CS 105: DIC on Discrete Structures

Graph theory

Stable matchings, and the end.

Lecture 35
Nov 09 2023

Topic 3: Graph theory

Topics in Graph theory

1. Basics concepts and definitions.
2. Eulerian graphs: Using degrees of vertices.
3. Bipartite graphs: Using odd length cycles.
4. Connected components: Using cycles.
5. Maximum matchings: Using augmenting paths.
6. Perfect matchings in bipartite graphs: Using neighbour sets. – Hall's theorem
7. Applications of Hall's theorem: Minimum vertex covers – Konig-Egervary's theorem
8. Stable matchings...

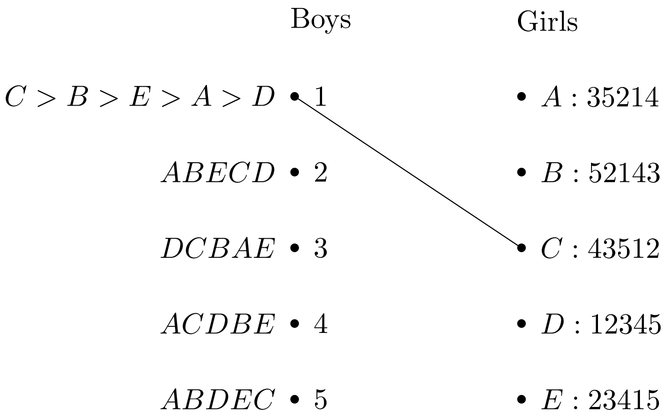
Stable matchings

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

Stable matchings

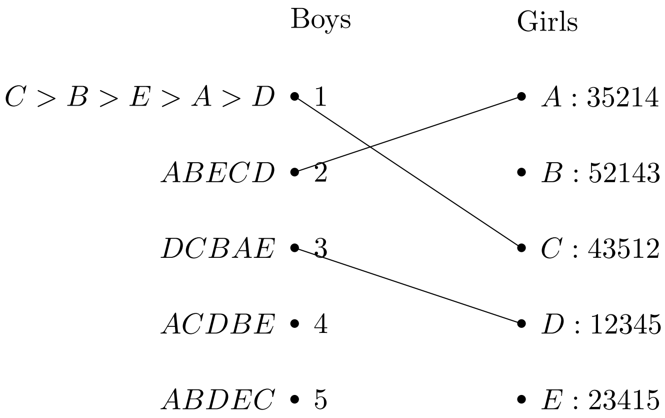
	Boys	Girls
$C > B > E > A > D$	• 1	• A : 35214
$ABECD$	• 2	• B : 52143
$DCBAE$	• 3	• C : 43512
$ACDBE$	• 4	• D : 12345
$ABDEC$	• 5	• E : 23415

Stable matchings



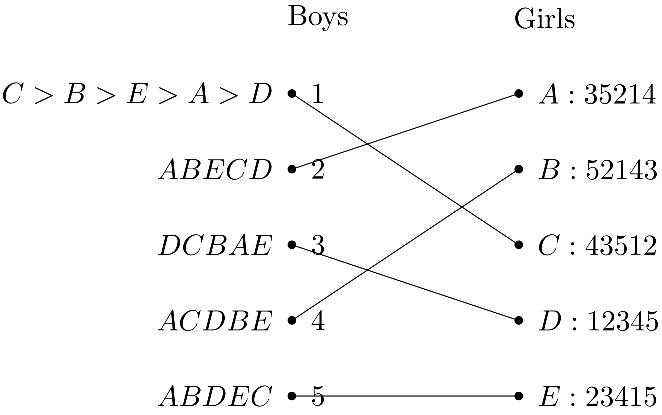
- Let us try a “greedy” marriage strategy for boys.

