Count-Min and Augmented Sketch

Sketches for data streams

[Cormode & Muthukrishnan, J. Algorithms 55, 2005]
[Roy, Khan & Alonso, ACM SIGMOD 2016]

Recommended:
[Metwally, Agrawal & Abbadi, 2005]
[Roy, Teubner & Alonso, ACM SIGKDD 2012]

(slides by Patrick Dinklage, released under CC0)

Sketches

Sketches

- **Streaming** scenario: (infinite) sequence of items from a universe $[n]$

Sketches

- **Streaming** scenario: (infinite) sequence of items from a universe $[n]$
- A **sketch** represents *implicitly* a vector of dimension n
  - Much less space (polylog) needed than for explicit representation

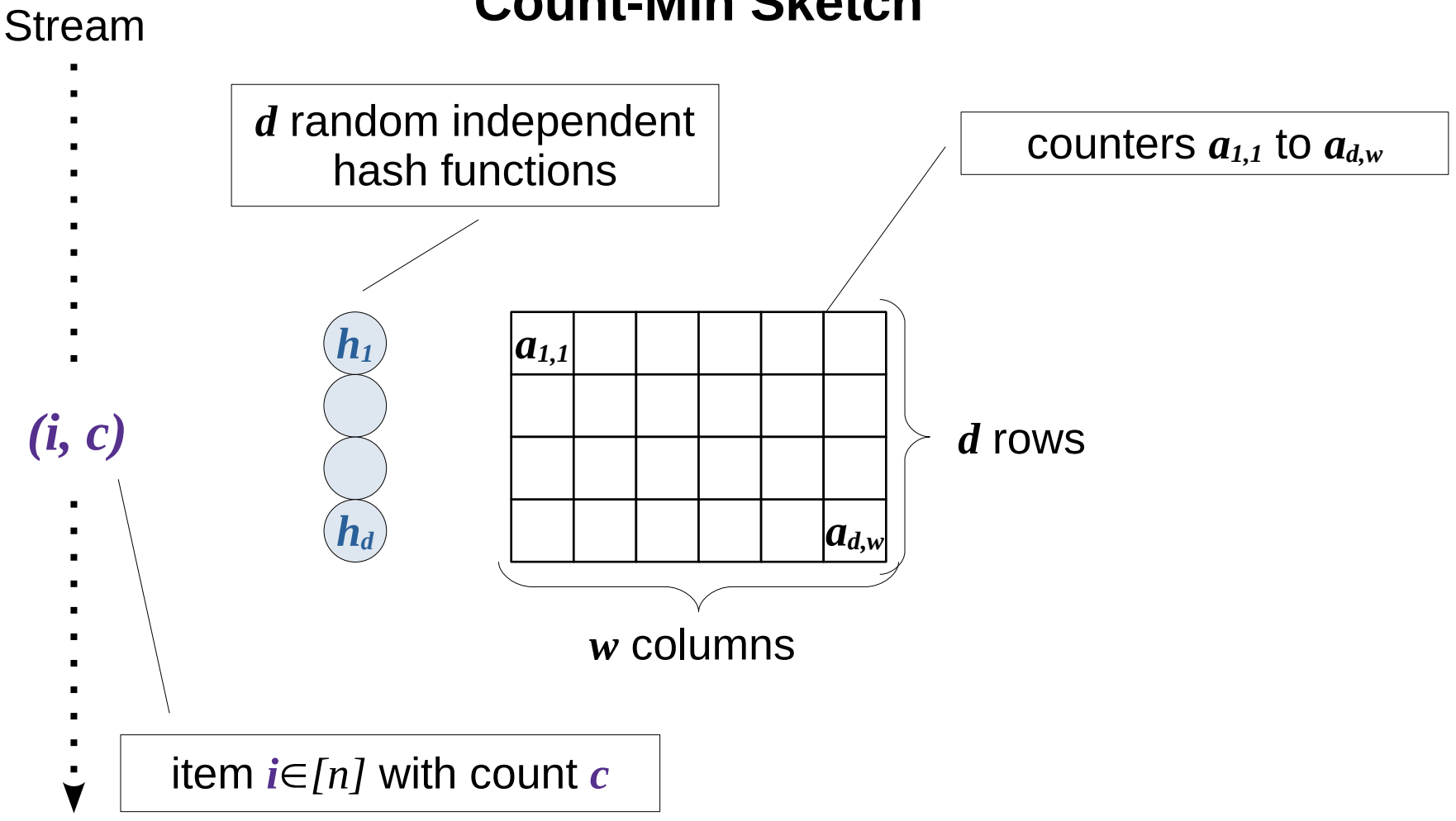
Sketches

- **Streaming** scenario: (infinite) sequence of items from a universe $[n]$
- A **sketch** represents *implicitly* a vector of dimension n
  - Much less space (polylog) needed than for explicit representation
- Commonly of interest: *frequency counters*
  - Frequency estimation, quantiles, heavy hitters, …

