CS 3330 Algorithms

1. What is going to happen?

- *course policy*
- *what is an algorithm?*
- *what is algorithm analysis?*
- *what is going on in the course?*
- *five representative problems*
- *schedule*

CS 3330 Algorithms

1. What is going to happen?

- *course policy*
- *what is an algorithm?*
- *what is algorithm analysis?*
- *what is going on in the course?*
- *five representative problems*
- *schedule*

Course policy

Course meeting time and place:

- Section 1 (CS 3330: 0001): Mehrdad Moharrami

  Tuesdays, Thursdays, 8:00 – 9:15 am, 112 MacBride Hall

- Section 2 (CS 3330: 0002): Bijaya Adhikari

  Tuesdays, Thursdays, 3:30 – 4:45 pm, 118 MacLean Hall

Instructor Contact Information

- Section 1 (CS 3330: 0001): Mehrdad Moharrami (moharami@uiowa.edu)

  Office location: 257 MacBride Hall

  Office hours: Tuesdays and Thursdays 3:00 pm – 4:00 pm

- Section 2 (CS 3330: 0002): Bijaya Adhikari (bijaya-adhikari@uiowa.edu)

  Office location: 256 MacBride Hall

  Office hours: Tuesdays and Thursdays 2:00 pm – 3:00 pm

Course policy

TA Information

- Erfan Mirzaei:
  - Office hours: Mondays (1:30pm-2:30pm), Wednesdays (10:30pm-11:30 pm), Fridays (2:30pm-3:30pm)
  - Office location: 201N MLH MacLean Hall
  - Email: erfan-mirzaei@uiowa.edu
- Max Johnson:
  - Office hours: Mondays (10:15am-11:15am), Wednesdays (1:30pm-2:30pm), Fridays (10:15am-11:15am)
  - Office location: 201N MacLean Hall
  - Email: max-johnson@uiowa.edu

Course policy

Textbook and Materials

- Algorithm Design, by Kleinberg and Tardos
- Slides will be shared on ICON

Course policy

Grading Policy

- Homeworks 30%:
  - There will be total of 10 homeworks, some will involve programming
  - Total of 10 late days for submission, you can use up to 3 days for any homework.
  - Solution to each homework will be posted 3 days after the deadline after which no late submission will be accepted.
  - We will drop your lowest homework score.
- Midterm Exam 35%:
  - Monday Oct 15th , 6:30 to 8:30, 100 Philips hall
- Final Exam 35%:
  - Final is cumulative with a focus on topic after the midterm
  - TBD

CS 3330 Algorithms

1. What is going to happen?

- course policy
- what is an algorithm?
- what is algorithm analysis?
- what is going on in the course?
- five representative problems
- schedule

What is an algorithm?

An algorithm is an step-by-step recipe designed to achieve a specific task or goal.

- The cooking recipe
- Ants path finding
- The world of artificial intelligence
- Almost anything can be thought of as an algorithm

What is an algorithm?

Algorithm's structure

Goal: design an algorithm that perform the task at hand efficiently:

- it should use reasonable amount of resources, given the size of input;
- it should run in a reasonable amount of time, given the size of input;
- It should give a reasonable output.

Imprecise, isn't it? Let's bring more math to the picture.

CS 3330 Algorithms

1. What is going to happen?

- course policy
- what is an algorithm?
- what is algorithm analysis?
- what is going on in the course?
- five representative problems
- schedule

What is algorithm analysis?

How do we check the correctness of an algorithm for certain task?

- Try proving it using math or disproving it using a counter example.

What is algorithm analysis?

How do we check the correctness of an algorithm for certain task?

- Try proving it using math or disproving it using a counter example.

Is it possible to solve all problems using efficient algorithms?

- For some problems, there is no efficient algorithm, period. How should we find them? Is it possible to classify problems to different categories?

What is algorithm analysis?

How do we check the correctness of an algorithm for certain task?

- Try proving it using math or disproving it using a counter example.

Is it possible to solve all problems using efficient algorithms?

- For some problems, there is no efficient algorithm, period. How should we find them? Is it possible to classify problems to different categories?

How do we measure the performance of an algorithm?

