CS 105: Department Introductory Course on Discrete Structures

Instructor : S. Akshay

Graph theory

Perfect Matchings and Vertex Covers, stable matchings

Lecture 23
Oct 28 2025

Applications of Hall's condition

- Matching: set of edges with no shared end-points.
- The vertices incident to edges in a matching are called saturated. Others are unsaturated.
- Perfect matching: saturates every vertex in graph.

Theorem (Hall'35)

A bipartite graph G with bipartitions X, Y has a matching that saturates X iff for all $S \subseteq X$, $|N(S)| \geq |S|$.

Applications of Hall's condition

- Matching: set of edges with no shared end-points.
- The vertices incident to edges in a matching are called saturated. Others are unsaturated.
- Perfect matching: saturates every vertex in graph.

Theorem (Hall'35)

A bipartite graph G with bipartitions X, Y has a matching that saturates X iff for all $S \subseteq X$, $|N(S)| \geq |S|$.

Application 1: Marriage theorem

- Consider n women and n men. If every woman is compatible with k men and every man compatible with k women, then a perfect matching must exist.
- For $k > 0$, every k-regular bipartite graph (i.e, every vertex has degree exactly k) has a perfect matching.

Application 2: Games on graphs

A two player game on a graph

1. Given a graph G, two players will alternatively choose distinct vertices.
2. One player starts by choosing any vertex.
3. Subsequent move must be adjacent to preceding choice (of other player).
4. Last player who can move wins.

Application 2: Games on graphs

A two player game on a graph

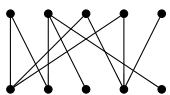
1. Given a graph G, two players will alternatively choose distinct vertices.
2. One player starts by choosing any vertex.
3. Subsequent move must be adjacent to preceding choice (of other player).
4. Last player who can move wins.

Application 2: Theorem (H.W: Qn 3.1.18 from Douglas West)

If G has a perfect matching, then player 2 has a winning strategy; otherwise, player 1 has a winning strategy.

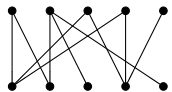
Application 3: Another game

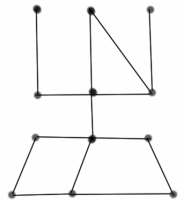
Consider the graph of a road network in IITB. When a VVIP is visiting, the Chief of Security wants to place a security guard to watch every road. What is the minimum number of guards required?



Application 3: Another game

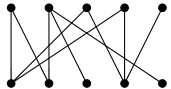
Consider the graph of a road network in IITB. When a VVIP is visiting, the Chief of Security wants to place a security guard to watch every road. What is the minimum number of guards required?





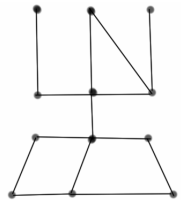
Application 3: Another game

Consider the graph of a road network in IITB. When a VVIP is visiting, the Chief of Security wants to place a security guard to watch every road. What is the minimum number of guards required?



X

Y



Definition

A vertex cover of a graph G is a set $Q \subseteq V$ that contains at least one endpoint of every edge. Vertices in Q are said to cover E.

Matchings and vertex covers

So, what is the link between matchings and vertex covers?

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.
- The end-points of a maximal matching

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.
- The end-points of a maximal matching form a vertex cover.

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.
- The end-points of a maximal matching form a vertex cover.
- Size of any vertex cover *vs* size of any matching?

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.
- The end-points of a maximal matching form a vertex cover.
- Size of any vertex cover *vs* size of any matching?

Questions/Pop-Quiz!

1. What is the size of the minimum vertex cover in K_m, $K_{m,n}$?
2. If ℓ is size of any matching and k is size of any vertex cover,
   - 2.1 how are ℓ, k related?
   - 2.2 Give an example of a graph where $k \neq \ell$

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.
- The end-points of a maximal matching form a vertex cover.
- Size of any vertex cover *vs* size of any matching?

Questions/Pop-Quiz!