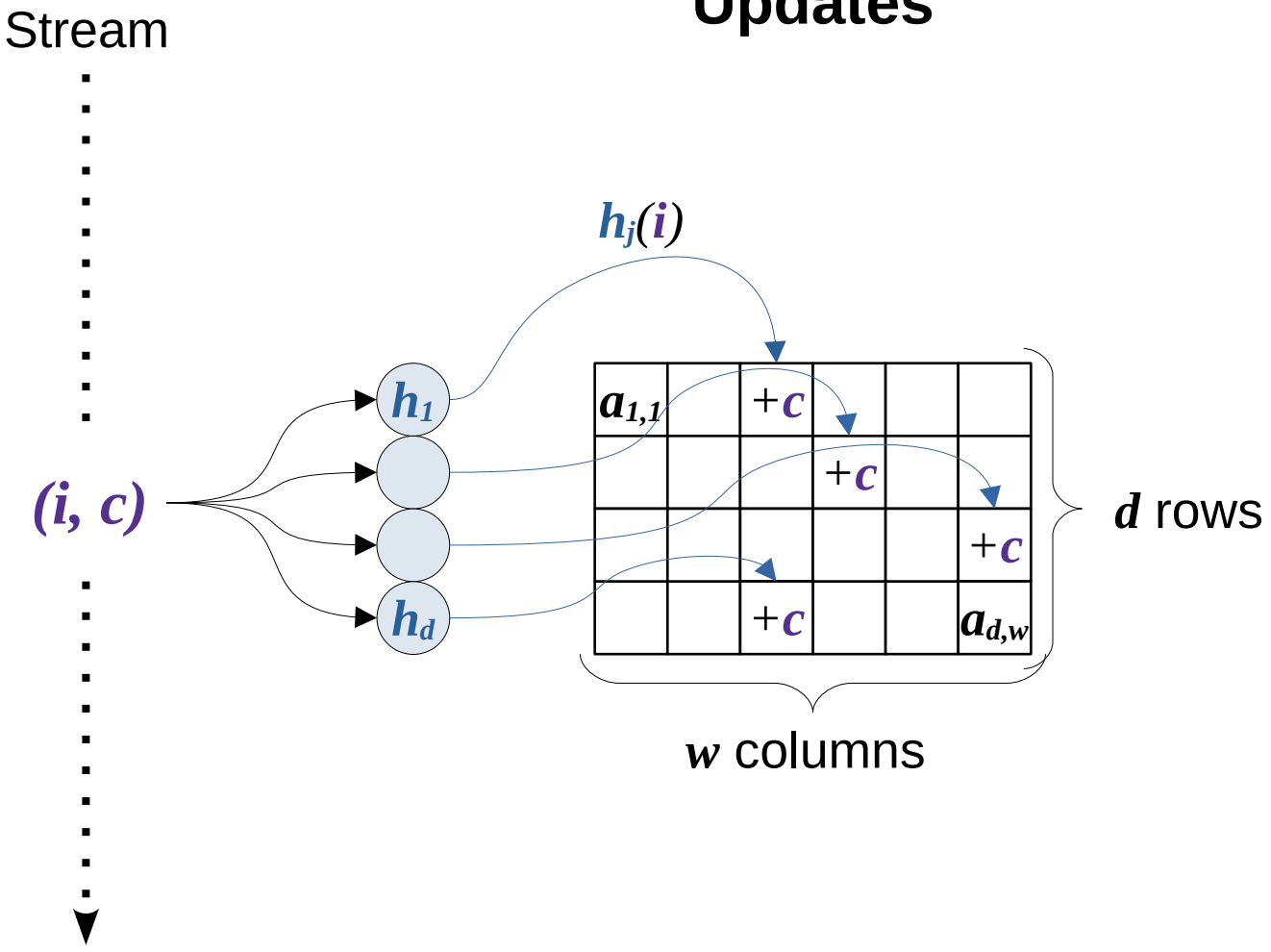
Sketches

- **Streaming** scenario: (infinite) sequence of items from a universe $[n]$
- A **sketch** represents *implicitly* a vector of dimension n
  - Much less space (polylog) needed than for explicit representation
- Commonly of interest: *frequency counters*
  - Frequency estimation, quantiles, heavy hitters, …
- Reasonable error margin w.h.p. for queries

Count-Min Sketch



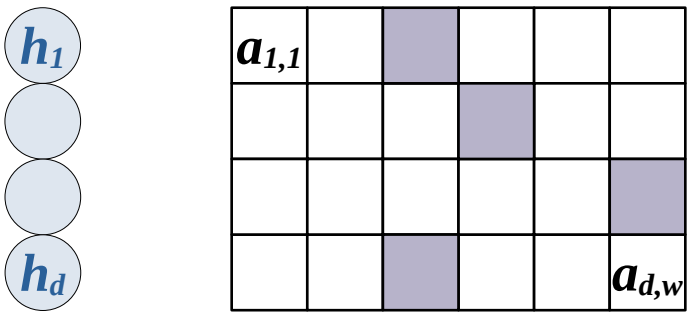
Updates



Point Queries

(non-negative case)

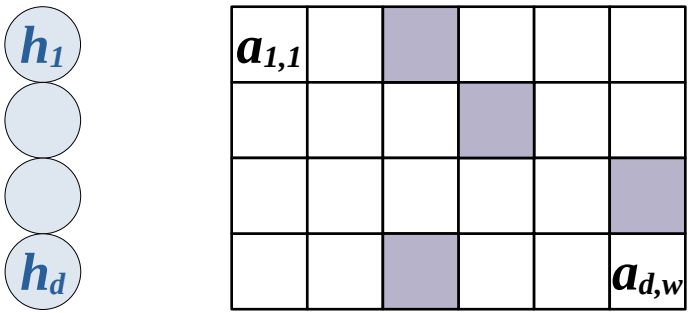
$$Q(i) := \min\{a_{j,h\_j(i)} \mid 1 \le j \le d\}$$



Point Queries

(non-negative case)

$$Q(i) := \min\{a_{j,h\_j(i)} \mid 1 \le j \le d\}$$

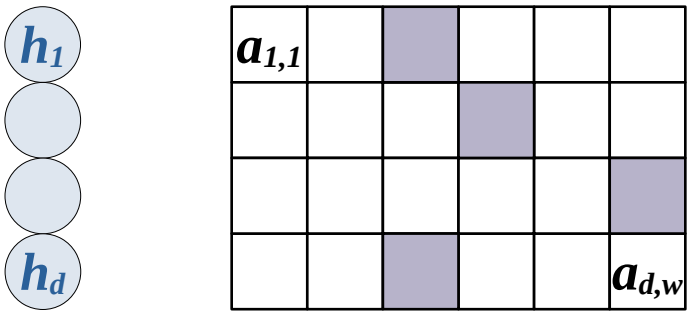


- With $w := \lceil e/\varepsilon \rceil$ and $d := \lceil \ln 1/\delta \rceil$, the minimum query satisfies:

Point Queries

(non-negative case)

$$Q(i) := \min\{a_{j,h\_j(i)} \mid 1 \le j \le d\}$$



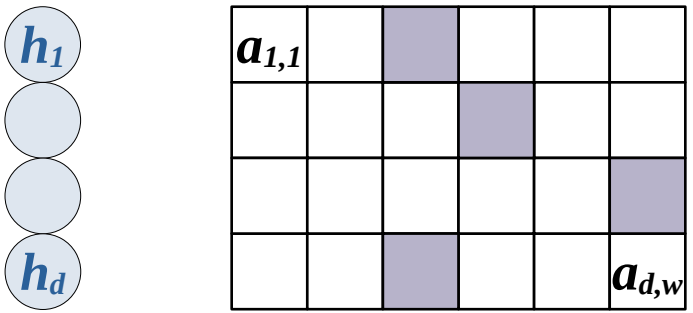
- With $w := \lceil e/\varepsilon \rceil$ and $d := \lceil \ln 1/\delta \rceil$, the minimum query satisfies:

  1) $f_i \le Q(i)$ (with f the actual frequency vector that we estimate)

Point Queries

(non-negative case)

$$Q(i) := \min\{a_{j,h\_j(i)} \mid 1 \le j \le d\}$$

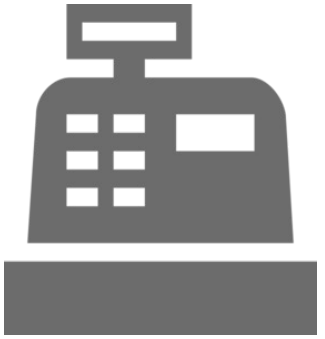


- With $w := \lceil e/\varepsilon \rceil$ and $d := \lceil \ln 1/\delta \rceil$, the minimum query satisfies:

  1) $f_i \le Q(i)$ (with f the actual frequency vector that we estimate)

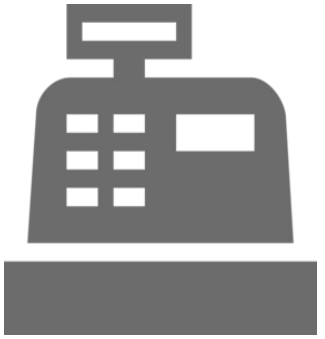
  2) $Q(i) \le f_i + \varepsilon ||f||_1$ with probability at least $1-\delta$

Heavy Hitters in the Cash Register



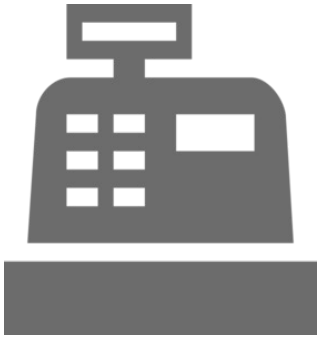
- In the *cash register case*, counts only ever increase

Heavy Hitters in the Cash Register



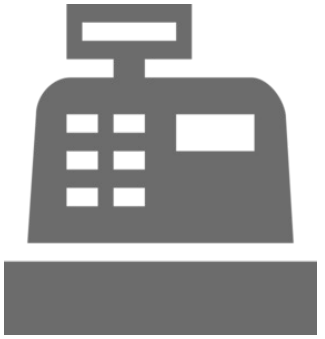
- In the *cash register case*, counts only ever increase
- A ϕ-heavy hitter is an item i with $f_i \geq \phi ||f||_1$

Heavy Hitters in the Cash Register



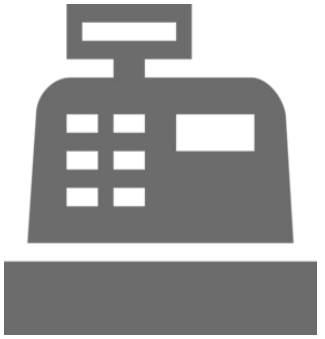
- In the *cash register case*, counts only ever increase
- A ϕ-heavy hitter is an item i with $f_i \geq \phi ||f||_1$
- Maintain $||f||_1$ under updates and heavy hitters in a heap H

Heavy Hitters in the Cash Register



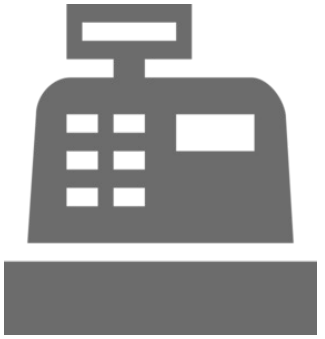
- In the *cash register case*, counts only ever increase
- A ϕ-heavy hitter is an item i with $f_i \geq \phi||f||_1$
- Maintain $||f||_1$ under updates and heavy hitters in a heap H
- After pair (i, c) has been counted in the sketch
  - If $Q(i) \geq \phi||f||_1$, insert i into H

Heavy Hitters in the Cash Register



- In the *cash register case*, counts only ever increase
- A ϕ-heavy hitter is an item i with $f_i \geq \phi||f||_1$
- Maintain $||f||_1$ under updates and heavy hitters in a heap H
- After pair (i, c) has been counted in the sketch
  - If $Q(i) \geq \phi||f||_1$, insert i into H
- If the item i' in H with the lowest count falls below threshold, remove i' from H

Heavy Hitters in the Cash Register



- In the *cash register case*, counts only ever increase
- A ϕ-heavy hitter is an item i with $f_i \geq \phi||f||_1$
- Maintain $||f||_1$ under updates and heavy hitters in a heap H
- After pair (i, c) has been counted in the sketch
  - If $Q(i) \geq \phi||f||_1$, insert i into H
- If the item i' in H with the lowest count falls below threshold, remove i' from H

1) H contains all items i with frequency $f_i \geq \phi||f||_1$

2) With probability $1-\delta$, H contains no items i' with $f_{i'} \leq (\phi-\varepsilon)||f||_1$

