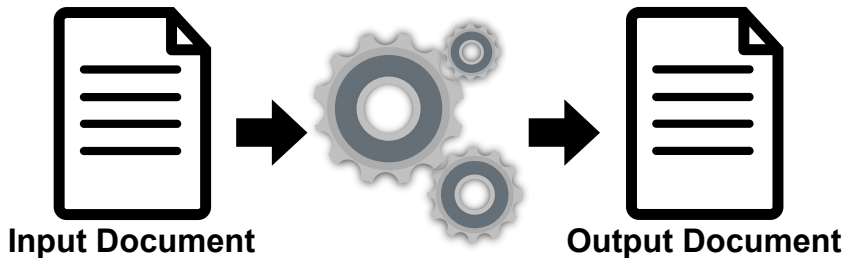


Data Stream Processing

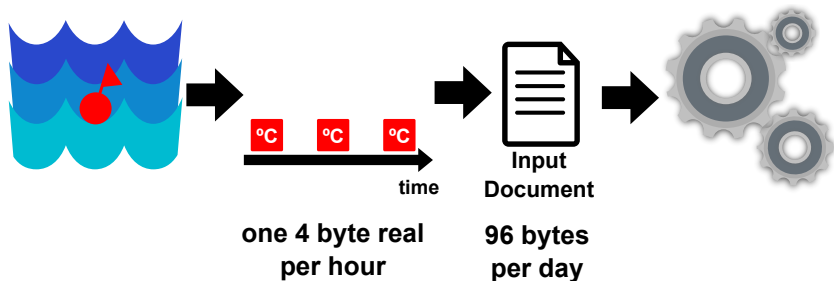
Part I

Homework 1 is due
this Friday the 20th of October

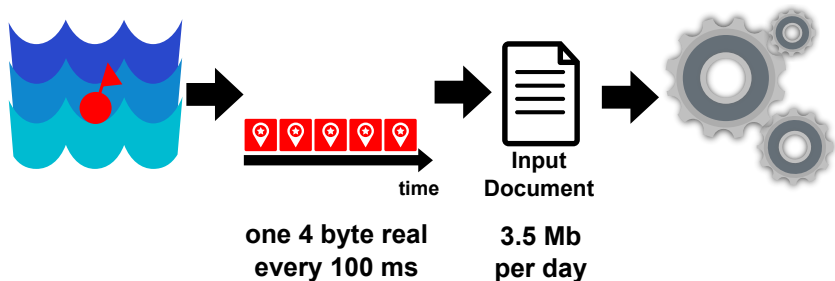
Data Processing so far ...



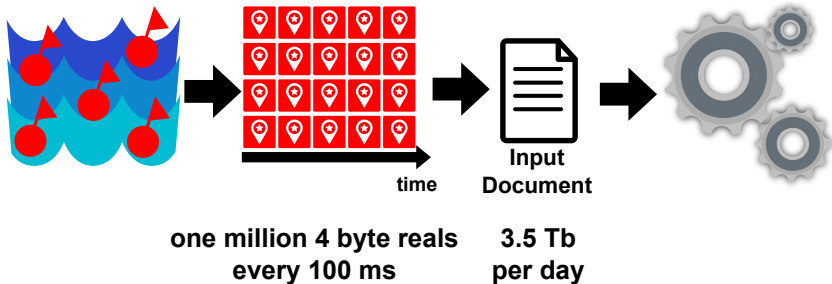
Sensor Data Example



Sensor Data Example



Sensor Data Example



Sensor Data Example

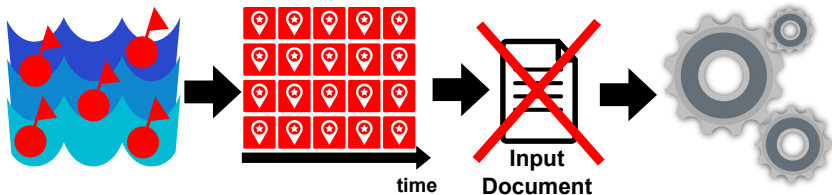
Stream of large unbounded data

too large for memory

too high latency for disk

We need real time processing!

Sensor Data Example



Process data stream directly

Data Streams

What is a Data Stream?

Definition (Golab and Ozs, 2003)

A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor it is feasible to locally store a stream in its entirety.

What is a Data Stream?

Definition (Golab and Ozsu, 2003)

A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor it is feasible to locally store a stream in its entirety.

