While waiting, in case you have your laptop/cell-phone with you:

- go to https://www.iclicker.com/students and register/log-in as a student
- find “Online Algorithms” course at “University of Manitoba”, and add it to your courses!
- join the class at the beginning of the class

Picture is from the cover of the book by Borodin and El-Yaniv. See Slide 6.
Introduction
In a Glance . . .

Online algorithms are

- Practical
- Diverse
- Fun (really!)
Online algorithms are

- Practical
- Diverse
- Fun (really!)

Let’s ‘play’ with online algorithms and enjoy
Instructor

Shahin Kamali

joined UM in July 2017, before that was a postdoc at MIT (2015-2017), and before that did a PhD at U. Waterloo (2008-2014)

has a broad research interest, including topics related to graph algorithms, online algorithms, block-chain technology, performance engineering, etc.
Introduction

Instructor

Shahin Kamali

joined UM in July 2017, before that was a postdoc at MIT (2015-2017), and before that did a PhD at U. Waterloo (2008-2014)
has a broad research interest, including topics related to graph algorithms, online algorithms, block-chain technology, performance engineering, etc.

Your turn to . . .

Introduce yourself.
Your research group, research interests.
Formalities

see http://www.cs.umanitoba.ca/~kamalis/fall19/info.pdf for all details.
Formalities

Logistics

- Lecture: Tuesdays and Thursdays, 10:00-11:15am
  EITC E2 Room: 164
Lecture: Tuesdays and Thursdays, 10:00-11:15am
EITC E2 Room: 164

Webpage:
http://www.cs.umanitoba.ca/kamalis/fall19/comp4060-7720.html
Formalities

Logistics

- Lecture: Tuesdays and Thursdays, 10:00-11:15am
  EITC E2 Room: 164

- Webpage:
  http://www.cs.umanitoba.ca/~kamalis/fall19/comp4060-7720.html
Formalities

Logistics

- Lecture: Tuesdays and Thursdays, 10:00-11:15am
  EITC E2 Room: 164

- Webpage:
  http://www.cs.umanitoba.ca/ kamalis/fall19/comp4060-7720.html

- Piazza: https://piazza.com/umanitoba.ca/fall2019/comp7720
Formalities

Logistics

- Lecture: Tuesdays and Thursdays, 10:00-11:15am
  EITC E2 Room: 164

- Webpage:
  http://www.cs.umanitoba.ca/ kamalis/fall19/comp4060-7720.html

- Piazza: https://piazza.com/umanitoba.ca/fall2019/comp7720

- Office hours: 11:30am-12:30pm, Mondays and 2:00pm-3:00pm
  Tuesdays, in E2 586 or by appointment
Formalities

**Textbook**

- A list of required reading will be provided on the course webpage.
- No book is required to be purchased.
- The following book is suggested as a reference:
  - Borodin and El-Yaniv, Online Computation and Competitive Analysis (2005)
Formalities

Grading

There will be:

- Four assignments
- Two exams
- A course project
- Class participation
Formalities

Grading

There will be:

- Four assignments
- Two exams
- A course project
- Class participation

Theorem

The focus of this course is on learning, discovering, and research.
Formalities

Grading

There will be:

- Four assignments
- Two exams
- A course project
- Class participation

Theorem

The focus of this course is on learning, discovering, and research.

Corollary

Don’t worry too much about your final mark (but worry a little bit).
Formalities

Grading (cntd.)

Four assignments:

- 30 percent of the final mark
- there will be extra marks for bonus questions.
- submit only pdf files (preferably use \LaTeX)
- we will use Crowdmark
Four assignments:

- 30 percent of the final mark
- there will be extra marks for bonus questions.
- submit only pdf files (preferably use \LaTeX)
- we will use Crowdmark
- An additional assignment, Assignment 0, will be posted shortly.
  - It gives you a chance to assess your background and learn to work with \LaTeX, Piazza, and Crowdmark.
  - It gives you a chance to drop the course before it is too late if you lack the background.
Formalities

Grading (cntd.)

Four assignments:

- 30 percent of the final mark
- there will be extra marks for bonus questions.
- submit only pdf files (preferably use \LaTeX)
- we will use Crowdmark
- An additional assignment, Assignment 0, will be posted shortly.
  - It gives you a chance to assess your background and learn to work with \LaTeX, Piazza, and Crowdmark.
  - It gives you a chance to drop the course before it is too late if you lack the background.

Exams:

- 30 percent of the final mark (15 percent each)
- Sample exams will be provided for practice
Formalities

Projects

- Course project
  - 30 percent of the final mark
  - Extra marks for outstanding projects (publishable projects)
  - Work individually or in groups of two

- Projects involve:
  - Proposal
  - Presentation
  - Final report (in form of a research paper)
Projects (cntd.)