1. What is the size of the minimum vertex cover in K_m, $K_{m,n}$?
2. If ℓ is size of any matching and k is size of any vertex cover,
   - 2.1 how are ℓ, k related?
   - 2.2 Give an example of a graph where $k \neq \ell$

So, how do you compute min no. of guards required, i.e., the size of the minimum vertex covers?

Matchings and vertex covers

So, what is the link between matchings and vertex covers?
First, does a graph always have vertex cover?

Examples and properties of vertex covers

- The set of all vertices is always a vertex cover.
- The end-points of a maximal matching form a vertex cover.
- Size of any vertex cover *vs* size of any matching?

Questions/Pop-Quiz!

1. What is the size of the minimum vertex cover in K_m, $K_{m,n}$?
2. If ℓ is size of any matching and k is size of any vertex cover,
   - 2.1 how are ℓ, k related?
   - 2.2 Give an example of a graph where $k \neq \ell$

So, how do you compute min no. of guards required, i.e., the size of the minimum vertex covers? Let's consider bipartite graphs...

A min-max theorem

Theorem (Konig '31, Egervary '31)

If G is a bipartite graph, then the size of the maximum matching of G equals the size of the minimum vertex cover of G.

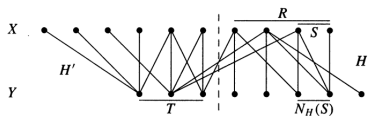
Proof.

- Suffices to show that we can achieve a matching which has size equal to min vertex cover, since $|Q| \geq |M|$ for any vertex cover Q, matching M.

A min-max theorem

Theorem (Konig '31, Egervary '31)

If G is a bipartite graph, then the size of the maximum matching of G equals the size of the minimum vertex cover of G.



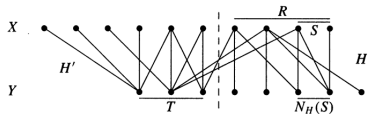
Proof.

- Suffices to show that we can achieve a matching which has size equal to min vertex cover, since $|Q| \geq |M|$ for any vertex cover Q, matching M.
- Take minimum vertex cover Q, partition into $R = Q \cap X$ and $T = Q \cap Y$.

A min-max theorem

Theorem (Konig '31, Egervary '31)

If G is a bipartite graph, then the size of the maximum matching of G equals the size of the minimum vertex cover of G.



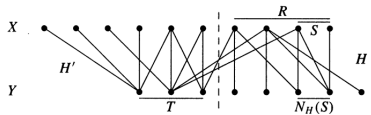
Proof.

- Suffices to show that we can achieve a matching which has size equal to min vertex cover, since $|Q| \geq |M|$ for any vertex cover Q, matching M.
- Take minimum vertex cover Q, partition into $R = Q \cap X$ and $T = Q \cap Y$.
- Consider subgraphs H, H' induced by $R \cup (Y \setminus T)$, $T \cup (X \setminus R)$.

A min-max theorem

Theorem (Konig '31, Egervary '31)

If G is a bipartite graph, then the size of the maximum matching of G equals the size of the minimum vertex cover of G.



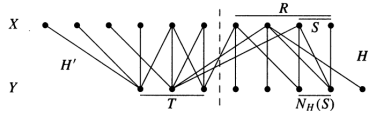
Proof.

- Suffices to show that we can achieve a matching which has size equal to min vertex cover, since $|Q| \geq |M|$ for any vertex cover Q, matching M.
- Take minimum vertex cover Q, partition into $R = Q \cap X$ and $T = Q \cap Y$.
- Consider subgraphs H, H' induced by $R \cup (Y \setminus T)$, $T \cup (X \setminus R)$.
- Show that H has a matching that saturates $Q \cap X$ into $Y \setminus T$, H' has a matching saturating T. (Use minimality of Q).

A min-max theorem

Theorem (Konig '31, Egervary '31)

If G is a bipartite graph, then the size of the maximum matching of G equals the size of the minimum vertex cover of G.



Proof.

- Suffices to show that we can achieve a matching which has size equal to min vertex cover, since $|Q| \geq |M|$ for any vertex cover Q, matching M.
- Take minimum vertex cover Q, partition into $R = Q \cap X$ and $T = Q \cap Y$.
- Consider subgraphs H, H' induced by $R \cup (Y \setminus T)$, $T \cup (X \setminus R)$.
- Show that H has a matching that saturates $Q \cap X$ into $Y \setminus T$, H' has a matching saturating T. (Use minimality of Q).
- Together this forms the desired matching (since H, H' are disjoint).

