Streaming 1: Estimating the number of elements

Philip Bille Inge Li Gørtz Eva Rotenberg

Algorithmic Techniques for Modern Data Models

Previous courses: RAM-model.

Not everything fits in RAM $\Rightarrow$ the RAM model is not always a suitable model.

In this course: Other models.

- Streaming,
- I/O,
- Approximate data structures,
- Parallel computation.

Four subjects, four mandatory assignments, one oral exam.

To pass: assignments + exam, overall assessment.

Prerequisites: algorithmic and mathematical maturity.

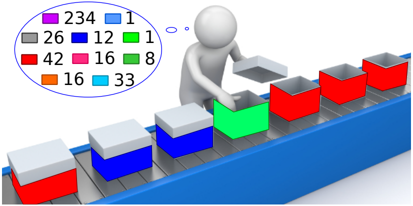
The streaming model

Stream, σ: $a_1, a_2, a_3, \ldots$ of elements $a_i \in U$ from some universe.

Maintain a small *working memory*. When seeing element a_i, update the memory depending only on a_i.

Goal: by the end of the stream, have completed some task.

Majority



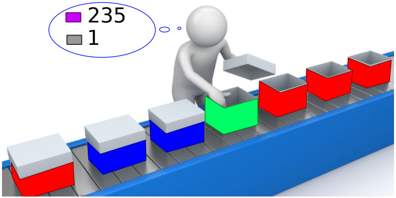
Majority: Is there a colour that more than half of the boxes have?

Naive: Count all colours. Space: n counters of size $\leq \log n$.

Idea: Allow one-sided error. If a majority exists, output the majority. Otherwise, output whatever. Advantage: A second pass can determine whether that one candidate has a true majority.

How many counters? (Try: one counter, and one other variable.)

Frequent elements



if $j \in keys(A)$ **then**
| A[j]++;
else if $|keys(A)| < k - 1$ **then**
| A[j] $\rightarrow$ 1;
else
| decrement all A[j].

Task: Detect very common colours. E.g: Is there some colour that $> 5\%$ of the boxes have? (i.e. $n/k = n/20$ boxes)

Naive: count all occurrences of the $\leq n$ colours.

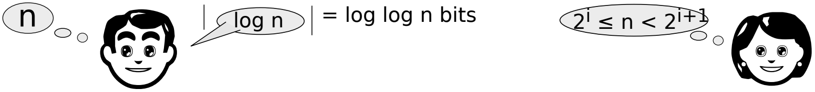
Misra-Gries: Keep track of the $k - 1 = 19$ most common colours.

Space-consumption: $k - 1$ counters of size $\leq \lg n$

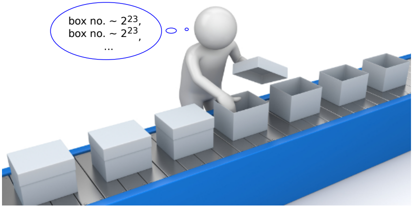
Intermission

Exercise: 4-6 (weekplan) and 1-1, 1-2 (notes)

Counting – in limited space!



Counting

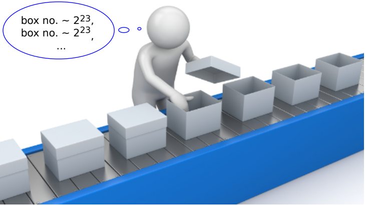


Imagine you want to count the elements.

Space of exact count: $\log n$ bits memory needed.

Approx count: $\log \log n$ bits. Challenge: when to update?  !

Morris' original idea:



Keep an approximate count: store c such that $2^c \simeq n$

Update randomly *with decreasing probability*. Maintain 2^c is n in expectation.

Question: With which probability?

When c turns c_0, $n \simeq 2^{c_0}$, so it should stay there for circa 2^{c_0} turns. $\Rightarrow$ probability circa $1/2^{c_0}$.

(Exercise: smart way of rolling a 2^m-sided dice?)