- continuous and sequential input
- typically unpredictable input rate
- can be large amounts of data
- not error free

Data Stream Applications

- Online, real time processing
- Event detection and reaction
- Aggregation
- Approximation

Data Stream Example

Stock monitoring

Data Stream Example

Stock monitoring
Website traffic monitoring

Data Stream Example

Stock monitoring

Website traffic monitoring

Network management

Data Stream Example

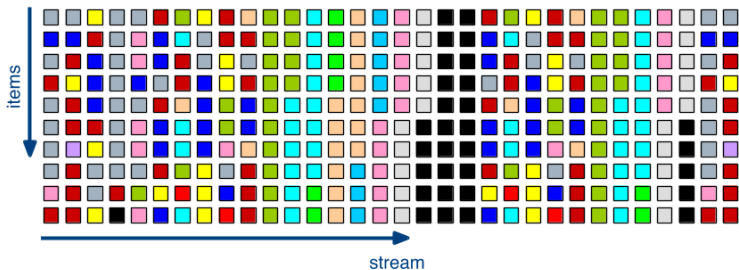
Stock monitoring

Website traffic monitoring

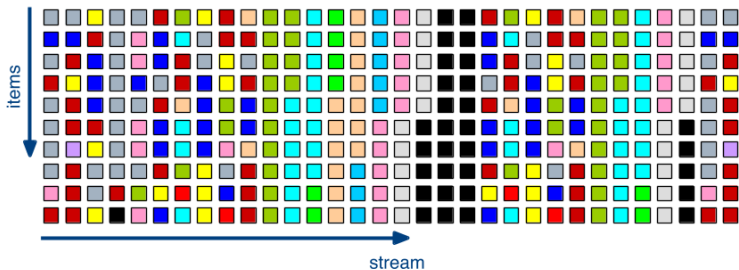
Network management

Highway traffic

Data Stream Characteristics

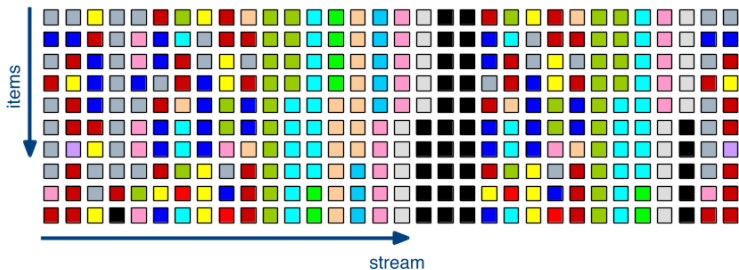


Data Stream Characteristics



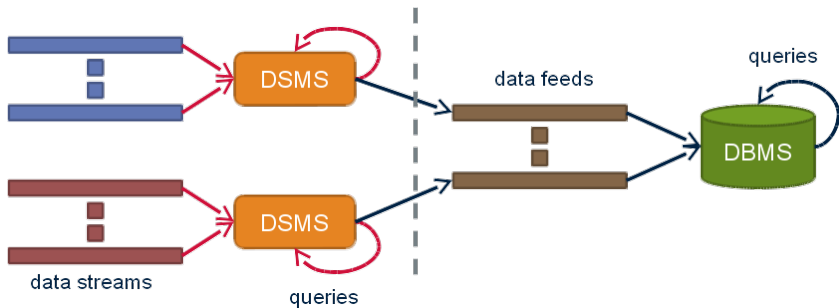
- All items have the same structure. For example a tuple or object: (sender, recipient, text body)

Data Stream Characteristics



- All items have the same structure. For example a tuple or object: (sender, recipient, text body)
- timestamps: explicite vs. implicite, physical vs. logical

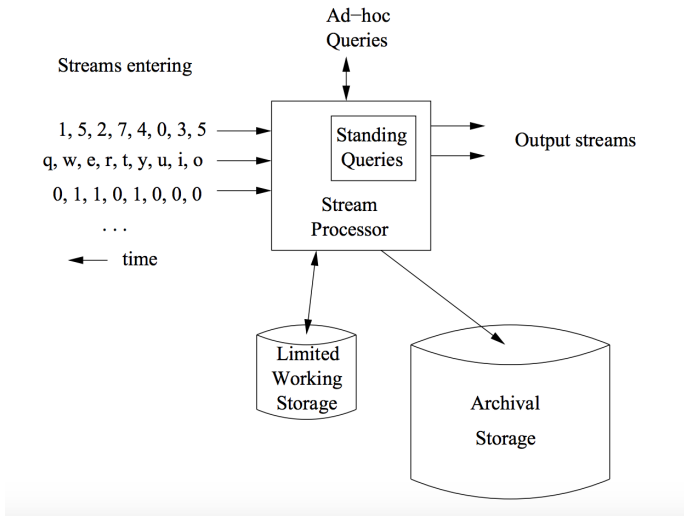
Database Management vs. Data Stream Management