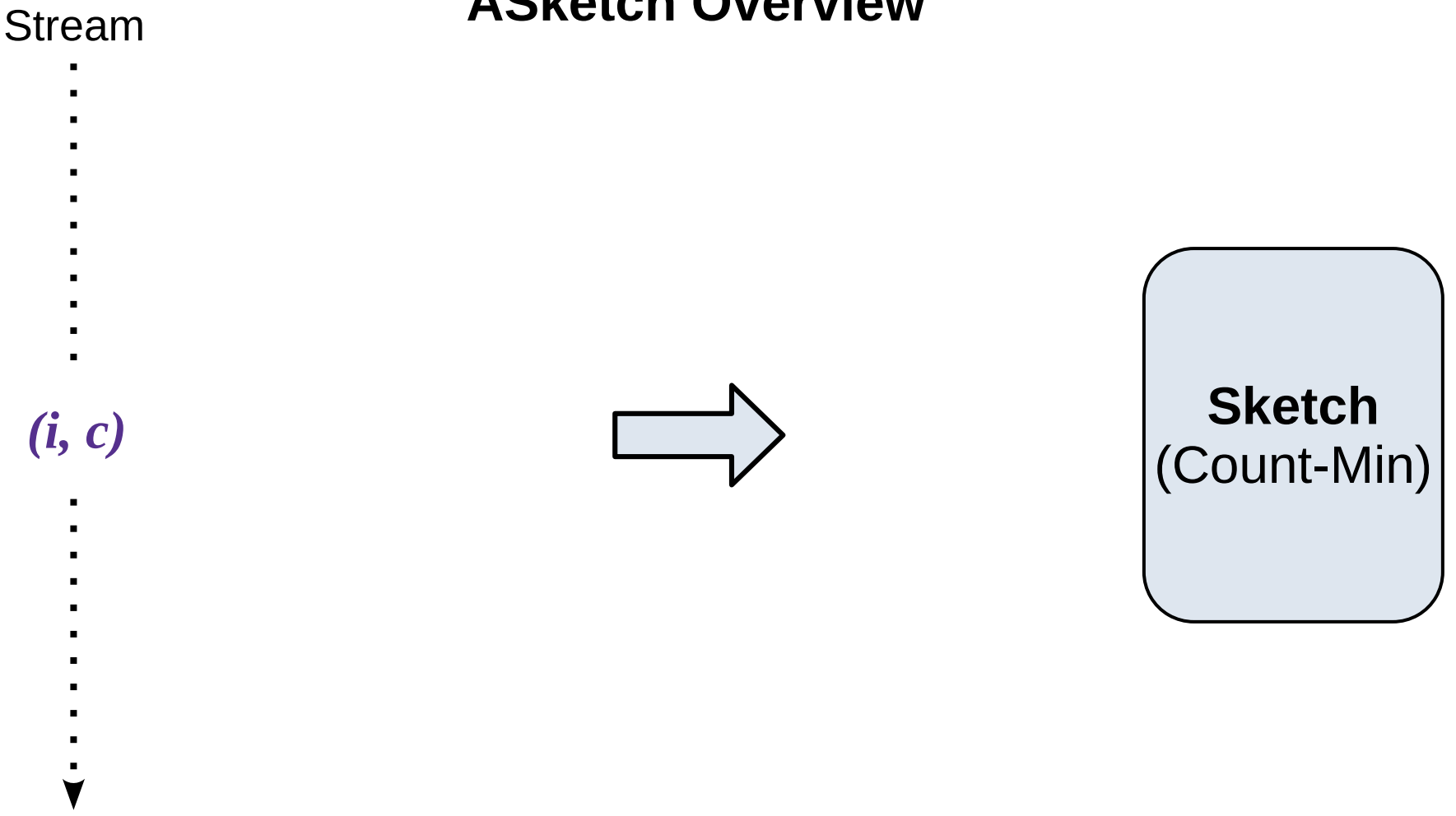
Motivation for Augmented Sketch

- Errors due to frequency over-estimation

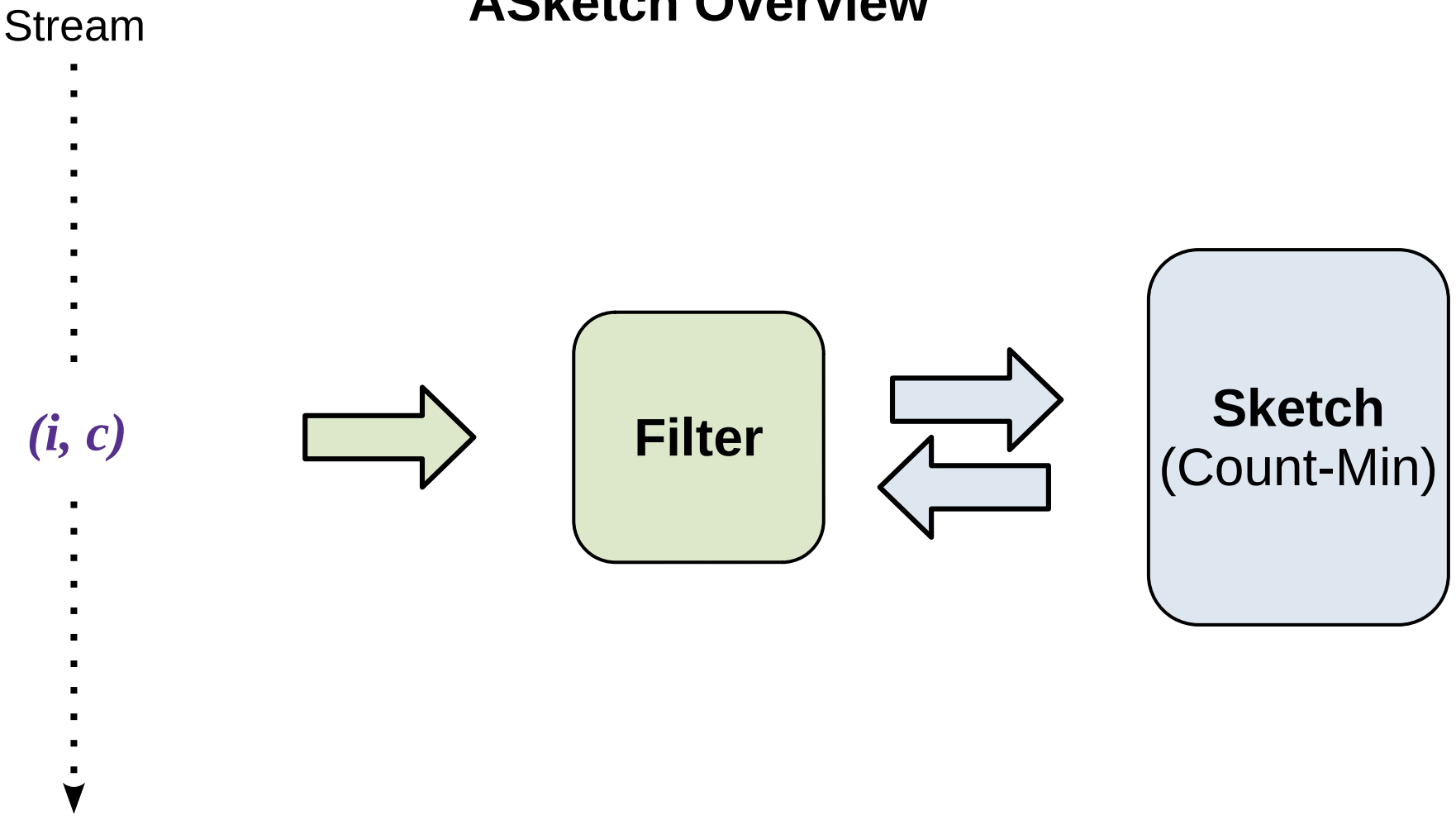
Motivation for Augmented Sketch

- Errors due to frequency over-estimation
  - Count-Min: 0.0024% → **ASketch**: 0.0004%
- False positives
  - Count-Min: few → **ASketch**: (practically) none
- Performance
  - **ASketch** throughput is 4-5 times higher than Count-Min