- Project topics will be suggested in the first few weeks of the class
  - You can chose your own topic based on your research
  - Come to office hours to talk about it!
Projects (cntd.)

- Project topics will be suggested in the first few weeks of the class
  - You can choose your own topic based on your research
  - Come to office hours to talk about it!

- Project categories:
  - Exploring possible solutions to an open problem
  - Writing a survey paper on a current topic related to online algorithms
  - Writing code to implement and compare the performance of online algorithms for a problem
Class participation:

- 10 percent of the final mark
- Don’t be shy; ask questions, answer my questions, seat in the frontline!
- We use iClicker: bring a laptop or a smart-phone to the class
  - Register as a student on https://www.iclicker.com/students
  - If you have a laptop/phone with you, take the following steps (otherwise, bring one for the next class):
    - Look for “Online Algorithms” under “University of Manitoba”.
    - Join the class!
Answer the following question using iClicker app/website.
There is not necessarily a single correct answer; your responses are not “marked”.

**Quiz**

*Indicate why did you take the online algorithm course:*

(a) I like Internet and programming on web

(b) My supervisor requested/forced me to take this course.

(c) I just need it to complete my program’s course requirements, and there are not many options.

(d) I love theoretical computer science.
Formalities

Important Dates (tentative)

September 5: the first class
September 18: assignment 1 due
September 25: project proposal due
October 4: assignment 2 due
October 16: assignment 3 due
October 18: chocolate cupcake day
October 24: exam 1 (in class)

November 12-15: fall break
November 18: VW deadline
November 21: assignment 4 due
November 25-28: project presentation dates
December 5: exam 2 (in class)
December 13: project final report due
Online Algorithms
Traditional algorithms are ‘offline’ in the sense that they have the whole input in their hand.

Online algorithms, in contrast, do not have/need the whole input in order to solve a problem

- The input is a ‘sequence’ which is processed by the online algorithm piece-by-piece
- The online algorithms often take irrevocable decisions to process the input.
The input is a set/sequence of items of various sizes

E.g., 9, 3, 8, 5, 1, 1, 3, 2, 4, 2, 4, 5, 5, 8, 6, 4, 5, ...
Bin Packing Problem

- The input is a set/sequence of items of various sizes
  - E.g., <9, 3, 8, 5, 1, 1, 3, 2, 4, 2, 4, 5, 5, 8, 6, 4, 5, ...>.

- The goal is to pack these items into a minimum number of bins of uniform capacity.
In the online setting:

- an algorithm receives items one by one
- when it receives an item, it has to place it in a bin without any knowledge about forthcoming items
- decisions of the algorithms are irrevocable (i.e., cannot move items between bins)
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

\[< 9 \ 3 \ 8 \ 5 \ 1 \ 1 \ 3 \ 2 \ 4 \ 2 \ 4 \ 5 \ 5 \ 8 \ 6 \ 4 \ 5 >\]
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
Online Algorithms

**First Fit (FF) Algorithm**

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

\[<9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5>\]
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

\[
< 9 \ 3 \ 8 \ 5 \ 1 \ 1 \ 3 \ 2 \ 4 \ 2 \ 4 \ 5 \ 5 \ 8 \ 6 \ 4 \ 5 >
\]
Online Algorithms

First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >

![Diagram showing the First Fit algorithm applied to a sequence of numbers]
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

\[<9\ 3\ 8\ 5\ 1\ 1\ 3\ 2\ 4\ 2\ 4\ 5\ 5\ 8\ 6\ 4\ 5>\]
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >

```
1

9

5

3

1

2

8

4

3
```
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
Online Algorithms

First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
Online Algorithms

First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

\[< 9 \ 3 \ 8 \ 5 \ 1 \ 1 \ 3 \ 2 \ 4 \ 2 \ 4 \ 5 \ 5 \ 8 \ 6 \ 4 \ 5 >\]
Online Algorithms

First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there.
- Open a new bin if such bin does not exist.

\[
< 9 \ 3 \ 8 \ 5 \ 1 \ 1 \ 3 \ 2 \ 4 \ 2 \ 4 \ 5 \ 5 \ 8 \ 6 \ 4 \ 5 >
\]
First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
Online Algorithms

First Fit (FF) Algorithm

- Find the first bin which has enough space for the item, and place the item there
- Open a new bin if such bin does not exist

< 9 3 8 5 1 1 3 2 4 2 4 5 5 8 6 4 5 >
Online Algorithms

Competitive Ratio

We use the framework of competitive analysis to compare online algorithms.
We use the framework of **competitive analysis** to compare online algorithms.

Let $\text{OPT}$ denote the best possible offline solution.

Given a sequence $\sigma$, $\text{OPT}$ is an algorithm which packs items in $\sigma$ in a minimum number of bins.
We use the framework of competitive analysis to compare online algorithms.

