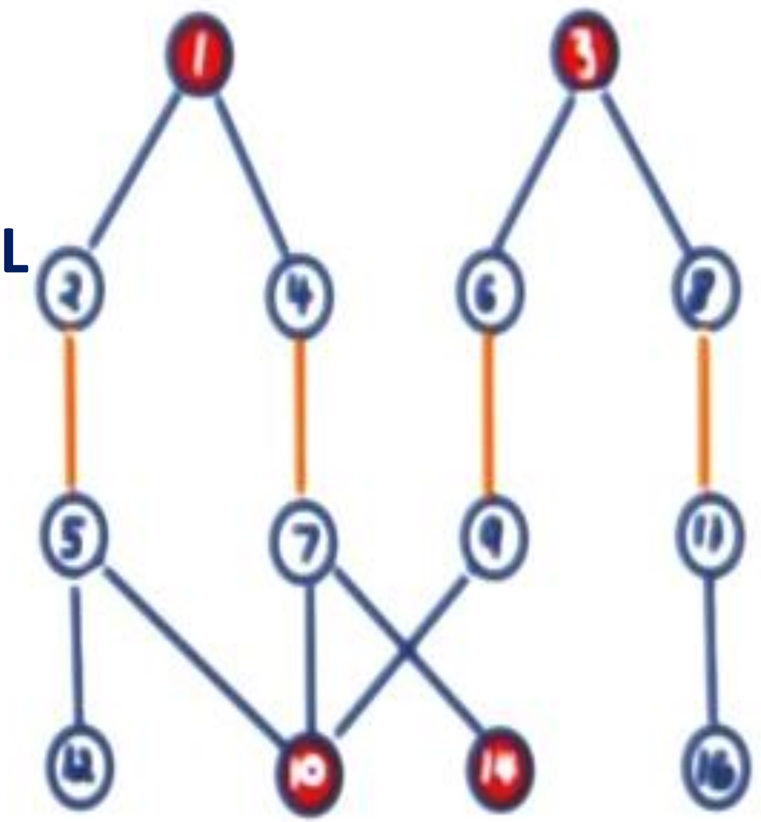
Alternative level graph using breadth first search (BFS)

Hopcroft-Karp Algorithm

1. Initialize M = Ø
2. Repeat

   -Build alternating level graph rooted at unmatched vertices in L using breadth first search (BFS)

   -Augment M with a maximal set of vertex disjoint shortest-length paths.

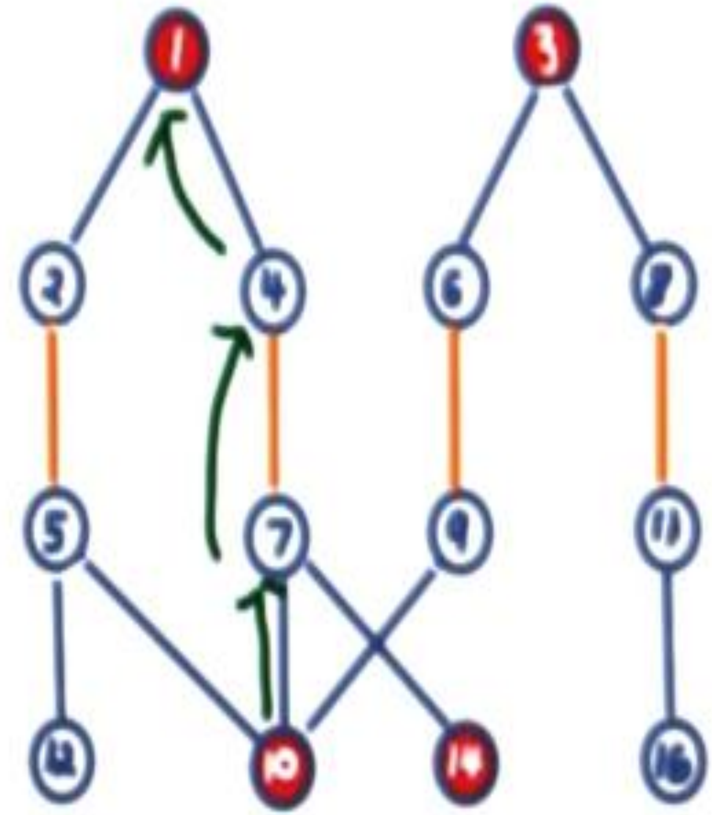


Hopcroft-Karp Algorithm

1. Initialize M = Ø
2. Repeat

   -Build alternating level graph
   rooted at unmatched vertices in L
   using breadth first search (BFS)

   -**Augment M with a maximal set of vertex disjoint shortest-length paths.**

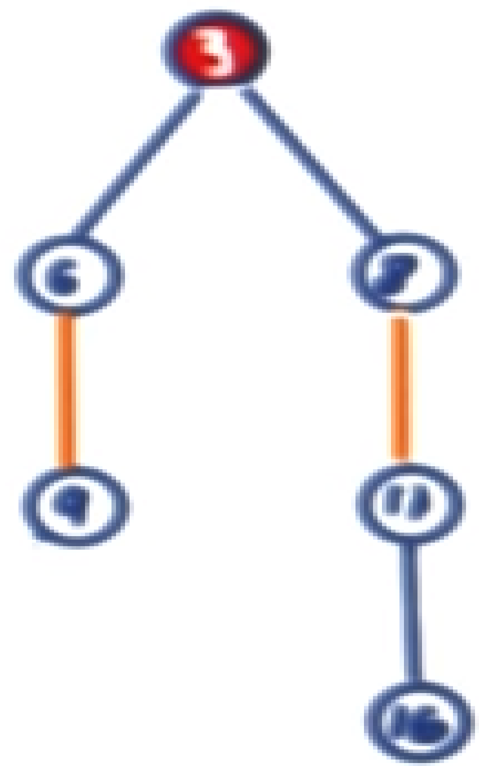


Hopcroft-Karp Algorithm

1. Initialize M = Ø
2. Repeat

   -Build alternating level graph
   rooted at unmatched vertices in L
   using breadth first search (BFS)

   -Augment M with a maximal
   set of vertex disjoint
   shortest-length paths.