- how much resources (RAM) do we need to implement the algorithm in machine
- how long we need to run the machine for certain size of inputs
- how close is the output of the algorithm to the desired solution

What is algorithm analysis?

How do we check the correctness of an algorithm for certain task?

- Try proving it using math or disproving it using a counter example.

Is it possible to solve all problems using efficient algorithms?

- For some problems, there is no efficient algorithm, period. How should we find them? Is it possible to classify problems to different categories?

How do we measure the performance of an algorithm?

- how much resources (RAM) do we need to implement the algorithm in machine
- how long we need to run the machine for certain size of inputs
- how close is the output of the algorithm to the desired solution

These are all mathematical questions that can be characterized without any implementation!

CS 3330 Algorithms

1. What is going to happen?

- *course policy*
- *what is an algorithm?*
- *what is algorithm analysis?*
- *what is going on in the course?*
- *five representative problems*
- *schedule*

What is going on in the course?

We will cover the following concepts throughout the semester:

- The stable marriage problem
- Running time analysis of algorithms
- Graphs and graph algorithms
- Greedy algorithms
- Divide-and-Conquer paradigm
- Dynamic programming
- Computational intractability

These are pretty much the opening 8 chapters of the book, except for Chapter 7.

My goal is for you to have a strong grasp of thinking in terms of algorithms!

