Lecture 1: Distinct Elements and Frequency Moments in Data Streams

Michael Kapralov

EPFL

May 23, 2017
Streaming model

Observe a (very long) stream of data, e.g. IP packets, tweets, search queries....

Task: maintain (approximate) statistics of the stream
Streaming model

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Small (sublinear) storage: typically $n^\alpha, \alpha < 1$ or $\log^{O(1)} n$
  
  Units of storage: bits, words or ‘data items’ (e.g., points, nodes/edges)

- Fast processing time per element

- Mostly randomized algorithms
  
  Randomness often necessary
In this lecture:

- Distinct elements
- Frequency moments (AMS sketch)
In this lecture:

- Distinct elements
- Frequency moments (AMS sketch)
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Distinct elements problem

- Single pass over the data: \( i_1, i_2, \ldots, i_{\text{poly}(n)} \)
  
  Typically, assume \( n \) is known, \( i_j \in [n] \)

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get \( \log^{O(1)} n \)
  
  Much better than storing all items!

![Diagram of distinct elements]
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

$\{1, 2, 3, 4, 5, 7, 9, 10\}$
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

```
#distinct elements= #
1 2 3 4 5 7 9 10 = 8
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

```
#distinct elements=
{1, 2, 3, 4, 5, 7, 9, 10}=8
```

```
3 4 3 2
```

```
1 2 3 4 5 6 7 8 9 10
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

```
3 4 3 2 10
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

```plaintext
3 4 3 2 10 3
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

```
#distinct elements=
{1, 2, 3, 4, 5, 7, 9, 10}=8
```

```
1 3 4 3 2 10 3 1
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!
Distinct elements problem

- Single pass over the data: \( i_1, i_2, \ldots, i_{\text{poly}(n)} \)
  
  Typically, assume \( n \) is known, \( i_j \in [n] \)

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get \( \log^{O(1)} n \)
  
  Much better than storing all items!

```
3 4 3 2 10 3 1 3 1 2
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!

```
3  4  3  2  10  3  1  3  1  2  2
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!

```
# distinct elements = {1, 2, 3, 4, 5, 7, 9, 10} = 8
```

```
1 2 3 4 5 6 7 8 9 10
head
tail
```

```
3 4 3 2 10 3 1 3 1 2 2 5
```
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

3 4 3 2 10 3 1 3 1 2 2 5 5
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

![Histogram showing distinct elements](image-url)
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

1 2 3 4 5 7 9 10

head
tail

3 4 3 2 10 3 1 3 1 2 2 5 5 5 9
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!
Distinct elements problem

- Single pass over the data: \( i_1, i_2, \ldots, i_{\text{poly}(n)} \)

  Typically, assume \( n \) is known, \( i_j \in [n] \)

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get \( \log^{O(1)} n \)

  Much better than storing all items!
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!

![Histogram of distinct elements](image)
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!
Distinct elements problem

- Single pass over the data: \( i_1, i_2, \ldots, i_{\text{poly}(n)} \)

  Typically, assume \( n \) is known, \( i_j \in [n] \)

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get \( \log^{O(1)} n \)

  Much better than storing all items!
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$

  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen

  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$

  Much better than storing all items!
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_{\text{poly}(n)}$
  
  Typically, assume $n$ is known, $i_j \in [n]$

- Output number of distinct elements seen
  
  (Approximately, randomness ok)

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!

\[
\text{#distinct elements} = \#\{1, 2, 3, 4, 5, 7, 9, 10\} = 8
\]
Estimating number of IP flows through a router

Estimate the # of IP flows through a router

```
| source | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
```
Estimating number of IP flows through a router

```
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0
```
Estimating number of IP flows through a router

```
<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
```

Src | Dst
---|---
DATA

D-Link Wireless Router

DATA
Estimating number of IP flows through a router

```
source                   destination
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

| Source | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| destination | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Source: 0
Destination: 0
Estimating number of IP flows through a router

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Src</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Dst</strong></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Source:** DATA
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 2 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 1 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

![Router Image]

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

destination

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

D-Link US-920

D-Link US-920 Wireless N Gigabit Router

Figure 1: Example of a router with a data flow matrix.
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 1</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**destination**

