Data Structures & Algorithms

Zhihong Chong
http://cse.seu.edu.cn/people/zhchong/

Outline

- **Why Data structures？**
- **Why Advanced Data Structures?**
- **Essentials of data structures**

Computer Science

- **Computer science**
  - Study of the theoretical foundations of information and computation
  - Study of practical techniques for their implementation and application.
    - Study of algorithmic processes that create, describe and transform information.

1. *theory of computation*
2. *algorithms and data structures*
3. *programming methodology and languages*
4. *computer elements and architecture*

Applied Computer Science

- software engineering
- artificial intelligence
- computer networking and communication
- database systems
- parallel computation, distributed computation
- computer-human interaction
- computer graphics
- operating systems
- numerical and symbolic computation

Summary

- The key problem of CS:
  - What can be efficiently automated?
  - What can be efficiently automated with a business model?
- The key idea
  - Recursive递归
  - Divide and Conquer分而治之
- The key techniques
  - Stratification成层
  - Abstraction
  - Virtualization虚拟化

Data structures and data-intensive applications

- Large data sets
  - RDF data sets
  - Large web pages stored by google and baidu
- Polynomial algorithms on large data sets
  - Can not react in real-time way
  - Even linear algorithms can not work

Essentials of Data structures要领

1. Data structure=f(Data set, R)

   ① R is a relation over data set

   ② Linear structures

      I. Stacks

      II. Queues

   ③ Non-linear structures

      I. Tree structures

      II. Graph structures

Data Organization

- Linear file
  - Physical Neighbor relation
- Linked file链接文件
  - Pointer
- Indexed file索引文件
  - Map function in table form
- Hash file散列文件
  - Map function in explicit form

Data structures

1. The fundamental data-structuring problem is that of maintaining sets of items drawn from a universe so as to efficiently support **search queries**, **update operations** and **operations involving entire sets(aggregation)**.

① Data sets are very large

② Data sets are dynamic

③ The relational structure of data are complex

I. Structured data

II. Semi-structured data

III. Non-structured data

Efficient

1. **O(log n) access time complexity**

2. **Reasonable memory cost!**

Data structure-conclusion

- 数据结构就是指按一定的逻辑结构组成的一批数据，使用某种存储结构将这批数据存储与计算机中，并在这些数据上定义了一个操作集合
  1. 数据的逻辑结构
     - 线性和非线性结构
  2. 数据的存储结构
     - 线性存储空间（连续的存储单元）
  3. 数据的操作集合
     - 查询、更新和聚集

Matching Residents to Hospitals

Goal. Given a set of preferences among hospitals and medical school students, design a self-reinforcing admissions process.

Unstable pair: applicant x and hospital y are unstable if:

- x prefers y to its assigned hospital.
- y prefers x to one of its admitted students.

Stable assignment. Assignment with no unstable pairs.

- Natural and desirable condition.
- Individual self-interest will prevent any applicant/hospital deal from being made.

Stable Matching Problem

Goal. Given n men and n women, find a "suitable" matching.

- Participants rate members of opposite sex.
- Each man lists women in order of preference from best to worst.
- Each woman lists men in order of preference from best to worst.

favorite → 1st; least favorite → 3rd

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → 1st; least favorite → 3rd

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Data Structure and Algorithm

Stable Matching Problem

Perfect matching: everyone is matched monogamously.

- Each man gets exactly one woman.
- Each woman gets exactly one man.

Stability: no incentive for some pair of participants to undermine assignment by joint action.

- In matching M, an unmatched pair m-w is unstable if man m and woman w prefer each other to current partners.
- Unstable pair m-w could each improve by eloping.

Stable matching: perfect matching with no unstable pairs.

Stable matching problem. Given the preference lists of n men and n women, find a stable matching if one exists.

Stable Matching Problem

Q. Is assignment X-C, Y-B, Z-A stable?

favorite ↓ least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite ↓ least favorite ↓

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Problem

Q. Is assignment X-C, Y-B, Z-A stable?
A. No. Bertha and Xavier will hook up.

	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	1st (favorite)	2nd	3rd (least favorite)
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Problem

Q. Is assignment X-A, Y-B, Z-C stable?
A. Yes.

	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	1st (favorite)	2nd	3rd (least favorite)
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Propose-And-Reject Algorithm

Propose-and-reject algorithm. [Gale-Shapley 1962] Intuitive method that guarantees to find a stable matching.