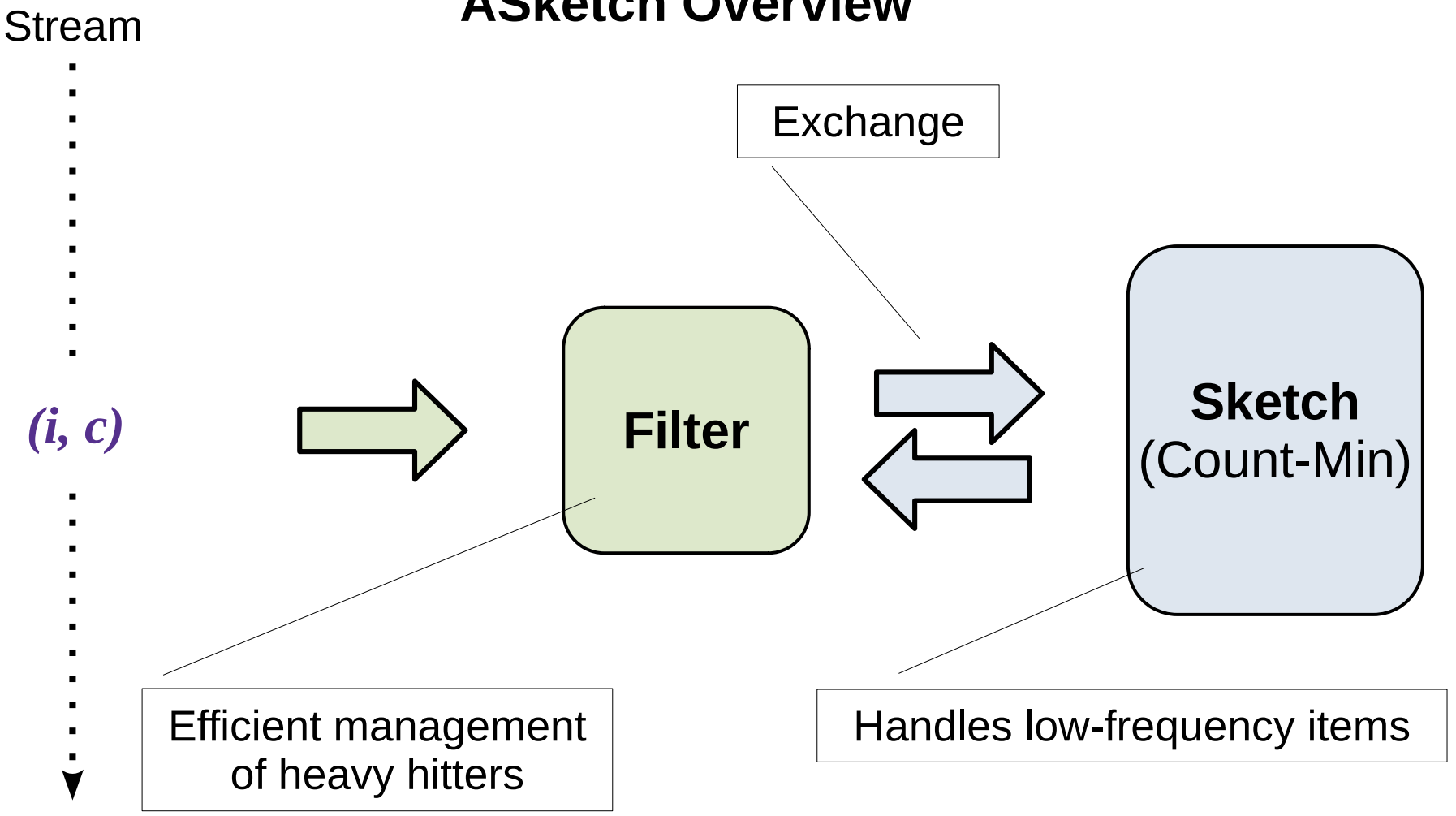
ASketch Overview



ASketch Overview



ASketch Overview



Filter/Sketch Exchange

- Removing items from a sketch is *problematic*

Filter/Sketch Exchange

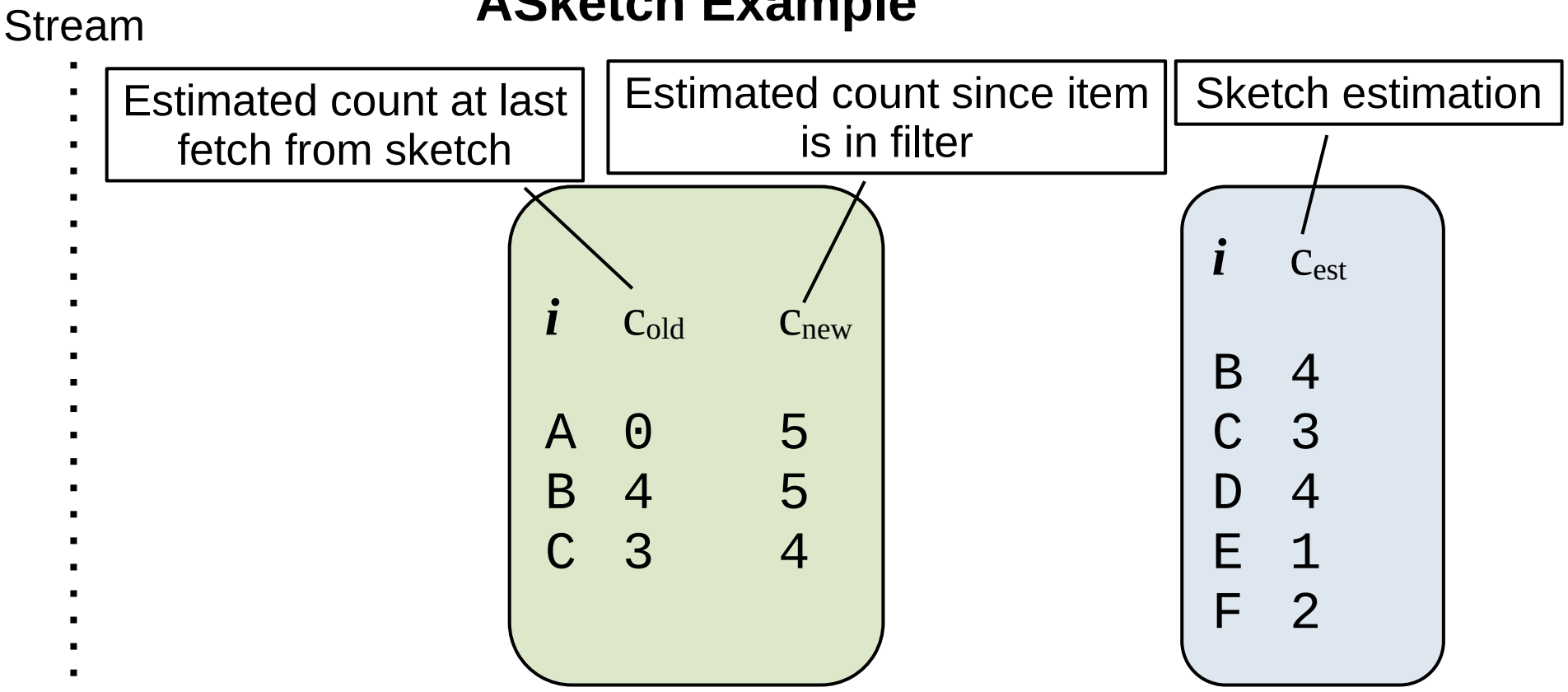
- Removing items from a sketch is *problematic*
- Frequency over-estimation avoids *false negatives* and must be retained