![Router Diagram](image)
Estimating number of IP flows through a router
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 4 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>3 0 0 0 0 0 0 0 1 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>DATA</td>
</tr>
</tbody>
</table>
### Estimating number of IP flows through a router

#### Source

<table>
<thead>
<tr>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 4 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 3 0 0 0 0 0 0 0 1 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

#### Destination

<table>
<thead>
<tr>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

```
<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>

Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 4 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 1 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATA</td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>source</th>
<th>destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 4 0 0 0 0 0 0 0 1 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>
### Estimating number of IP flows through a router

| Source | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Source:**
- 1: The source of the IP flow.

**Destination:**
- 0: No flow.
- 1: Flow from source to destination.

**DATA:**
- The table above represents the number of IP flows between different source and destination pairs.
Estimating number of IP flows through a router

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 4 0 0 0 0 0 0 0 0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Estimating number of IP flows through a router

Estimate the # of IP flows through a router
Estimating number of IP flows through a router

Estimate the # of IP flows through a router

Trivial: store all distinct IP pairs

Space complexity: $\Omega(n)$
Estimating number of IP flows through a router

Estimate the # of IP flows through a router

Trivial: store all distinct IP pairs

Space complexity: $\Theta(n)$

This lecture: solve in space $\log^{O(1)} n$

Exponential improvement!
Estimating search statistics

Given a set of items as a stream (e.g. queries on google.com over a period of time)

Geneva to NYC, coffee in Geneva, Geneva to NYC
Estimating search statistics

Given a set of items as a stream (e.g. queries on google.com over a period of time)

Geneva to NYC, coffee in Geneva, Geneva to NYC

Find the # of distinct items in the set

Geneva to NYC, coffee in Geneva
**Streaming model**

<table>
<thead>
<tr>
<th>Solution</th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>hash&lt;string&gt; h; # of distinct items</td>
<td>HYPERLOGLOG log&lt;sup&gt;O(1)&lt;/sup&gt; n</td>
</tr>
</tbody>
</table>
## Streaming model

<table>
<thead>
<tr>
<th>Solution</th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>$\text{hash&lt;}\text{string}&gt; \ h;$</td>
<td>HYPERLOGLOG $\log^{O(1)} n$</td>
</tr>
<tr>
<td></td>
<td># of distinct items</td>
<td></td>
</tr>
</tbody>
</table>

Are constants small?
Streaming model

<table>
<thead>
<tr>
<th></th>
<th>Trivial</th>
<th>This lecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution</td>
<td>$\text{hash&lt;string&gt; h;}$</td>
<td>$\text{HyperLogLog}$</td>
</tr>
<tr>
<td>Space</td>
<td>$# \text{ of distinct items}$</td>
<td>$\log^{O(1)} n$</td>
</tr>
</tbody>