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman) {
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé m')
        assign m and w to be engaged, and m' to be free
    else
        w rejects m
}
```

Proof of Correctness: Termination

Observation 1. Men propose to women in decreasing order of preference.

Observation 2. Once a woman is matched, she never becomes unmatched; she only "trades up."

Claim. Algorithm terminates after at most n^2 iterations of while loop.
Pf. Each time through the while loop a man proposes to a new woman. There are only n^2 possible proposals. ▪

Proof of Correctness: Perfection

Claim. All men and women get matched.

Pf. (by contradiction)

- Suppose, for sake of contradiction, that Zeus is not matched upon termination of algorithm.
- Then some woman, say Amy, is not matched upon termination.
- By Observation 2, Amy was never proposed to.
- But, Zeus proposes to everyone, since he ends up unmatched. ▪

Efficient Implementation

Efficient implementation. We describe $O(n^2)$ time implementation.

Representing men and women.

- Assume men are named 1, …, n.
- Assume women are named 1', …, n'.

Engagements.

- Maintain a list of free men, e.g., in a queue.
- Maintain two arrays `wife[m]`, and `husband[w]`.
  - set entry to `0` if unmatched
  - if m matched to w then `wife[m]=w` and `husband[w]=m`

Men proposing.

- For each man, maintain a list of women, ordered by preference.
- Maintain an array `count[m]` that counts the number of proposals made by man `m`.

Efficient Implementation

Women rejecting/accepting.

- Does woman `w` prefer man `m` to man `m'`?
- For each woman, create inverse of preference list of men.
- Constant time access for each query after $O(n)$ preprocessing.

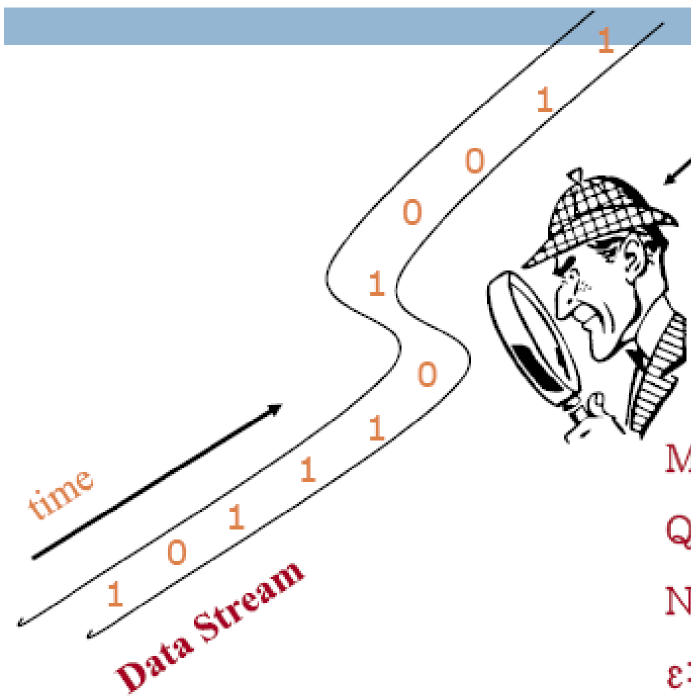
Amy	1st	2nd	3rd	4th	5th	6th	7th	8th
Pref	8	3	7	1	4	5	6	2

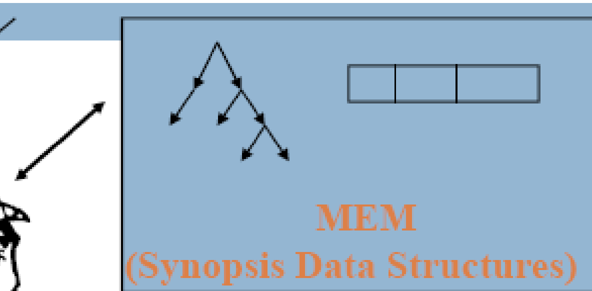
Amy	1	2	3	4	5	6	7	8
Inverse	4th	8th	2nd	5th	6th	7th	3rd	1st