Many min max theorems exist

Dual problems

- Minimizing one thing is same as maximizing another!
- Very useful in optimization problems.

Many min max theorems exist

Dual problems

- Minimizing one thing is same as maximizing another!
- Very useful in optimization problems.

Before we go to the last topic on graph theory, let us do a quick summary of whole course!

Summary

- Part 1: Mathematical proofs and reasoning
- Part 2: Basic discrete structures

Summary

- Part 1: Mathematical proofs and reasoning
  - How to reason and write proofs formally
  - Propositions, predicates
  - Proof techniques: contradiction, contrapositive
  - Induction: strong induction, well-ordering principle
- Part 2: Basic discrete structures

Summary

- Part 1: Mathematical proofs and reasoning
  - How to reason and write proofs formally
  - Propositions, predicates
  - Proof techniques: contradiction, contrapositive
  - Induction: strong induction, well-ordering principle
  - Applications: Seen throughout this course and most future courses!
- Part 2: Basic discrete structures

Summary

- Part 1: Mathematical proofs and reasoning
  - How to reason and write proofs formally
  - Propositions, predicates
  - Proof techniques: contradiction, contrapositive
  - Induction: strong induction, well-ordering principle
  - Applications: Seen throughout this course and most future courses!
- Part 2: Basic discrete structures
  - Sets: finite and infinite sets, countable and uncountable sets
  - Functions: bijections (from e.g., $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$), injections and surjections, Cantor's diagonalization technique
  - Relations: equivalence relations and partitions; partial orders, chains, anti-chains, lattices

Summary

- Part 1: Mathematical proofs and reasoning
  - How to reason and write proofs formally
  - Propositions, predicates
  - Proof techniques: contradiction, contrapositive
  - Induction: strong induction, well-ordering principle
  - Applications: Seen throughout this course and most future courses!
- Part 2: Basic discrete structures
  - Sets: finite and infinite sets, countable and uncountable sets
  - Functions: bijections (from e.g., $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$), injections and surjections, Cantor's diagonalization technique
  - Relations: equivalence relations and partitions; partial orders, chains, anti-chains, lattices
  - Applications: showing impossibility theorems for CS, parallel task scheduling algorithms.

Summary Contd.

- Part 3: Counting and Combinatorics

Summary Contd.

- Part 3: Counting and Combinatorics
  - Basic counting principles, double counting
  - Binomial theorem, permutations and combinations, Estimating $n!$
  - Recurrence relations and generating functions
  - Principle of Inclusion-Exclusion (PIE) and its applications.
  - Pigeon-Hole Principle (PHP) and its applications.
  - Some special numbers and sequences: Fibonacci, Catalan
  - Introduction to Ramsey theory

Summary Contd.

- Part 3: Counting and Combinatorics
  - Basic counting principles, double counting
  - Binomial theorem, permutations and combinations, Estimating $n!$
  - Recurrence relations and generating functions
  - Principle of Inclusion-Exclusion (PIE) and its applications.
  - Pigeon-Hole Principle (PHP) and its applications.
  - Some special numbers and sequences: Fibonacci, Catalan
  - Introduction to Ramsey theory
  - Applications: Bounding the complexity of algorithms, proving existence of structures.

Summary (Contd.)

Part 4: Topics in Graph theory

- Basics: graphs, paths, cycles, walks, trails, ...
- Graph representations, isomorphisms and automorphisms.
- Matchings: maximal, maximum, perfect and stable.

Summary (Contd.)

Part 4: Topics in Graph theory

- Basics: graphs, paths, cycles, walks, trails, ...
- Graph representations, isomorphisms and automorphisms.
- Matchings: maximal, maximum, perfect and stable.