</table>

Are constants small?

**HyperLogLog**: estimate Shakespeare’s vocabulary using 128 bits of memory
Streaming model

Widely used in practice for **scalable data analytics**

*most frequent searches on google.com over a time period*

*most frequent tweets*
Distinct elements problem

- Single pass over the data: $i_1, i_2, \ldots, i_n$
  
  integers between 1 and poly($n$)

- Output $(1 \pm \varepsilon)$-approximation to # of distinct elements
  
  $(1 - \varepsilon)\text{DE} \leq \hat{\text{DE}} \leq (1 + \varepsilon)\text{DE}$

- Small storage: will get $\log^{O(1)} n$
  
  Much better than storing all items!

- Success probability $\geq 1 - \delta$
Simpler goal: for a given $T > 0$, provide an algorithm ALG that, with probability $1 - \delta$:

- answers YES if $DE > (1 + \varepsilon)T$
- answers NO if $DE < (1 - \varepsilon)T$
Simpler goal: for a given $T > 0$, provide an algorithm ALG that, with probability $1 - \delta$:

- answers YES if $\text{DE} > (1 + \varepsilon)T$
- answers NO if $\text{DE} < (1 - \varepsilon)T$

To achieve the original goal, run in ALG with thresholds

$$T = 1, 1 + \varepsilon, (1 + \varepsilon)^2, \ldots, n$$
Simpler goal: for a given $T > 0$, provide an algorithm ALG that, with probability $1 - \delta$:

- answers YES if $\text{DE} > (1 + \varepsilon)T$
- answers NO if $\text{DE} < (1 - \varepsilon)T$

To achieve the original goal, run in ALG with thresholds

$$T = 1, 1 + \varepsilon, (1 + \varepsilon)^2, \ldots, n$$

- total space multiplied by $\log_{1+\varepsilon} n \approx \frac{1}{\varepsilon} \log n$
- failure probability multiplied by same factor
Vector interpretation

- Initially, $x = 0$
- Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

- Want to estimate $DE(x)$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

\[ x \in \mathbb{R}^n \]

- Initially, \( x = 0 \)
- Insertion of \( i \) interpreted as \( x_i := x_i + 1 \)
- Want to estimate \( \text{DE}(x) \)
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

Initially, \( x = 0 \)

- Insertion of \( i \) interpreted as
  \[ x_i := x_i + 1 \]

- Want to estimate \( \text{DE}(x) \)
Vector interpretation

 Initially, \( x = 0 \)

 Insertion of \( i \) interpreted as

\[ x_i := x_i + 1 \]

 Want to estimate \( \text{DE}(x) \)
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $DE(x)$
Vector interpretation

- Initially, $x = 0$
- Insertion of $i$ interpreted as $x_i := x_i + 1$
- Want to estimate $\text{DE}(x)$
Vector interpretation

$\mathbf{x} \in \mathbb{R}^n$

1 2 3 4 5 6 7 8 9 10

- Initially, $\mathbf{x} = 0$
- Insertion of $i$ interpreted as $\mathbf{x}_i := \mathbf{x}_i + 1$
- Want to estimate $DE(\mathbf{x})$
Vector interpretation

- Initially, $x = 0$
- Insertion of $i$ interpreted as $x_i := x_i + 1$
- Want to estimate $DE(x)$

$\exists \text{ distinct elements} = \{1, 2, 3, 4, 5, 7, 9, 10\} = 8$

$3 \ 4 \ 3 \ 2 \ 10 \ 3 \ 1 \ 3 \ 1 \ 2$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $DE(x)$
Vector interpretation

$\mathbf{x} \in \mathbb{R}^n$

\begin{align*}
3 & \ 4 & \ 3 & \ 2 & \ 10 & \ 3 & \ 1 & \ 3 & \ 1 & \ 2 & \ 2 & \ 5 & \ 5 & \ 5 \\
\end{align*}

- Initially, $\mathbf{x} = 0$
- Insertion of $i$ interpreted as $x_i := x_i + 1$
- Want to estimate $\text{DE}(\mathbf{x})$
Vector interpretation

Initially, $x = 0$

- Insertion of $i$ interpreted as