Stable matchings



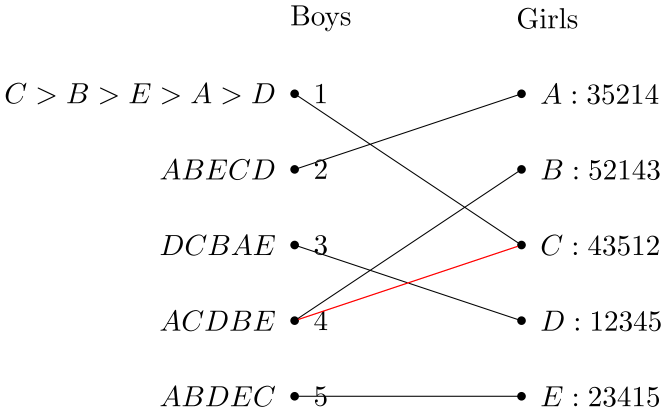
- Let us try a “greedy” marriage strategy for boys.

Stable matchings



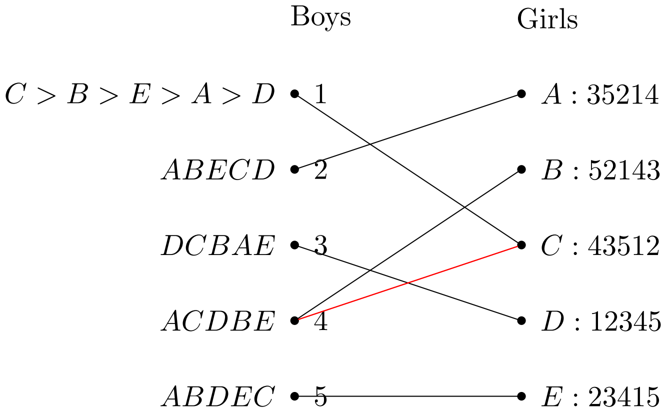
- Let us try a “greedy” marriage strategy for boys.

Stable matchings



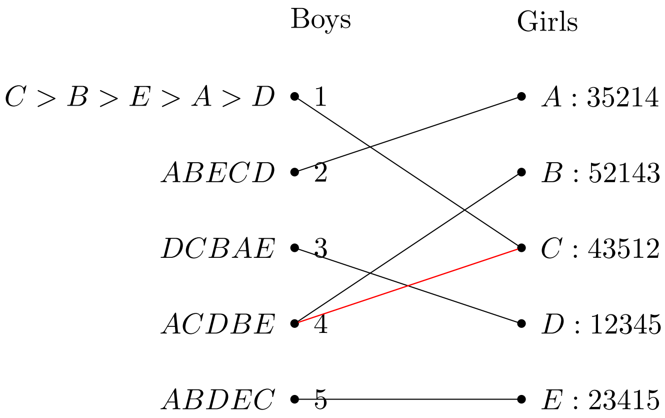
- Let us try a "greedy" marriage strategy for boys.
- Danger!

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.
- 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.
- Matching hospitals and residents.
- Matching dancing partners.
- Matching students with jobs.
- Matching (PG) TAs with courses.
- JEE algorithm...

Stable matchings

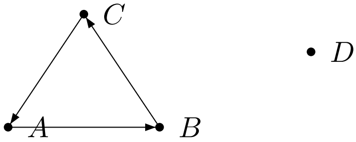
Definition

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.
- 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

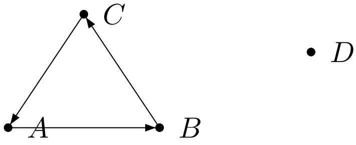
- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$



- What can you observe from this?

Roommates Problem

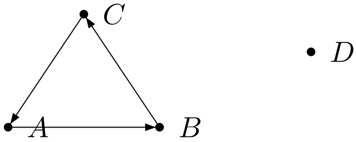
- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$



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

Roommates Problem

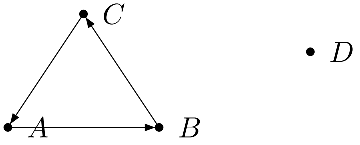
- $A : BCD$
- $B : CAD$
- $C : ABD$
- $D : ABC$



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

Roommates Problem

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



- 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?

