Algorithms and Data Structures for Big Data

Hu Fu @ SUFE. Sept 5, 2023

Teaching Staff

- Instructor: Hu Fu 伏虎
- Office: 504 School of Information Management & Engineering
- Email: fuhu@mail.shufe.edu.cn
- Website: http://www.fuhuthu.com/BigData2023/

What is Big Data?

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Retail and wholesale trade

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Retail and wholesale trade
  - Banking and securities

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Retail and wholesale trade
  - Banking and securities
  - Communications, media and entertainment

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Retail and wholesale trade
  - Banking and securities
  - Communications, media and entertainment
  - Healthcare

What is Big Data?

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Insurance

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Insurance
  - Government

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Insurance
  - Government
  - Scientific research

What is Big Data?

- Broadly speaking, big data is simply exceptionally large datasets
  - In popular media, it is sometimes synonym of machine learning with large training datasets
- Applications include:
  - Insurance
  - Government
  - Scientific research
  - Transportation…

Focus of this course

Focus of this course

- This course focuses on basic operations on such datasets, such as

Focus of this course

- This course focuses on basic operations on such datasets, such as
  - Accessing and storing such datasets

Focus of this course

- This course focuses on basic operations on such datasets, such as
  - Accessing and storing such datasets
  - Estimating simple statistics

Focus of this course

- This course focuses on basic operations on such datasets, such as
  - Accessing and storing such datasets
  - Estimating simple statistics
  - Extracting meaningful sketches to be used by upper level applications

Focus of this course

- This course focuses on basic operations on such datasets, such as
  - Accessing and storing such datasets
  - Estimating simple statistics
  - Extracting meaningful sketches to be used by upper level applications
- We do not look at upper level applications such as learning

Focus of this course

- This course focuses on basic operations on such datasets, such as
  - Accessing and storing such datasets
  - Estimating simple statistics
  - Extracting meaningful sketches to be used by upper level applications
- We do not look at upper level applications such as learning
  - For that you should take machine learning or statistical learning theory (the latter not offered this year)

(Tentative) Syllabus

(Tentative) Syllabus

- Review of basic probability theory

(Tentative) Syllabus

- Review of basic probability theory
- Hashing

(Tentative) Syllabus

- Review of basic probability theory
- Hashing
- Concentration inequalities

(Tentative) Syllabus

- Review of basic probability theory
- Hashing
- Concentration inequalities
- Some randomized data structures

(Tentative) Syllabus

- Review of basic probability theory
- Hashing
- Concentration inequalities
- Some randomized data structures
- Dimensionality reductions

(Tentative) Syllabus

- Review of basic probability theory
- Hashing
- Concentration inequalities
- Some randomized data structures
- Dimensionality reductions
- Streaming algorithms

(Tentative) Syllabus

- Review of basic probability theory
- Hashing
- Concentration inequalities
- Some randomized data structures
- Dimensionality reductions
- Streaming algorithms
- (Maybe) Compressed sensing

Coursework

Coursework

- Homework:

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in
  - Typesetting your solutions is highly encouraged

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in
  - Typesetting your solutions is highly encouraged
- Project: literature survey on a chosen topic (I can suggest candidate topics)

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in
  - Typesetting your solutions is highly encouraged
- Project: literature survey on a chosen topic (I can suggest candidate topics)
  - Done in groups of up to 4 people

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in
  - Typesetting your solutions is highly encouraged
- Project: literature survey on a chosen topic (I can suggest candidate topics)
  - Done in groups of up to 4 people
  - Presentation at the end of the semester

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in
  - Typesetting your solutions is highly encouraged
- Project: literature survey on a chosen topic (I can suggest candidate topics)
  - Done in groups of up to 4 people
  - Presentation at the end of the semester
- Take-home final: 1-3 days' work, done independently. Time TBD

Coursework

- Homework:
  - Students are encouraged to work in groups of up to 3 people
  - Everyone should be able to explain the solutions turned in
  - Typesetting your solutions is highly encouraged
- Project: literature survey on a chosen topic (I can suggest candidate topics)
  - Done in groups of up to 4 people
  - Presentation at the end of the semester
- Take-home final: 1-3 days' work, done independently. Time TBD
- Grade makeup: 40% homework + 20% project + 40% final