Filter/Sketch Exchange

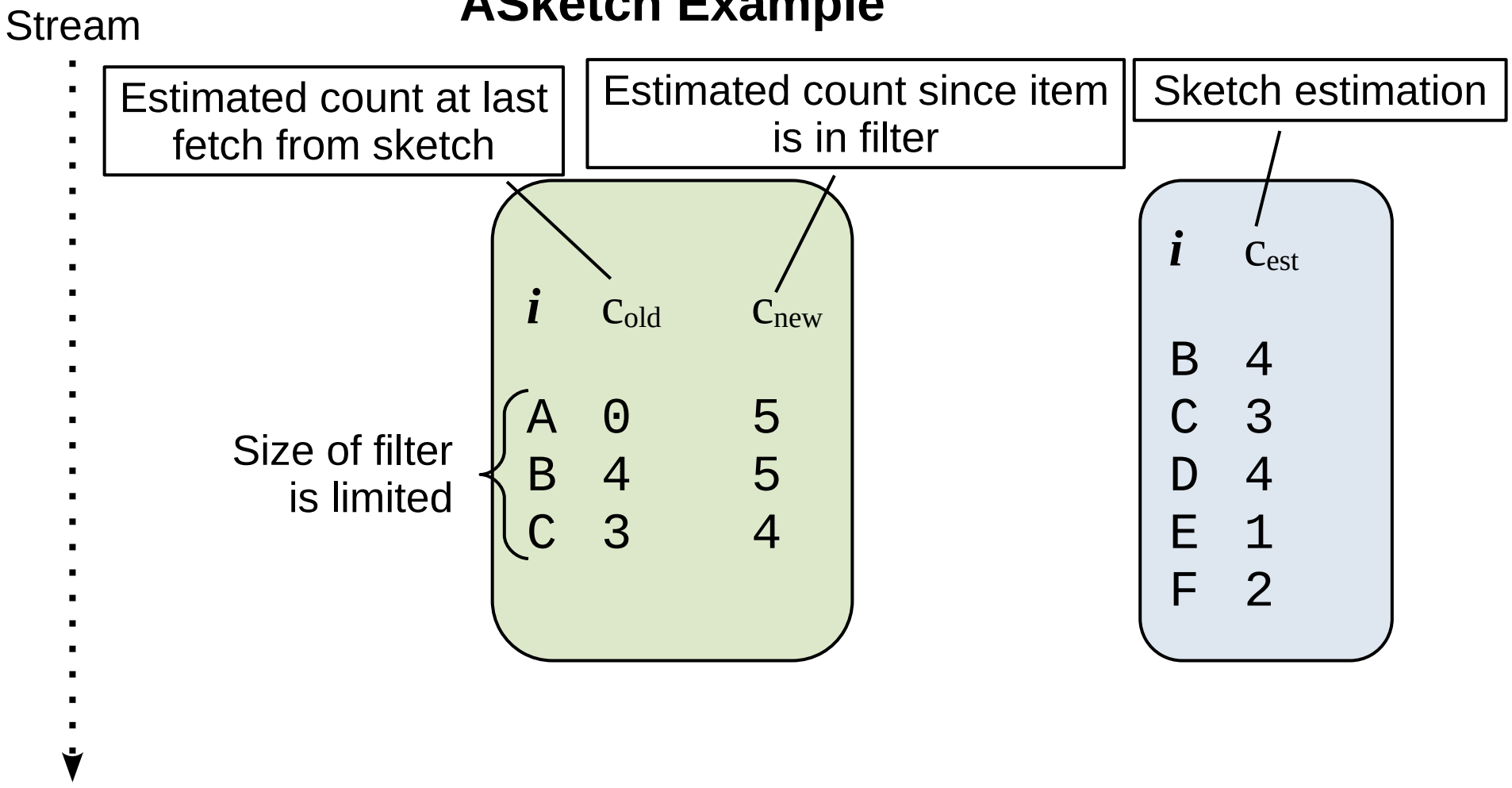
- Removing items from a sketch is *problematic*
- Frequency over-estimation avoids *false negatives*

  and must be retained
- Additional small error for low-frequency items is tolerable