DBMS vs. DSMS

Feature	DBMS	DSMS
Model	persistent relation	transient relation
Relation	tuple set/bag	tuple sequence
Data update	modifications	appends
Query	transient	persistent
Query answer	exact	approximate
Query evaluation	arbitrary	one pass
Query plan	fixed	adaptive

DSMS Architecture



Data Stream Mining

Data Stream Mining

- event detection and reaction
- counting frequency of specific items
- pattern detection
- aggregation
- approximation
- sampling

Data Stream Mining

- event detection and reaction
- counting frequency of specific items
- pattern detection
- aggregation
- approximation
- sampling

Reservoir Sampling

Problem: Sampling

Lines from a large text file

Stream: Sample search engine queries, updated live

The Simple Way

- 1 Scan the text file, counting lines
- 2 Generate random line numbers $[0, |lines|)$
- 3 Sort the line numbers
- 4 Scan the text file, outputting selected lines

The Simple Way

- 1 Scan the text file, counting lines
- 2 Generate random line numbers $[0, |lines|)$
- 3 Sort the line numbers
- 4 Scan the text file, outputting selected lines

Cost: two scans

The Simple Way

- 1 Scan the text file, counting lines
- 2 Generate random line numbers $[0, |lines|)$
- 3 Sort the line numbers
- 4 Scan the text file, outputting selected lines

Cost: two scans

Impossible / Impractical for stream

The Simple Way for a Stream

Problem: Sample top 1000 queries

- 1 assign each query a random number
- 2 keep the queries with the top 1000 highest random numbers
- 3 discard the rest

The Simple Way for a Stream

Problem: Sample top 1000 queries

- 1 assign each query a random number
- 2 keep the queries with the top 1000 highest random numbers
- 3 discard the rest

Additional storage required for random numbers.

The Simple Way for a Stream

Problem: Sample top 1000 queries

- 1 assign each query a random number
- 2 keep the queries with the top 1000 highest random numbers
- 3 discard the rest

Additional storage required for
random numbers.

So far not reservoir sampling!

Sample One Line

Probability of keeping a line and dropping all others?

- keep 1st line:

Sample One Line

Probability of keeping a line and dropping all others?

- keep 1st line: 1

Sample One Line

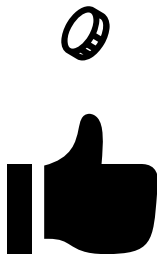
Probability of keeping a line and dropping all others?

- keep 1st line: 1
- keep 2nd line:

Sample One Line

Probability of keeping a line and dropping all others?

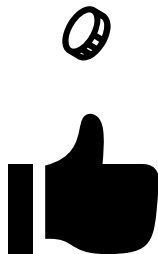
- keep 1st line: 1
- keep 2nd line:



Sample One Line

Probability of keeping a line and dropping all others?

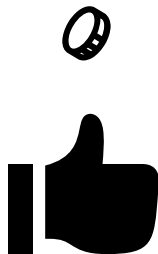
- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$



Sample One Line

Probability of keeping a line and dropping all others?

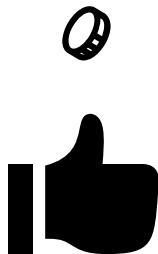
- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line: $\frac{1}{2}$



Sample One Line

Probability of keeping a line and dropping all others?

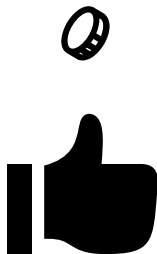
- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line: $\frac{1}{2}$
- keep nth line:



Sample One Line

Probability of keeping a line and dropping all others?

- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line: $\frac{1}{2}$
- keep nth line: $\frac{1}{2}$



Sample One Line

Flip a coin at each line.
If it's heads, record the line (and forget the others).

```
#!/usr/bin/env python
import sys
import random
resevoir = sys.stdin.readline().strip()
for line in sys.stdin:
    if random.randint(0,1) == 0:
        resevoir = line.strip()
print(resevoir)
```

Sample One Line

Flip a coin at each line.
If it's heads, record the line (and forget the others).

```
#!/usr/bin/env python
import sys
import random
resevoir = sys.stdin.readline().strip()
for line in sys.stdin:
    if random.randint(0,1) == 0:
        resevoir = line.strip()
print(resevoir)
```

This is biased. The last line has probability $\frac{1}{2}$.