    $$x_i := x_i + 1$$

- Want to estimate $DE(x)$
Vector interpretation

\[ x \in \mathbb{R}^n \]

#distinct elements = \{1, 2, 3, 4, 5, 7, 9, 10\} = 8

Initially, \( x = 0 \)

- Insertion of \( i \) interpreted as \( x_i := x_i + 1 \)

Want to estimate \( \text{DE}(x) \)
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $DE(x)$
Vector interpretation

- Initially, $x = 0$
- Insertion of $i$ interpreted as $x_i := x_i + 1$
- Want to estimate $DE(x)$

\[ x \in \mathbb{R}^n \]

3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4 4 2
Vector interpretation

\[ x \in \mathbb{R}^n \]

\#distinct elements = \{1, 2, 3, 4, 5, 7, 9, 10\} = 8

Initially, \( x = 0 \)

Insertion of \( i \) interpreted as

\[ x_i := x_i + 1 \]

Want to estimate \( DE(x) \)
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Vector interpretation

- Initially, $x = 0$
- Insertion of $i$ interpreted as $x_i := x_i + 1$
- Want to estimate $\text{DE}(x)$
Vector interpretation

Initially, $x = 0$

Insertion of $i$ interpreted as

$$x_i := x_i + 1$$

Want to estimate $\text{DE}(x)$
Estimating $DE(x)$ – decision problem

- Choose a random set $S \subseteq [n]$ s.t. for each $i \in [n]$
  
  \[ \Pr[i \in S] = \frac{1}{T} \]
Estimating $DE(x)$ – decision problem

- Choose a random set $S \subseteq [n]$ s.t. for each $i \in [n]$
  \[
  \Pr[i \in S] = \frac{1}{T}
  \]
Estimating $\text{DE}(x)$ – decision problem

Choose a random set $S \subseteq [n]$ s.t. for each $i \in [n]$

$$\Pr[i \in S] = 1/T$$

Maintain $c_S := \sum_{i \in S} x_i$
Estimating $\text{DE}(x)$ – decision problem

- Choose a random set $S \subseteq [n]$ s.t. for each $i \in [n]$
  \[ \Pr[i \in S] = \frac{1}{T} \]

- Maintain $c_S := \sum_{i \in S} x_i$
- Estimation:
  - If $c_S > 0$, output YES
  - If $c_S = 0$, output NO
Basic algorithm (decision problem)

Algorithm:
- Choose a random set $S \subseteq [n]$ s.t. for each $i \in [n]$
  \[ \Pr[i \in S] = 1/T \]
- Maintain $c_S := \sum_{i \in S} x_i$
- Estimation:
  - If $c_S > 0$, output YES
  - If $c_S = 0$, output NO

Analysis:
- For $T$ large enough: $\Pr[c_S = 0] = (1 - 1/T)^{DE} \approx e^{-DE/T}$
- So for small enough $\varepsilon$
  - If $DE > (1 + \varepsilon)T$, then $\Pr[c_S = 0] \approx e^{-(1+\varepsilon)} < 1/e - \varepsilon/3$
  - If $DE < (1 - \varepsilon)T$, then $\Pr[c_S = 0] \approx e^{-(1-\varepsilon)} > 1/e + \varepsilon/3$
Full algorithm for decision problem

Basic algorithm:

- If $DE > (1 + \varepsilon)T$, then $\Pr[c_S = 0] < 1/e - \varepsilon/3$
- If $DE < (1 - \varepsilon)T$, then $\Pr[c_S = 0] > 1/e + \varepsilon/3$
Full algorithm for decision problem

Basic algorithm:

- If $DE > (1 + \varepsilon) T$, then $\Pr[c_S = 0] < 1/e - \varepsilon/3$
- If $DE < (1 - \varepsilon) T$, then $\Pr[c_S = 0] > 1/e + \varepsilon/3$

Full algorithm:

- Select sets $S_1, \ldots, S_k$, $k = O(\frac{1}{\varepsilon^2} \log(1/\delta))$
- Maintain counters $c_{S_j}, j \in [k]$
- $Z := \| \{ j \in [k] : c_{S_j} = 0 \} \|
- If $Z < k/e$, say YES
- If $Z \geq k/e$, say NO

Space complexity? Correctness?
Full algorithm for decision problem – space complexity

Basic algorithm:
- If \( \text{DE} > (1 + \varepsilon) T \), then \( \Pr[c_S = 0] < 1/e - \varepsilon/3 \)
- If \( \text{DE} < (1 - \varepsilon) T \), then \( \Pr[c_S = 0] > 1/e + \varepsilon/3 \)

Full algorithm:
- Select sets \( S_1, \ldots, S_k \), \( k = O\left(\frac{1}{\varepsilon^2} \log(1/\delta)\right) \)
- \( Z := \| \{ j \in [k] : c_{S_j} = 0 \} \| \)
- If \( Z < k/e \), say YES
- If \( Z \geq k/e \), say NO

Space:
- Decision problem: \( O\left(\frac{1}{\varepsilon^2} \log(1/\delta)\right) \) numbers in \([0..n^{O(1)}]\)
- Estimation: \( O\left(\frac{1}{\varepsilon^3} \log n \log(1/\delta)\right) \) numbers in \([0..n^{O(1)}]\)
  (error probability \( O(\delta \cdot \frac{1}{\varepsilon} \log n) \))
Theorem

Let $Z_1, \ldots, Z_n$ be i.i.d. Bernoulli random variables with $\mathbb{E}[Z_i] = p$, and let $Z = \sum_{i=1}^{n} Z_i$. Then for every $\varepsilon \in (0, 1)$

$$\Pr \left[ \left| \sum_{i=1}^{n} Z_i - \mathbb{E}[Z] \right| > \varepsilon \mathbb{E}[Z] \right] \leq 2 \exp(-\varepsilon^2 \mathbb{E}[Z]/3).$$
How do we store the set \( S \)?

Choose a hash function

\[
h : [n] \to [1 : T],
\]

let

\[
S = \{ i \in [n] : h(i) = 1 \}
\]
How do we store the set $S$?

Choose a hash function

$$h: [n] \rightarrow [1 : T],$$

let

$$S = \{i \in [n] : h(i) = 1\}$$

- How do we store $h$? :)

Use a pseudorandom number generator (e.g. Nisan's PRG) or redo analysis (with slight modifications) for a pairwise independent $h$. Pairwise independent $h$ can be stored using $O(\log n)$ bits (think $ax + b \mod p$). Ex: redo analysis assuming that $h$ is pairwise independent only.
How do we store the set $S$?

Choose a hash function

$$h : [n] \rightarrow [1 : T],$$

let

$$S = \{ i \in [n] : h(i) = 1 \}$$

- How do we store $h$? :)
- Use a pseudorandom number generator (e.g. Nisan’s PRG)
How do we store the set $S$?

Choose a hash function

\[ h : [n] \rightarrow [1 : T], \]

let

\[ S = \{ i \in [n] : h(i) = 1 \} \]

- How do we store $h$? :)
- Use a pseudorandom number generator (e.g. Nisan’s PRG)
  or
- redo analysis (with slight modifications) for a pairwise independent $h$
  - pairwise independent $h$ can be stored using $O(\log n)$ bits (think $ax + b \mod p$)

Ex: redo analysis assuming that $h$ is pairwise independent only
Linear sketching

Maintain $Sx$ for a matrix $S \in \mathbb{R}^{m \times n}$, $m$ small

$(m = O\left(\frac{1}{\varepsilon^3} \log^2 n \cdot \log(1/\delta)\right))$

So algorithm also works if some elements are deleted:

$x_i := x_i - 1,$

as long as $x \geq 0$ at the end of the stream.
Optimal space bounds, practical algorithms

Asymptotically tight space: $O\left(\frac{1}{\varepsilon^2} + \log n\right)$ bits \text{Kane-Nelson-Woodruff’10}

Practical: Durand-Flajolet’03

And recent practical improvements:

HyperLogLog in Practice: Algorithmic Engineering of a State of The Art Cardinality Estimation Algorithm

Stefan Heule
ETH Zurich and Google, Inc.
stheule@ethz.ch

Marc Nunkesser
Google, Inc.
marcnunkesser@google.com

Alexander Hall
Google, Inc.
alexhall@google.com
Linear sketching

\[ S \begin{bmatrix} m \\ n \end{bmatrix} \cdot x = b \]

sketching matrix

space requirement = number of rows

Later this week: more sketching algorithms for basic statistics, and then graph sketching
Approximate $\|x\|_p$ for other $p$?

Approximate

$$\|x\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

in small space?

Note:

- $\|x\|_\infty = \max_{i \in [n]} |x_i|$
- $\|x\|_0 = \#$distinct elements in $x$
Approximate $\|x\|_p$ for other $p$?

Approximate