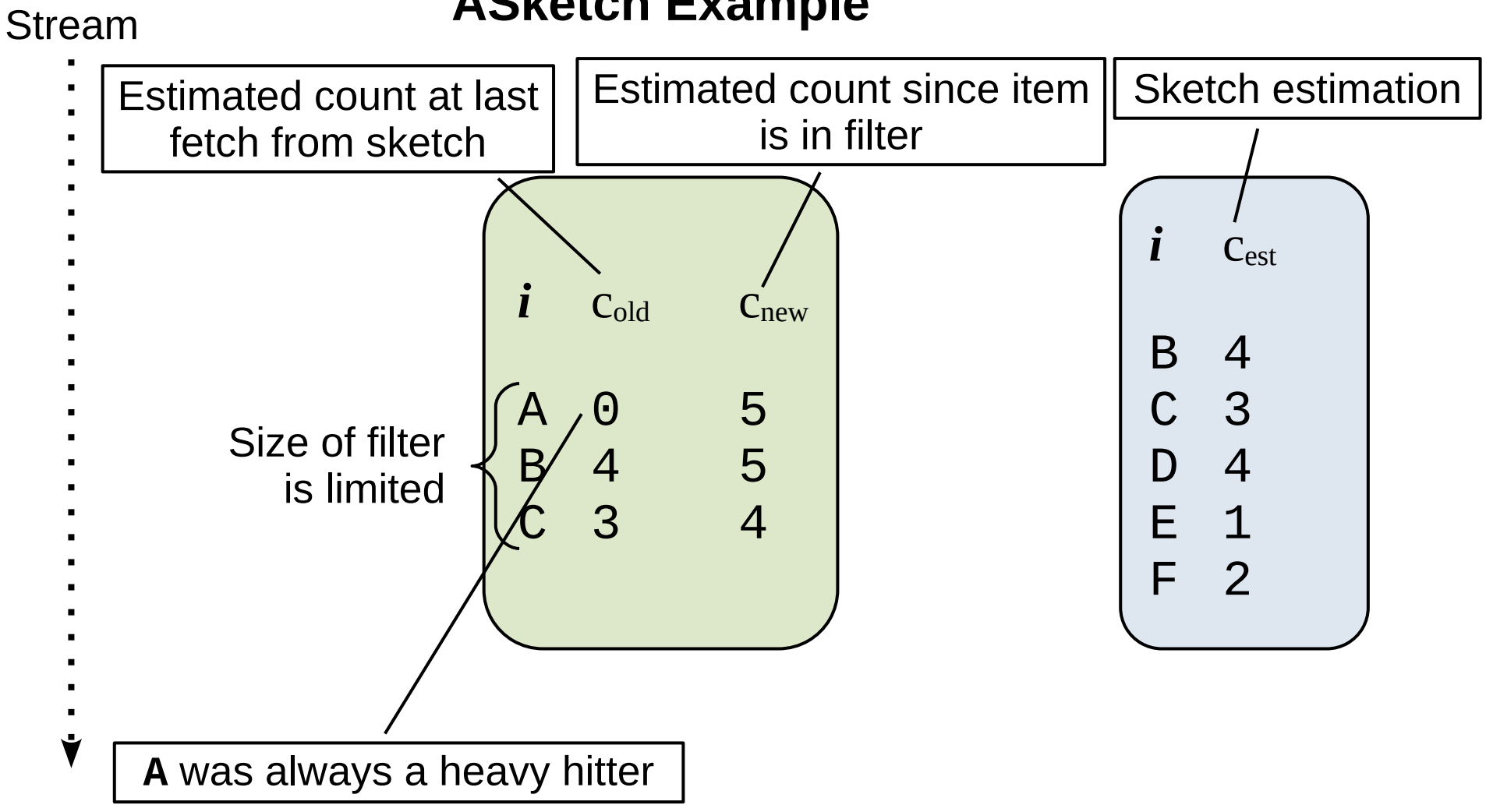
ASketch Example



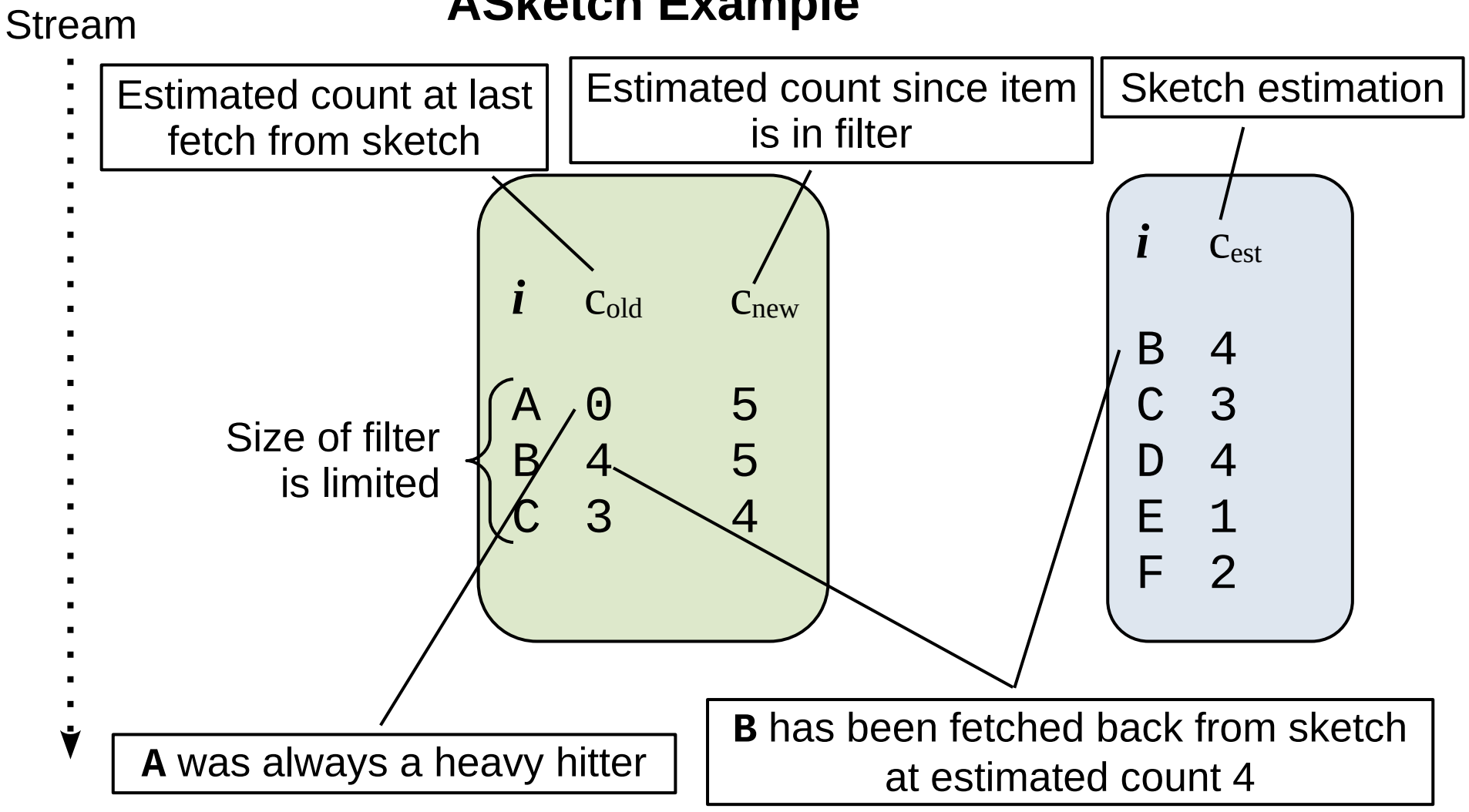
ASketch Example



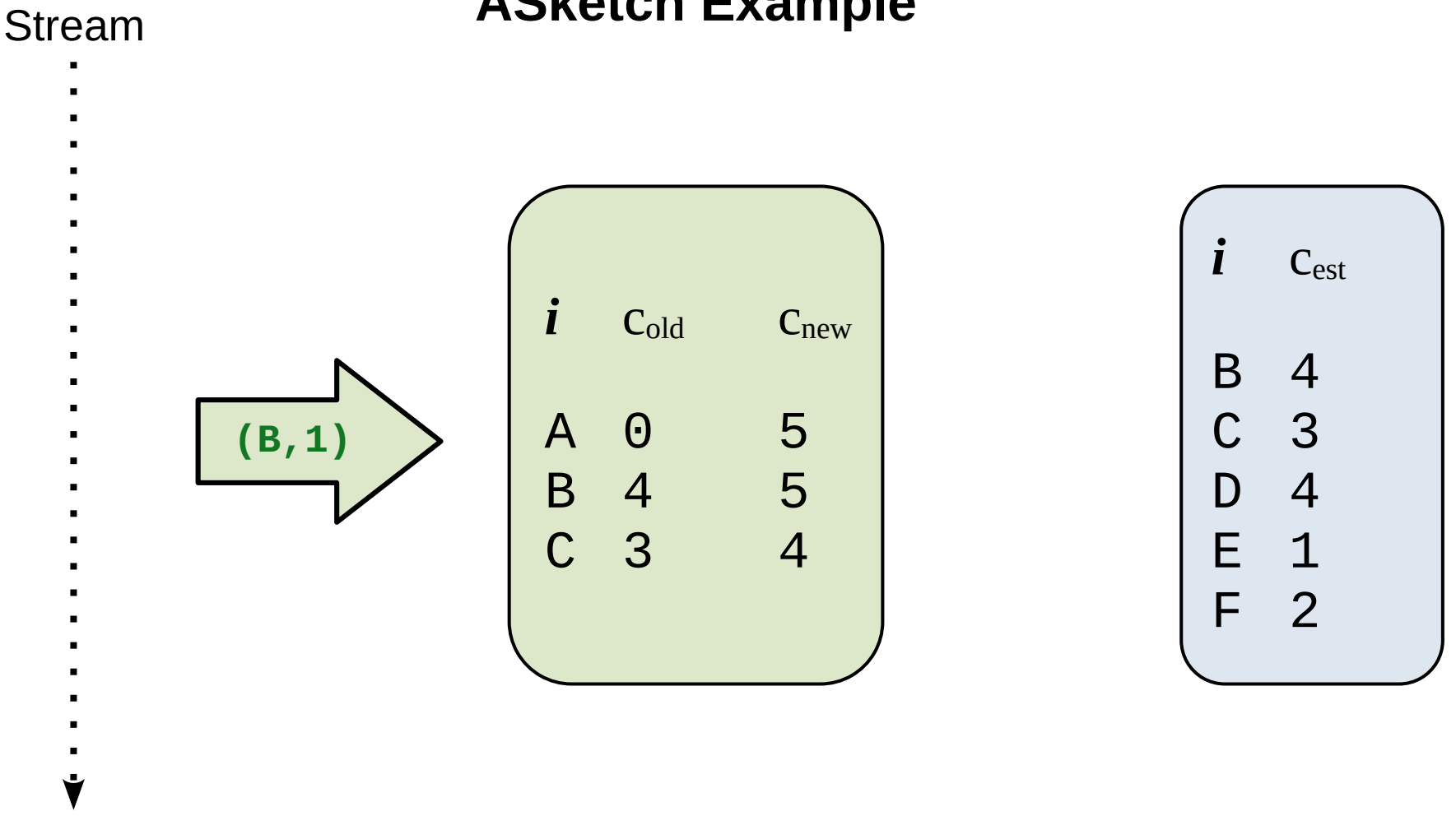
ASketch Example