Prerequisites

Prerequisites

- We will assume basic familiarity of data structures and algorithms

Prerequisites

- We will assume basic familiarity of data structures and algorithms
  - At the very least, you should have some rough idea on how computer programs work

Prerequisites

- We will assume basic familiarity of data structures and algorithms
  - At the very least, you should have some rough idea on how computer programs work
  - Comfort with running time analysis (e.g. familiarity with the big $O(\cdot)$ notation and worst case analysis)

Prerequisites

- We will assume basic familiarity of data structures and algorithms
  - At the very least, you should have some rough idea on how computer programs work
  - Comfort with running time analysis (e.g. familiarity with the big $O(\cdot)$ notation and worst case analysis)
  - Knowledge of basic data structures. We will use arrays, linked lists, trees.

Prerequisites

- We will assume basic familiarity of data structures and algorithms
  - At the very least, you should have some rough idea on how computer programs work
  - Comfort with running time analysis (e.g. familiarity with the big $O(\cdot)$ notation and worst case analysis)
  - Knowledge of basic data structures. We will use arrays, linked lists, trees.
  - Comfort with basic probability theory will go a long way, but is not strictly required. We start with a quick review.

This is a *Theory* course

This is a *Theory* course

- All materials are proof-based, and so is the homework

This is a *Theory* course

- All materials are proof-based, and so is the homework
- Implementation of algorithms is not required; coding things up may help with understanding

This is a *Theory* course

- All materials are proof-based, and so is the homework
- Implementation of algorithms is not required; coding things up may help with understanding
- Mathematical maturity helps

This is a *Theory* course

- All materials are proof-based, and so is the homework
- Implementation of algorithms is not required; coding things up may help with understanding
- Mathematical maturity helps
  - Grasping the mathematical essence is often more important than the “knowledge”

This is a *Theory* course

- All materials are proof-based, and so is the homework
- Implementation of algorithms is not required; coding things up may help with understanding
- Mathematical maturity helps
  - Grasping the mathematical essence is often more important than the “knowledge”
  - Ideas, intuitions, tricks, facts

A Brain Teaser

A Brain Teaser

- The following problem gives you a taste of streaming algorithms

A Brain Teaser

- The following problem gives you a taste of streaming algorithms
- Say you have a very large array of size n, each containing a URL. Strictly more than half of them have the same content. Design an algorithm to find out this URL.

A Brain Teaser

- The following problem gives you a taste of streaming algorithms
- Say you have a very large array of size n, each containing a URL. Strictly more than half of them have the same content. Design an algorithm to find out this URL.
  - Your algorithm must run in linear time ($O(n)$ time)

A Brain Teaser

- The following problem gives you a taste of streaming algorithms
- Say you have a very large array of size n, each containing a URL. Strictly more than half of them have the same content. Design an algorithm to find out this URL.
  - Your algorithm must run in linear time ($O(n)$ time)
    - Better still, go over the array only once

A Brain Teaser

- The following problem gives you a taste of streaming algorithms
- Say you have a very large array of size n, each containing a URL. Strictly more than half of them have the same content. Design an algorithm to find out this URL.
  - Your algorithm must run in linear time ($O(n)$ time)
    - Better still, go over the array only once
  - You have only $O(1)$ additional memory

One Solution

- Use the external memory to remember: a URL (initiated to empty) and a counter (initiated to 0).
- Go over the array. At each new entry, do the following:
  - If the counter is 0, copy the current entry's URL to the stored content, and set the counter to 1
  - Otherwise, compare the current entry's content and the stored content
    - If they are the same, counter++; otherwise counter--
- At the end, output the stored URL.

Extensions

- What if there are at most k URL's, each appearing in strictly more than $\frac{1}{k+1}$ fraction of the entries, for some $k \geq 2$? Can you design an algorithm that finds them all out, in linear time and with $O(1)$ memory?

- Such entries are called *heavy hitters*.