$$\|x\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

in small space?

Note:

- $\|x\|_\infty = \max_{i \in [n]} |x_i|$
- $\|x\|_0 = \#\text{distinct elements in } x$

Frequency moments: $F_p = \|x\|_p^p$. 
Approximate $\|x\|_p$ for other $p$?

Approximate

$$\|x\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

in small space?

Note:
- $\|x\|_\infty = \max_{i \in [n]} |x_i|$
- $\|x\|_0 = \# \text{distinct elements in } x$

Frequency moments: $F_p = \|x\|^p_p$.

How much space is needed for $(1 \pm \epsilon)$-approximation to $\|x\|_p$ for constant $\epsilon$?
- $\log^{O(1)} n$ suffices for $p \in (0, 2]$
- $\Omega(n^{1-2/p})$ needed for $p > 2$. 
In this lecture:

- Distinct elements
- Frequency moments (AMS sketch)
In this lecture:

- Distinct elements
- Frequency moments (AMS sketch)
AMS sketch (Alon-Matias-Szegedy’96)

Goal: approximate

$$\|x\|_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to $x_i$. 
AMS sketch (Alon-Matias-Szegedy’96)

Goal: approximate

\[ \|x\|_2 = \sqrt{\sum_{i\in[n]} x_i^2} \]

from a stream of increments/decrements to \(x_i\).

Choose \(r_1, \ldots, r_n\) to be i.i.d. r.v., with

\[ \Pr[r_i = +1] = \Pr[r_i = -1] = 1/2. \]
AMS sketch (Alon-Matias-Szegedy’96)

**Goal:** approximate

\[
\|x\|_2 = \sqrt{\sum_{i \in [n]} x_i^2}
\]

from a stream of increments/decrements to \(x_i\).

Choose \(r_1, \ldots, r_n\) to be i.i.d. r.v., with

\[
\Pr[r_i = +1] = \Pr[r_i = -1] = 1/2.
\]

Maintain

\[
Z = \sum_{i=1}^{n} r_i x_i
\]

under increments/decrements of \(x\).
AMS sketch (Alon-Matias-Szegedy’96)

Goal: approximate

\[ \|x\|_2 = \sqrt{\sum_{i \in [n]} x_i^2} \]

from a stream of increments/decrements to \(x_i\).

Choose \(r_1, \ldots, r_n\) to be i.i.d. r.v., with

\[ \Pr[r_i = +1] = \Pr[r_i = -1] = 1/2. \]

Maintain

\[ Z = \sum_{i=1}^{n} r_i x_i \]

under increments/decrements of \(x\).

Basic algorithm: output \(Z^2\)
AMS sketch (Alon-Matias-Szegedy’96)

Goal: approximate
\[
\|x\|_2 = \sqrt{\sum_{i \in [n]} x_i^2}
\]
from a stream of increments/decrements to \(x_i\).

Choose \(r_1, \ldots, r_n\) to be i.i.d. r.v., with
\[
\Pr[r_i = +1] = \Pr[r_i = -1] = 1/2.
\]

Maintain
\[
Z = \sum_{i=1}^{n} r_i x_i
\]
under increments/decrements of \(x\).

Basic algorithm: output \(Z^2\)

Want to claim that \(Z^2\) is ‘close’ to \(\|x\|_2^2\) with ‘high probability’
Want to claim that $Z^2$ is ‘close’ to $\|x\|_2^2$ with ‘high probability’

**Expectation:**

$$E[Z^2] = E[\left(\sum_{i=1}^{n} r_i x_i\right)^2] = n \sum_{i=1}^{n} \sum_{j=1}^{n} E[r_i r_j x_i x_j] = n \sum_{i=1}^{n} x_i^2 + n \sum_{i \neq j} E[r_i r_j] x_i x_j = \|x\|^2$$

(our estimator is unbiased!)
Want to claim that $Z^2$ is ‘close’ to $||x||_2^2$ with ‘high probability’

Compute expectation of $Z^2$, then bound the variance
Alon-Matias-Szegedy – analysis (expectation)

Want to claim that $Z^2$ is ‘close’ to $||x||^2_2$ with ‘high probability’

Compute expectation of $Z^2$, then bound the variance

Expectation:

$$E[Z^2] = E \left[ \left( \sum_{i=1}^{n} r_i x_i \right)^2 \right]$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} E[r_i r_j x_i x_j]$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} E[r_i r_j] x_i x_j$$

$$= \sum_{i=1}^{n} x_i^2 + \sum_{i,j:i \neq j}^{n} E[r_i] E[r_j] x_i x_j$$

$$= \sum_{i=1}^{n} x_i^2$$

$$= ||x||^2_2$$

(our estimator is unbiased!)