Let $\text{OPT}$ denote the best possible offline solution.

- Given a sequence $\sigma$, $\text{OPT}$ is an algorithm which packs items in $\sigma$ in a minimum number of bins.

- Competitive ratio of an algorithm $A$ is the maximum ratio between the cost of $A$ and that of $\text{OPT}$ over all sequences:

$$cr(A) \equiv \max_{\sigma} \frac{\text{cost}_A(\sigma)}{\text{cost}_{\text{OPT}}(\sigma)}$$
For First Fit, the competitive ratio is 1.7 [Johnson 1973]
For First Fit, the competitive ratio is 1.7 [Johnson 1973]

- The number of bins opened by FF for any sequence is at most 1.7 times that of $\text{OPT}$, i.e., $c.r. \leq 1.7$ (upper bound for FF)
- There are sequences for which the number of bins opened by FF is 1.7 times that of $\text{OPT}$, i.e., $c.r. \geq 1.7$ (lower bound for FF)
For First Fit, the competitive ratio is $1.7$ [Johnson 1973]

- The number of bins opened by FF for any sequence is at most $1.7$ times that of $\text{OPT}$, i.e., $c.r. \leq 1.7$ (upper bound for FF)
- There are sequences for which the number of bins opened by FF is $1.7$ times that of $\text{OPT}$, i.e., $c.r. \geq 1.7$ (lower bound for FF)

The best existing online algorithm has c.r. of $1.5783$ [Balogh et al. 2017]
Online Algorithms

Competitive Ratio of First Fit

For First Fit, the competitive ratio is 1.7 [Johnson 1973]

- The number of bins opened by FF for any sequence is at most 1.7 times that of Opt, i.e., c.r. ≤ 1.7 (upper bound for FF)
- There are sequences for which the number of bins opened by FF is 1.7 times that of Opt, i.e., c.r. ≥ 1.7 (lower bound for FF)

The best existing online algorithm has c.r. of 1.5783 [Balogh et al. 2017]

No algorithm can be better than 1.54037-competitive (best general lower bound) [Balogh et al. 2015].
Ski-rental problem

- Assume you want to go skiing for \( x \) number of days
  - In the online setting, the value of \( x \) is unknown!
Assume you want to go skiing for $x$ number of days

- In the online setting, the value of $x$ is unknown!

You can buy the equipment for a one-time cost of $b$ or rent each day for a cost of 1 per day
Online Algorithms

Ski-rental problem

Assume you want to go skiing for $x$ number of days
  - In the online setting, the value of $x$ is unknown!

You can buy the equipment for a one-time cost of $b$ or rent each day for a cost of 1 per day

If we know $x$, what is the best solution?
Ski-rental problem

- Assume you want to go skiing for $x$ number of days
  - In the online setting, the value of $x$ is unknown!
- You can buy the equipment for a one-time cost of $b$ or rent each day for a cost of 1 per day
- If we know $x$, what is the best solution?
  - Buy at the beginning if $x \geq b$, otherwise, rent every day
Online strategy **break-even**: rent for the first $b - 1$ days and buy in the next day.
Online strategy **break-even**: rent for the first $b - 1$ days and buy in the next day.

What is the competitive ratio of Break-even algorithm?
Online strategy **break-even**: rent for the first $b - 1$ days and buy in the next day.

What is the competitive ratio of Break-even algorithm?

**Theorem**

*Competitive ratio is roughly 2, and it is the best for any deterministic online algorithm.*
Syllabus
Doubling technique

In many occasions, a ‘doubling technique’
can be used to design and analyze online
algorithms

- The lost cow problem
- Online bidding
Potential function technique is a classic approach for analysis of online problems

- The paging problem, Sleator-Tarjan proof, randomized paging
Data structures

- Self-adjusting data structures
  - list update problem
  - data compression
  - self-adjusting binary trees, and dynamic optimality conjecture
Packing problems

- Weighting technique
  - Bin packing
  - Renting servers in the cloud
  - Online scheduling
Algorithm design as a “game” between an online algorithm and an adversary.

How you can “train” an algorithm based on an input data that is being evolved?

“Combining experts advice” problem
Graph problems

- Graph problems
  - $k$-server problem
  - Graph coloring
  - Bipartite matching
Syllabus

Computation geometry

- Robot searching
- 2-dimensional bin packing
Syllabus

Advice complexity of online problems

- Online algorithms with Advice: what if we have partial information about future?
  - Algorithms with advice for paging, bin packing, list update

- Many project ideas here!
Syllabus

Alternative analysis techniques

- Competitive ratio is a **worst-case measure**
- Alternative analysis techniques are used to compare algorithms based on their typical behaviour
  - Bijective analysis
  - Relative worst-order analysis