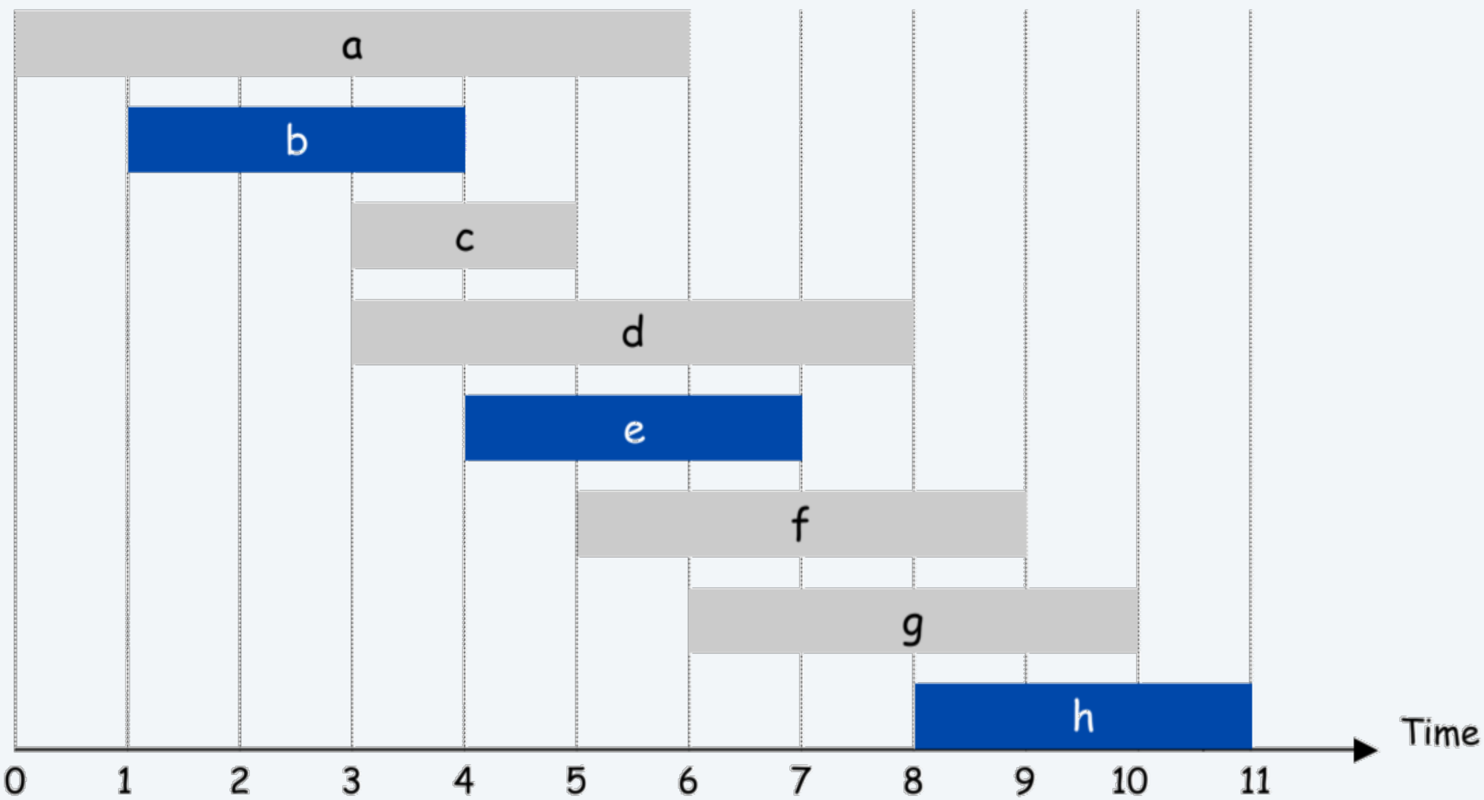
CS 3330 Algorithms

1. What is going to happen?

- *course policy*
- *what is an algorithm?*
- *what is algorithm analysis?*
- *what is going on in the course?*
- *five representative problems*
- *schedule*

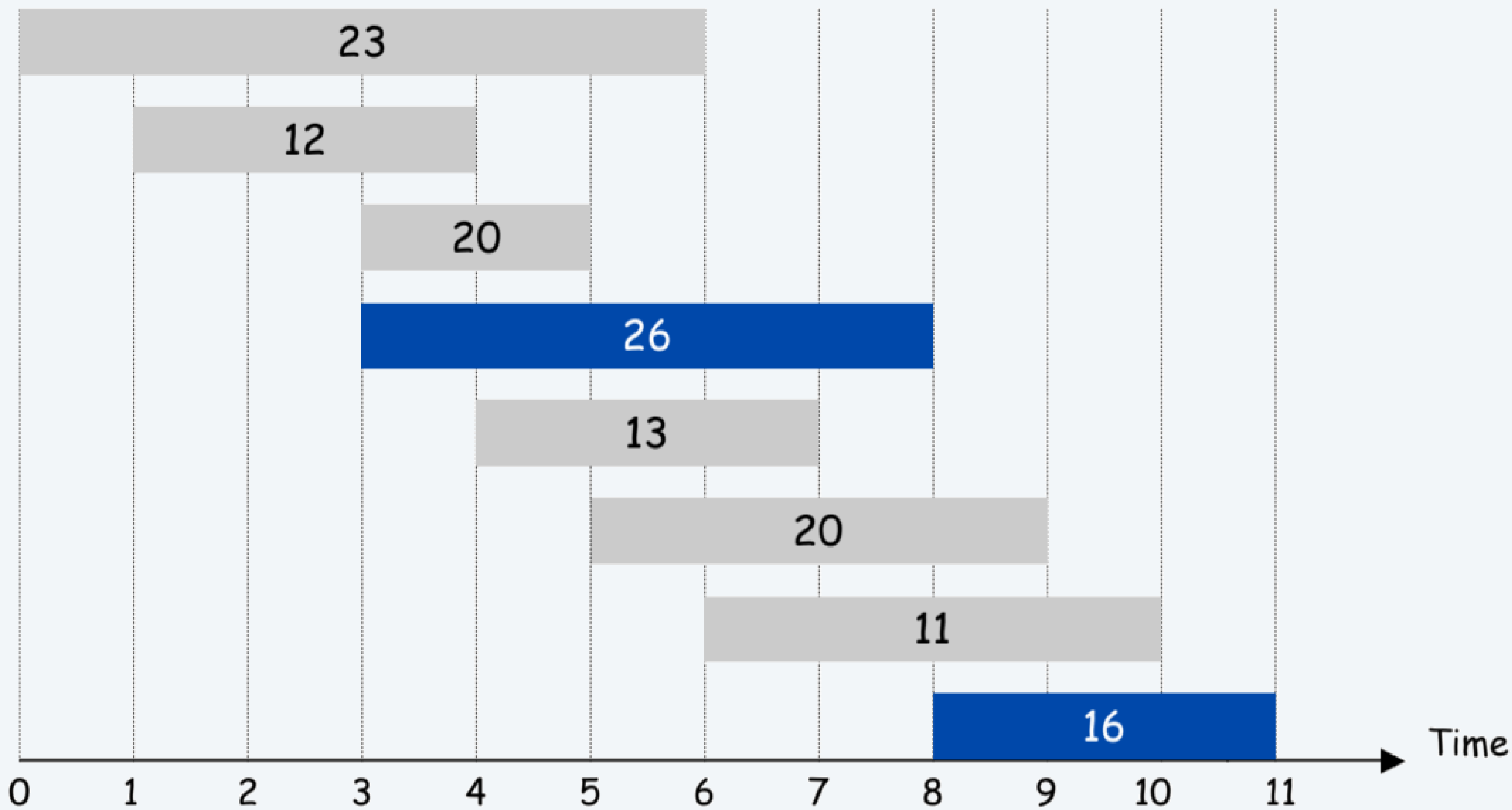
Five representative problems: Interval Scheduling