```
for i = 1 to n
   inverse[pref[i]] = i
```

Amy prefers man 3 to 6
since `inverse[3]` < `inverse[6]`
2 7

Streaming Algorithms





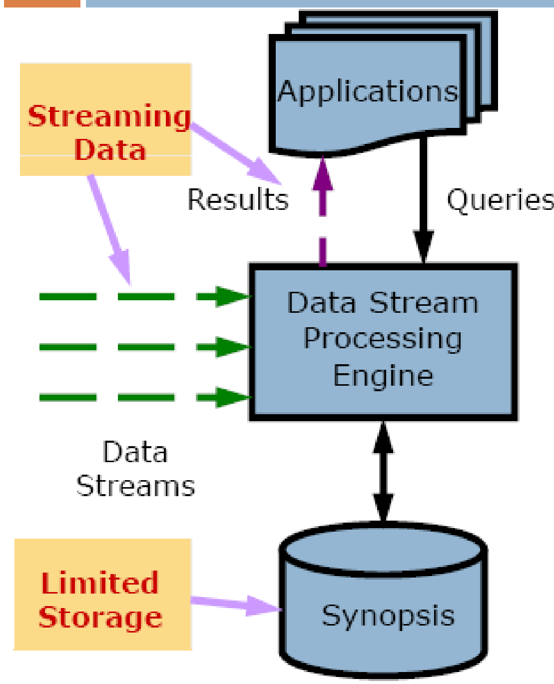
MEM: poly(1/ε, log N)

Query/Update: poly(1/ε, log N)

N: stream size or window size

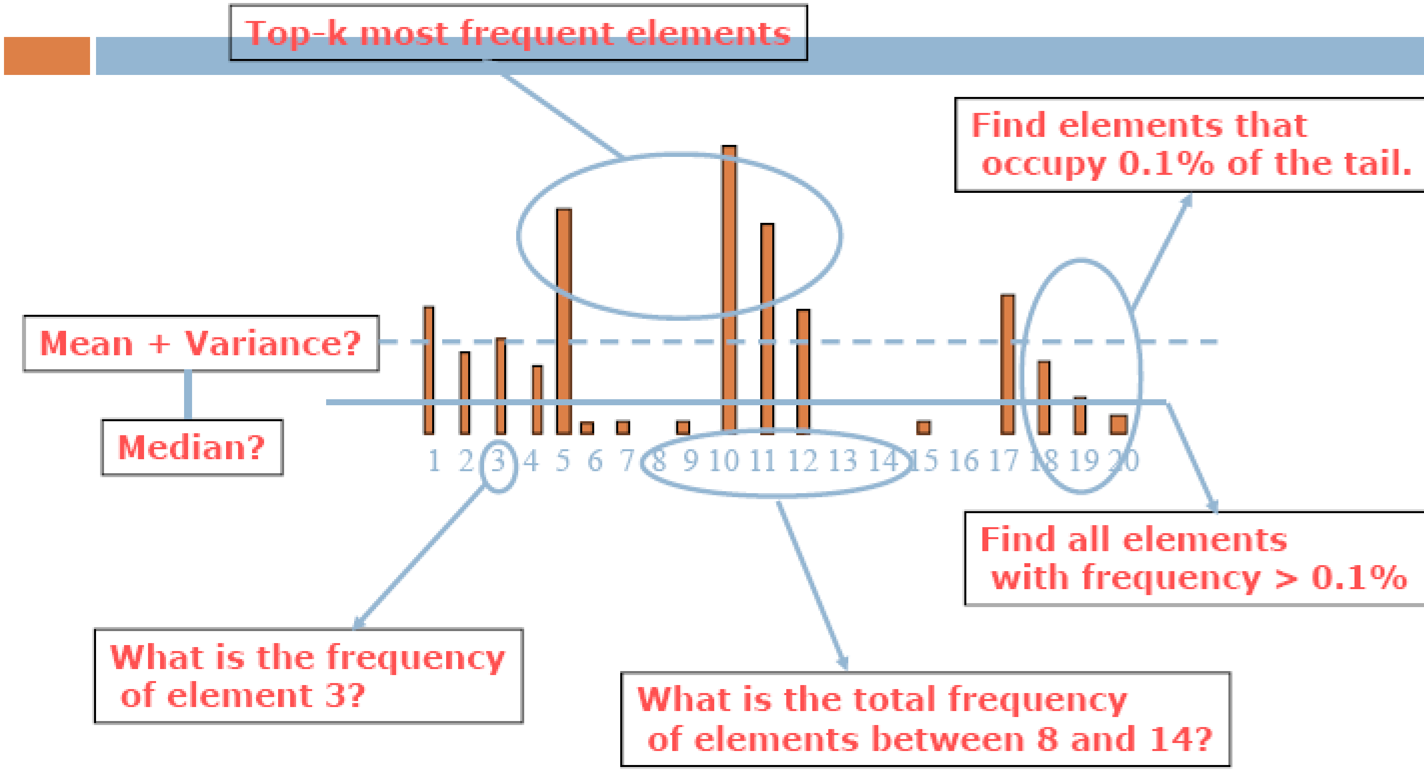
ε: error parameter

Data Stream and Queries



- Data streams
  - Digits (e.g., head of a data packet)
  - Structured tuple (e.g., Web-log)
  - XML documents (e.g., pub/sub system)
- Queries
  - Continuous/ad hoc Queries
  - May have different complexity
    - Counting
    - Structured queries (SQL)
    - Data analysis and mining

Frequency Related Problems