Graph theory: Characterizations

1. Eulerian graphs: Using degrees of vertices.
2. Bipartite graphs: Using odd length cycles.
3. Connected components: Using cycles.
4. Maximum matchings: Using augmenting paths.
5. Perfect matchings in bipartite graphs: Using neighbour sets.
6. Maximum matchings in bipartite graphs: Minimum vertex covers.
7. Final topic: Stable matchings...

Beyond this course

Topics we didn't cover in discrete structures

- Number theory and cryptographic applications

Beyond this course

Topics we didn't cover in discrete structures

- Number theory and cryptographic applications
- Discrete Probability Theory

Beyond this course

Topics we didn't cover in discrete structures

- Number theory and cryptographic applications
- Discrete Probability Theory
- Symbolic logic and its applications

Beyond this course

Topics we didn't cover in discrete structures

- Number theory and cryptographic applications
- Discrete Probability Theory
- Symbolic logic and its applications
- Finite Automata Theory and transition systems

Beyond this course

Topics we didn't cover in discrete structures

- Number theory and cryptographic applications
- Discrete Probability Theory
- Symbolic logic and its applications
- Finite Automata Theory and transition systems

Many core courses and electives to choose from

- Core: CS 228 (Logic), CS213 (DSA), CS215 (DAI), CS 218 (DAA), CS 310 (Automata), CS317 (Databases), ...

Beyond this course

Topics we didn't cover in discrete structures

- Number theory and cryptographic applications
- Discrete Probability Theory
- Symbolic logic and its applications
- Finite Automata Theory and transition systems

Many core courses and electives to choose from

- Core: CS 228 (Logic), CS213 (DSA), CS215 (DAI), CS 218 (DAA), CS 310 (Automata), CS317 (Databases), ...
- Electives: Crypto, Graph Theory, Game theory, Linear/Convex Optimization, Automated Reasoning, Applied Algo, Geom Algo, Combinatorics, Verification of Programs, Program Analysis, Petri nets, Special topics in Automata and Logics, Quantitative Verification, Learning with Graphs...

Final topic: Stable matchings

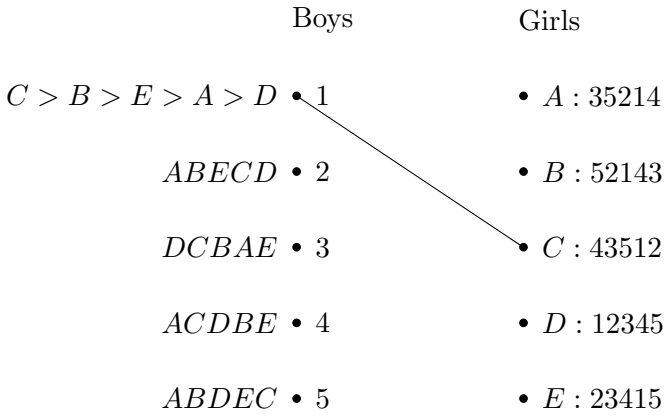
Stable matchings

Boys	Girls
• 1	• A
• 2	• B
• 3	• C
• 4	• D
• 5	• E

Stable matchings

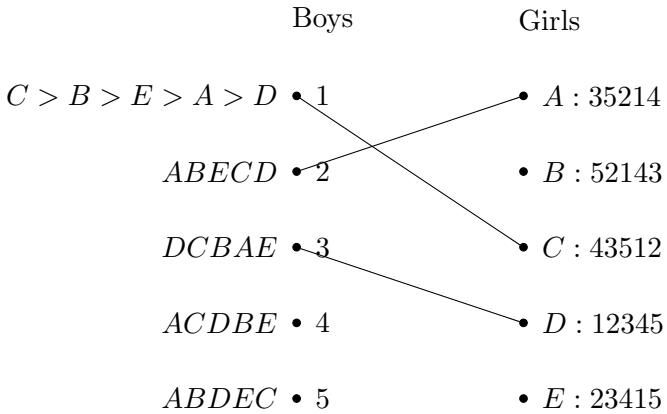
Boys		Girls
$C > B > E > A > D$	• 1	• A : 35214
$ABECD$	• 2	• B : 52143
$DCBAE$	• 3	• C : 43512
$ACDBE$	• 4	• D : 12345
$ABDEC$	• 5	• E : 23415