Sample One Line

Flip a coin at each line.
If it's heads, record the line (and forget the others).

```
#!/usr/bin/env python
import sys
import random
resevoir = sys.stdin.readline().strip()
for line in sys.stdin:
    if random.randint(0,1) == 0:
        resevoir = line.strip()
print(resevoir)
```

This is biased. The last line has probability $\frac{1}{2}$.
It should be the same probability for each line!

Uniformly Sample One Line

- keep 1st line: 1
- keep 2nd line:
- keep 3rd line:
- keep nth line:

Uniformly Sample One Line

- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line:
- keep nth line:

Uniformly Sample One Line

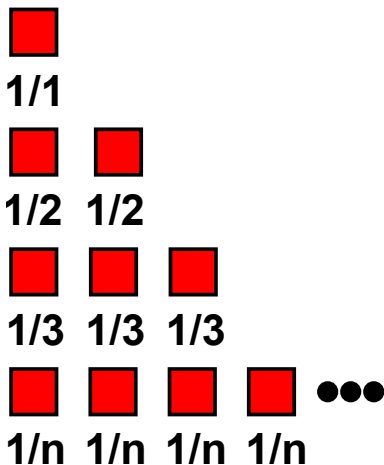
- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line: $\frac{1}{3}$
- keep nth line:

Uniformly Sample One Line

- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line: $\frac{1}{3}$
- keep nth line: $\frac{1}{n}$

Uniformly Sample One Line

- keep 1st line: 1
- keep 2nd line: $\frac{1}{2}$
- keep 3rd line: $\frac{1}{3}$
- keep nth line: $\frac{1}{n}$



Uniformly Sample One Line

```
#!/usr/bin/env python
import sys
import random
line_number = 0
for line in sys.stdin:
    if random.randint(0, line_number) == 0:
        resevoir = line.strip()
    line_number += 1
print(resevoir)
```

Line n overwrites the resevoir with probability $\frac{1}{n}$
 \implies Uniform sampling

Proof Sketch: Induction

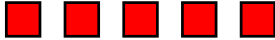
Base One line with probability 1.

Inductive Assume n lines were sampled with probability $\frac{1}{n}$ each. When the $n + 1$ th line is added, the reservoir is kept with probability $\frac{n}{n+1}$. Thus the first n lines each have probability

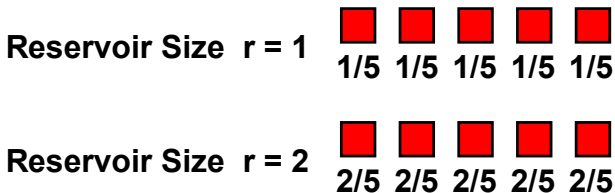
$$\frac{1}{n} \cdot \frac{n}{n+1} = \frac{1}{n+1}$$

And the $n + 1$ th line also has probability $\frac{1}{n+1}$ by construction.

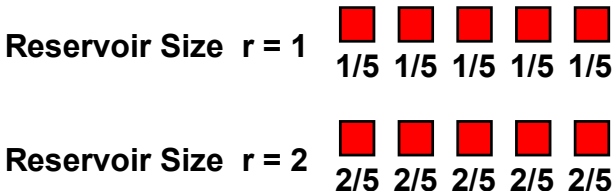
Sample Multiple Lines

Reservoir Size $r = 1$ 
 $1/5$ $1/5$ $1/5$ $1/5$ $1/5$

Sample Multiple Lines



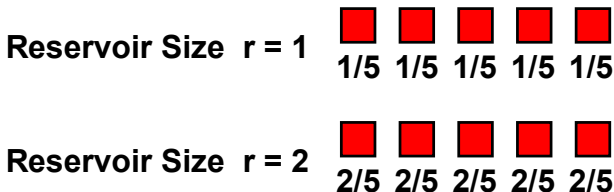
Sample Multiple Lines



with reservoir size r and sample count n

Substitute an entry with probability:

Sample Multiple Lines



with reservoir size r and sample count n

Substitute an entry with probability: $\frac{r}{n}$

Sample Multiple Lines Without Replacement

First few lines: Fill the reservoir

Afterwards: Substitute an entry with probability $\frac{|\text{samples}|}{|\text{lines}|}$

Summary

Efficiently sample streaming data
Small memory