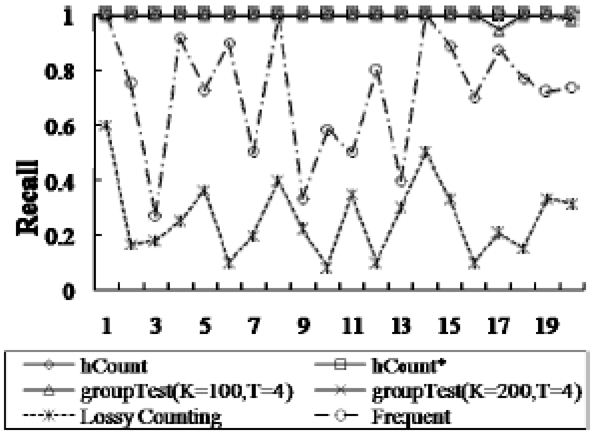


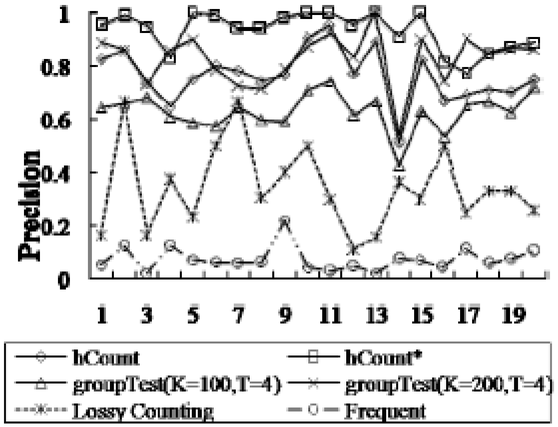
WebDB.cn Open Group,
Southeast University

8	3	11	6	2
7	0	1	14	8
2	5	5	10	8
8	2	6	2	12

Estimate(6)=min(2, 8, 2, 2)=2

- Low space complexity: $O(\varepsilon^{-1}\log M)$
- Fast per-tuple processing cost
- High recall and precision





Readings

What's Hot and What's Not: Tracking Most Frequent Items Dynamically

Graham Cormode
Center for Discrete Mathematics
and Computer Science (DIMACS)
Rutgers University, Piscataway NJ
graham@dimacs.rutgers.edu

S. Muthukrishnan
Department of Computer Science
Rutgers University, Piscataway NJ
muthu@cs.rutgers.edu