Hopcroft-Karp Algorithm

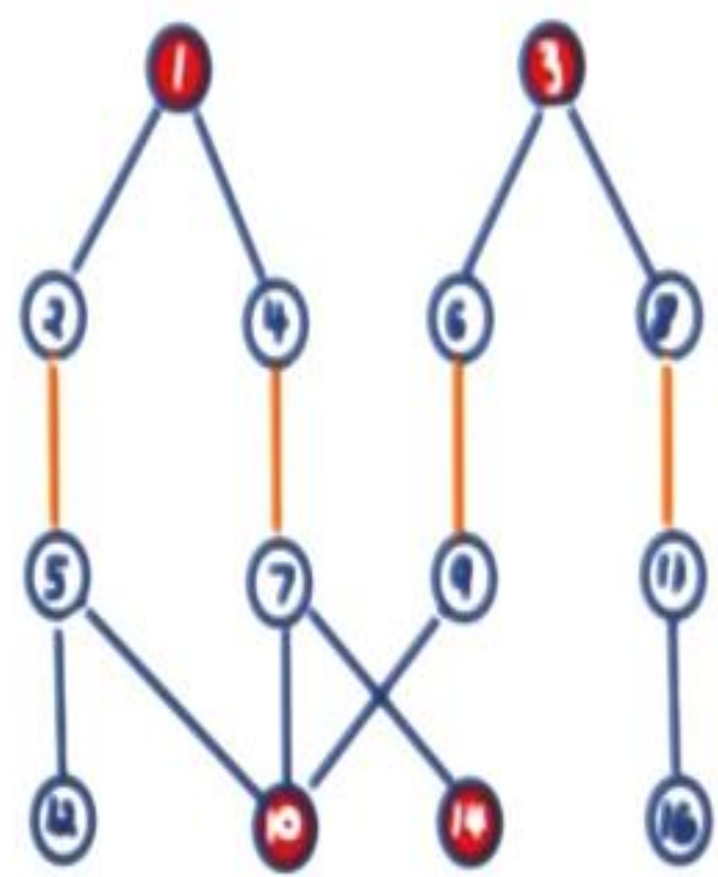
1. Initialize M = Ø
2. Repeat

   -Build alternating level graph rooted at unmatched vertices in L using breadth first search (BFS)

   -Augment M with a maximal set of vertex disjoint shortest-length paths.

   Until no augmenting paths exist

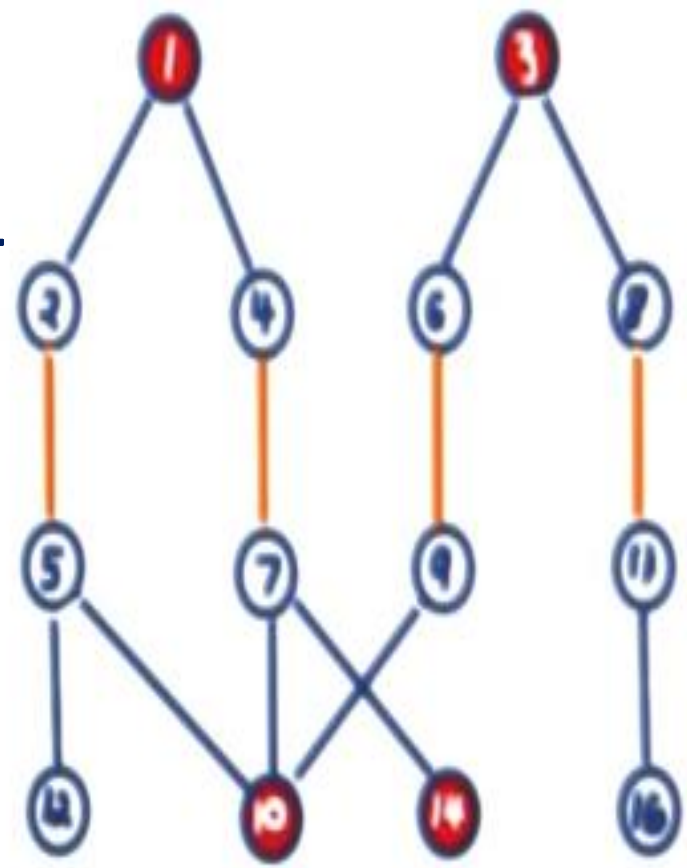
3. Return M.



Hopcroft-Karp Algorithm

1. Initialize M = Ø
2. Repeat ← O(|V|)^{½} "phases"
   - Build alternating level graph rooted at unmatched vertices in L using breadth first search (BFS)
   - Augment M with a maximal set of vertex disjoint shortest-length paths.

   O(|E|) Until no augmenting paths exist
3. Return M.



Theorem 3.2.22

- Given a bipartite graph $G=(V,E)$, Hopcroft-karp finds a maximum matching in time $O(|E||V|^{1/2})$ with V vertices and E edges.

Conclusion

- In this lecture, we have discussed Stable Matchings, Gale-Shapley Algorithm and Faster Bipartite Matching *i.e.* Hopcroft-Karp algorithm.

- In upcoming lectures, we will discuss Matchings in General Graphs and also discuss Tutte's 1-Factor Theorem, *f*-Factor of Graphs, Edmonds' Blossom Algorithm.