Stable matchings



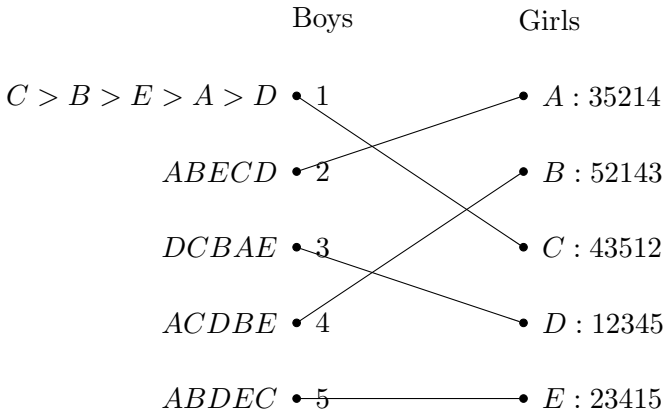
- Let us try a "greedy" marriage strategy for boys.

Stable matchings



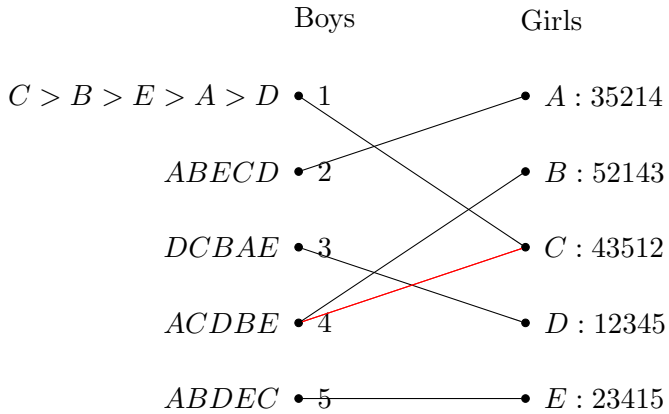
- Let us try a "greedy" marriage strategy for boys.

Stable matchings



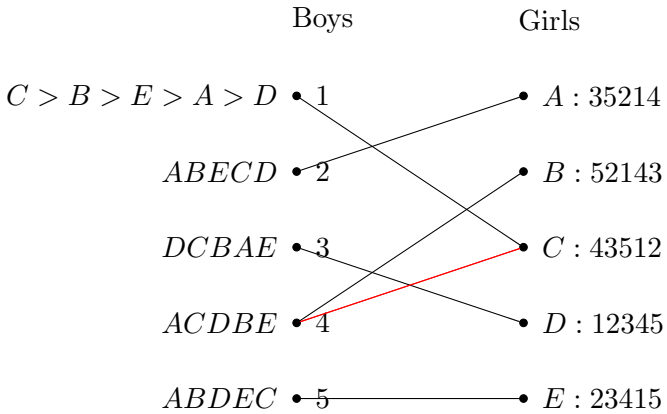
- Let us try a "greedy" marriage strategy for boys.

Stable matchings



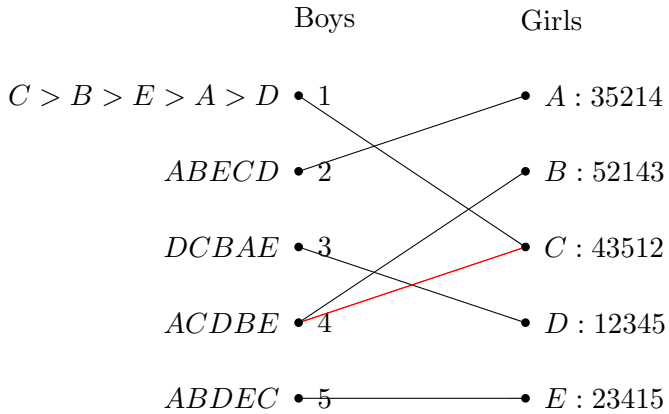
- Let us try a "greedy" marriage strategy for boys.
- Danger!

Stable matchings



- Let us try a "greedy" marriage strategy for boys.
- Danger! 4 prefers C to B and C prefers 4 to 1. Divorce!