Termination and Correctness of the proposal algo

- Try out the algo on the example.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
   - The algo terminates within n^2 days.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
    - The algo terminates within n^2 days.
    - For each day (except last), at least one woman is crossed off some man's list.
    - As there are n men and each has list of size n, algo must terminate in n^2 days.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

- If (M, W) is pair in current matching, s.t., M prefers W'.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

- If (M, W) is pair in current matching, s.t., M prefers W'.
- We will show that W' prefers some other M' and hence no unstable pair.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

- If (M, W) is pair in current matching, s.t., M prefers W'.
- We will show that W' prefers some other M' and hence no unstable pair.
  - Since $W' > W$ for M, he must have proposed to W' before proposing to W.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

- If (M, W) is pair in current matching, s.t., M prefers W'.
- We will show that W' prefers some other M' and hence no unstable pair.
  - Since $W' > W$ for M, he must have proposed to W' before proposing to W.
  - W' rejected him only because she preferred some M'' to M.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

- If (M, W) is pair in current matching, s.t., M prefers W'.
- We will show that W' prefers some other M' and hence no unstable pair.
  - Since $W' > W$ for M, he must have proposed to W' before proposing to W.
  - W' rejected him only because she preferred some M'' to M.
  - By Lemma 2, she likes her final partner at least as much as M'', so better than M.

Termination and Correctness of the proposal algo

Lemmas

1. The algo terminates.
2. If W says maybe to M on k^{th} day, then on every subsequent day she says maybe to someone whom she likes at least as much as M.
3. The algo terminates with a perfect matching.

Theorem

The algorithm produces a stable matching.

- If (M, W) is pair in current matching, s.t., M prefers W'.
- We will show that W' prefers some other M' and hence no unstable pair.
- Thus no man can be part of an unstable pair, implies stable matching. □

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?
- Can men do any better?

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?
- Can men do any better?
- Can the women do any better?

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?
- Can men do any better?
- Can the women do any better?
- Define an optimal woman for a man as the best he can get under any stable matching. A matching is male-optimal if every man is matched to his optimal woman.

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?
- Can men do any better?
- Can the women do any better?
- Define an optimal woman for a man as the best he can get under any stable matching. A matching is male-optimal if every man is matched to his optimal woman.
- Define an optimal man for a woman as the best she can get under any stable matching.

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?
- Can men do any better?
- Can the women do any better?
- Define an optimal woman for a man as the best he can get under any stable matching. A matching is male-optimal if every man is matched to his optimal woman.
- Define an optimal man for a woman as the best she can get under any stable matching.

Theorem (H.W.: Prove this!)

The male-proposal algorithm is male-optimal (and "woman-pessimal").

The proposal algorithm: who does better?

Features of this proposal algorithm

- Who is happier with the result? Men or women?
- Can men do any better?
- Can the women do any better?
- Define an optimal woman for a man as the best he can get under any stable matching. A matching is male-optimal if every man is matched to his optimal woman.
- Define an optimal man for a woman as the best she can get under any stable matching.

Theorem (H.W.: Prove this!)

The male-proposal algorithm is male-optimal (and "woman-pessimal").

Conclusion: Propose first!

Further reading

- Many questions, rich theory.
- How many stable marriages are possible?
- Can you do better by lying? Boys - no!, Girls - yes!
- What if there are brother-sisters (who should not be matched!)?

Further reading

- Many questions, rich theory.
- How many stable marriages are possible?
- Can you do better by lying? Boys - no!, Girls - yes!
- What if there are brother-sisters (who should not be matched!)?
- D. Gale and L.S. Shapley, College Admissions and the Stability of Marriage, American Mathematical Monthly 69(1962), pp. 9-14.
- D. Gusfield and R.W. Irving, The Stable Marriage Problem: Structure and Algorithms, MIT Press, 1989.

The 2012 Nobel prize in Economics to Shapley and Roth: "for the theory of stable allocations and the practice of market design".

Summary

What we covered in this course

1. Mathematical proofs and basic structures
2. Counting and combinatorics
3. Introduction to graph theory

Summary: Part 1

- Mathematical proofs and reasoning
- 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
- 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!
- 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!
- 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!
- 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: Part 2

- Counting and Combinatorics

Summary: Part 2

- 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: Part 2

- 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: Part 3

Topics in Graph theory

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

Summary: Part 3

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. – Hall's theorem
6. Maximum matchings in bipartite graphs: Minimum vertex covers. – Konig-Egervary's theorem
7. Stable matchings... and the Gale Shapley Algo

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