Input. Set of jobs with start times and finish times.
Goal. Find maximum cardinality subset of mutually compatible jobs.
Difficulty. Easy.



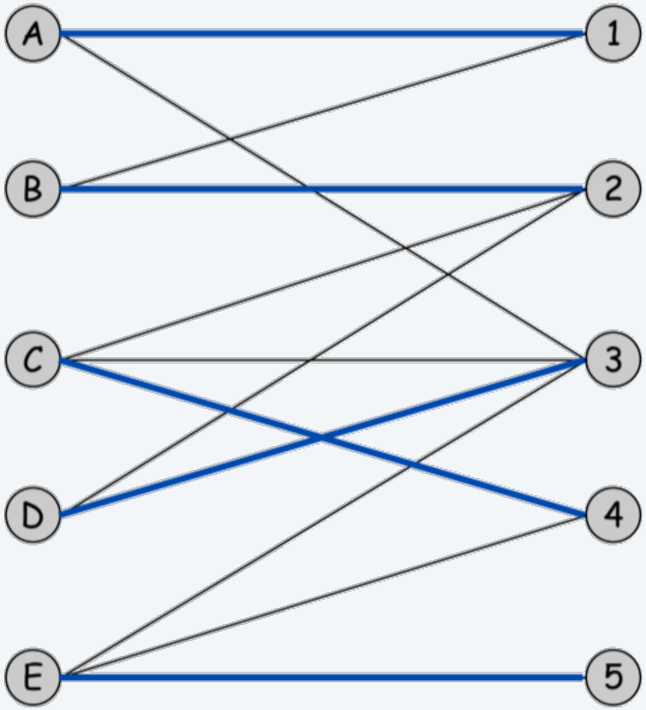
Five representative problems: Weighted Interval Scheduling

Input. Set of jobs with start times, finish times, and weights.
Goal. Find maximum weight subset of mutually compatible jobs.
Difficulty. Easy but more complicated.



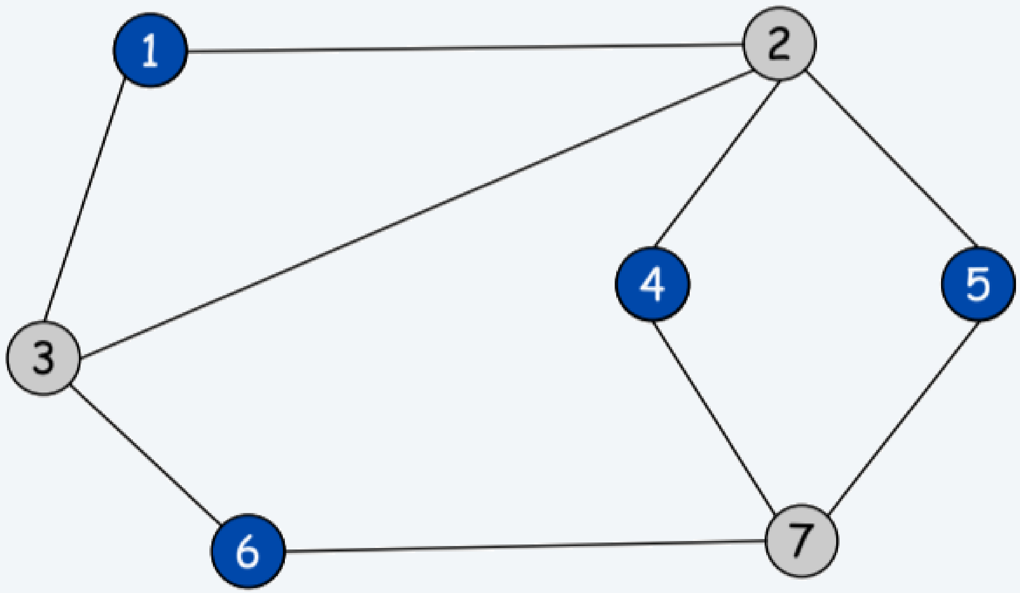
Five representative problems: Bipartite Matching