Stable matchings



- Let us try a "greedy" marriage strategy for boys.
- Danger! 4 prefers C to B and C prefers 4 to 1. Divorce!
- Qn: Can you match everyone without such Rogue couples?!

More than just a funny puzzle

- College admissions: Original Gale and Shapley paper, 1962. (Shapeley won the Nobel Prize in Econ in 2012 for this, not Turing award!)
- Matching hospitals and residents.
- Matching dancing partners.
- Matching students with jobs.

More than just a funny puzzle

- College admissions: Original Gale and Shapley paper, 1962. (Shapeley won the Nobel Prize in Econ in 2012 for this, not Turing award!)
- Matching hospitals and residents.
- Matching dancing partners.
- Matching students with jobs.
- Matching (PG) TAs with courses.
- JEE algorithm...

Stable matchings

Definition

Given a matching M in a graph with preference lists of nodes.

- Unstable pair: Two vertices x, y such that x prefers y to its assigned vertex and vice versa.

Stable matchings

Definition

Given a matching M in a graph with preference lists of nodes.

- Unstable pair: Two vertices x, y such that x prefers y to its assigned vertex and vice versa.
- x, y would be happier by eloping.

Stable matchings

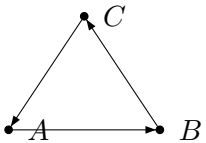
Definition

Given a matching M in a graph with preference lists of nodes.

- Unstable pair: Two vertices x, y such that x prefers y to its assigned vertex and vice versa.
- x, y would be happier by eloping.
- Qn: Find a perfect matching with no unstable pairs. Such a matching is called a Stable Matching.

Roommates Problem

- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$

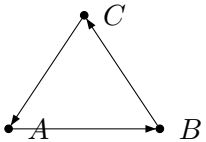


- D

- What can you observe from this?

Roommates Problem

- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$

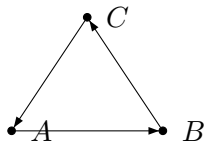


- D

- What can you observe from this?
- Everybody hates D.

Roommates Problem

- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$

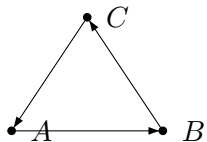


- D

- What can you observe from this?
- Stable matchings don't always exist.

Roommates Problem

- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$



- D

- What can you observe from this?
- Stable matchings don't always exist.
- So, do they exist for bipartite graphs and how can we prove this?

The proposal algorithm

Given: bipartite graph, preference list for n men/women

- 8am: Every man goes to first woman on his list not yet crossed off, and proposes to her!

The proposal algorithm

Given: bipartite graph, preference list for n men/women

- 8am: Every man goes to first woman on his list not yet crossed off, and proposes to her!
- 6pm: Every woman says “maybe” to the man she likes best among the proposals, and says “never” to all others!

The proposal algorithm

Given: bipartite graph, preference list for n men/women

- 8am: Every man goes to first woman on his list not yet crossed off, and proposes to her!
- 6pm: Every woman says “maybe” to the man she likes best among the proposals, and says “never” to all others!
- 10pm: Each rejected suitor crosses off woman from his list.

The proposal algorithm

Given: bipartite graph, preference list for n men/women

- 8am: Every man goes to first woman on his list not yet crossed off, and proposes to her!
- 6pm: Every woman says "maybe" to the man she likes best among the proposals, and says "never" to all others!
- 10pm: Each rejected suitor crosses off woman from his list.

The above loop is repeated every day until there are no more rejected suitors. On that day, the women says "yes" to her "maybe" guy!

The proposal algorithm

Given: bipartite graph, preference list for n men/women

- 8am: Every man goes to first woman on his list not yet crossed off, and proposes to her!
- 6pm: Every woman says "maybe" to the man she likes best among the proposals, and says "never" to all others!
- 10pm: Each rejected suitor crosses off woman from his list.

The above loop is repeated every day until there are no more rejected suitors. On that day, the women says "yes" to her "maybe" guy!

- Does this algorithm terminate?
- If yes, does it produce a stable matching when it terminates?