Input. Bipartite graph.
Goal. Find maximum cardinality matching.
Difficulty. Easy but computationally harder.



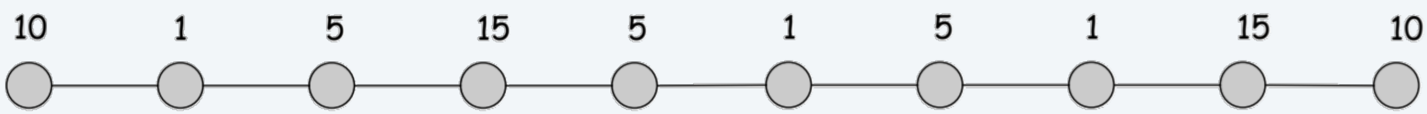
Five representative problems: Independent Set

Input. Graph.
Goal. Find maximum cardinality independent set.
Difficulty. Hard but easy to verify solution.



Five representative problems: Competitive Facility Location

Input. Graph with weight on each node.
Game. Two competing players alternate in selecting nodes. Not allowed to select a node if any of its neighbors have been selected.
Goal. Select a maximum weight subset of nodes.
Difficulty. Hard to even verify solution.



Second player can guarantee 20, but not 25.

CS 3330 Algorithms

1. What is going to happen?

- *course policy*
- *what is an algorithm?*
- *what is algorithm analysis?*
- *what is going on in the course?*
- *five representative problems*
- *schedule*

Schedule

Week 1	8/27/2024	Introduction + Course policy	
	8/29/2024	Stable Matching	HW1 released
Week 2	9/3/2024	Running time Analysis	
	9/5/2024		HW2 released, HW1 due
Week 3	9/10/2024	Graphs	
	9/12/2024		HW3 released, HW2 due
Week 4	9/17/2024		
	9/19/2024	Coin changing and Interval Scheduling	HW4 released. HW 3 due
Week 5	9/24/2024	Interval Partioning and Interval Scheduling	
	9/26/2024	Minimizing Lateness and Google Foo.bar	HW 4 due
Week 6	10/1/2024		
	10/3/2024	Minimum Spanning Trees	HW5 released
Week 7	10/8/2024		
	10/10/2024	Dijkstra's Algorithm	HW5 due
Week 8	10/15/2024	Review	Midterm Exam 1
	10/17/2024	Merge Sort and Counting Inversion	HW 6 released
Week 9	10/22/2024	Randomized QuickSort	
	10/24/2024	Median and Selection	HW 7 released, HW 6 due
Week 10	10/29/2024	Master Theorem and Integer	
	10/31/2024	Strassen's Algorithms	HW 8 released, HW 7 due
Week 11	11/5/2024	Weighted Interval Schedule	
	11/7/2024	Segemented Least Squares and Knapsack	HW 9 released, HW 8 due
Week 12	11/12/2024		
	11/14/2024	Sequence Alignment	HW 9 due
Week 13	11/19/2024	Hirschberg	
	11/21/2024	Bellman–Ford–Moore algorithm	Homework 10 released
Week 14	11/26/2024	Thanksgiving Break	
	11/28/2024		
Week 15	12/3/2024	NP completeness	
	12/5/2024		Homework 10 due
Week 16	12/10/2024	Review	
	12/12/2024		
			Final Exam