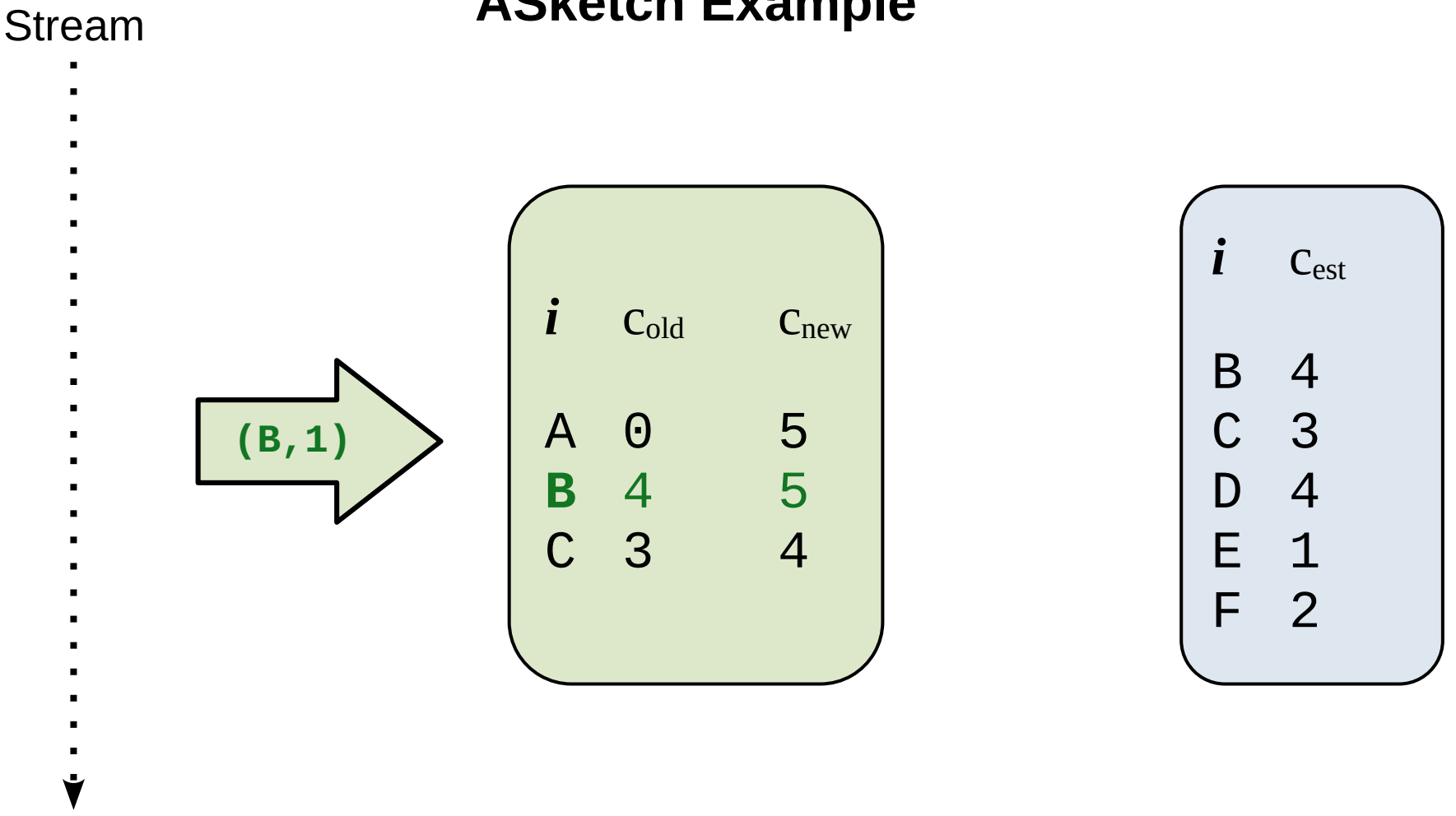
ASketch Example



ASketch Example

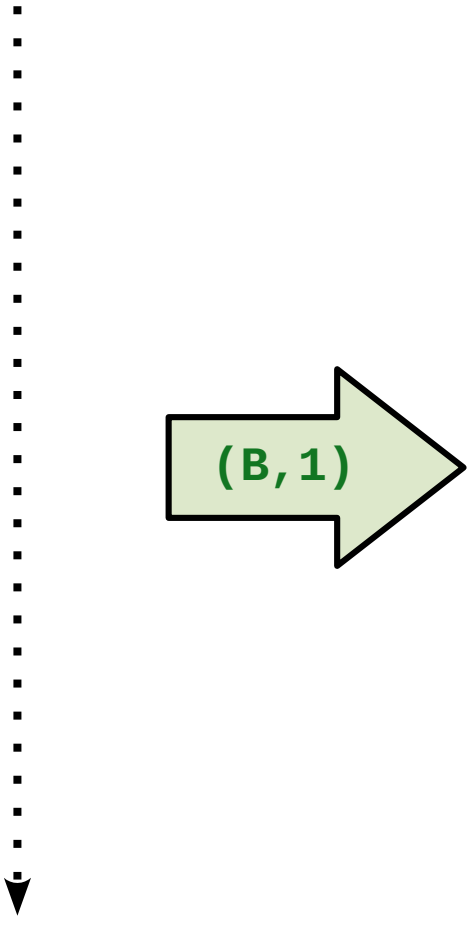


ASketch Example



ASketch Example

Stream

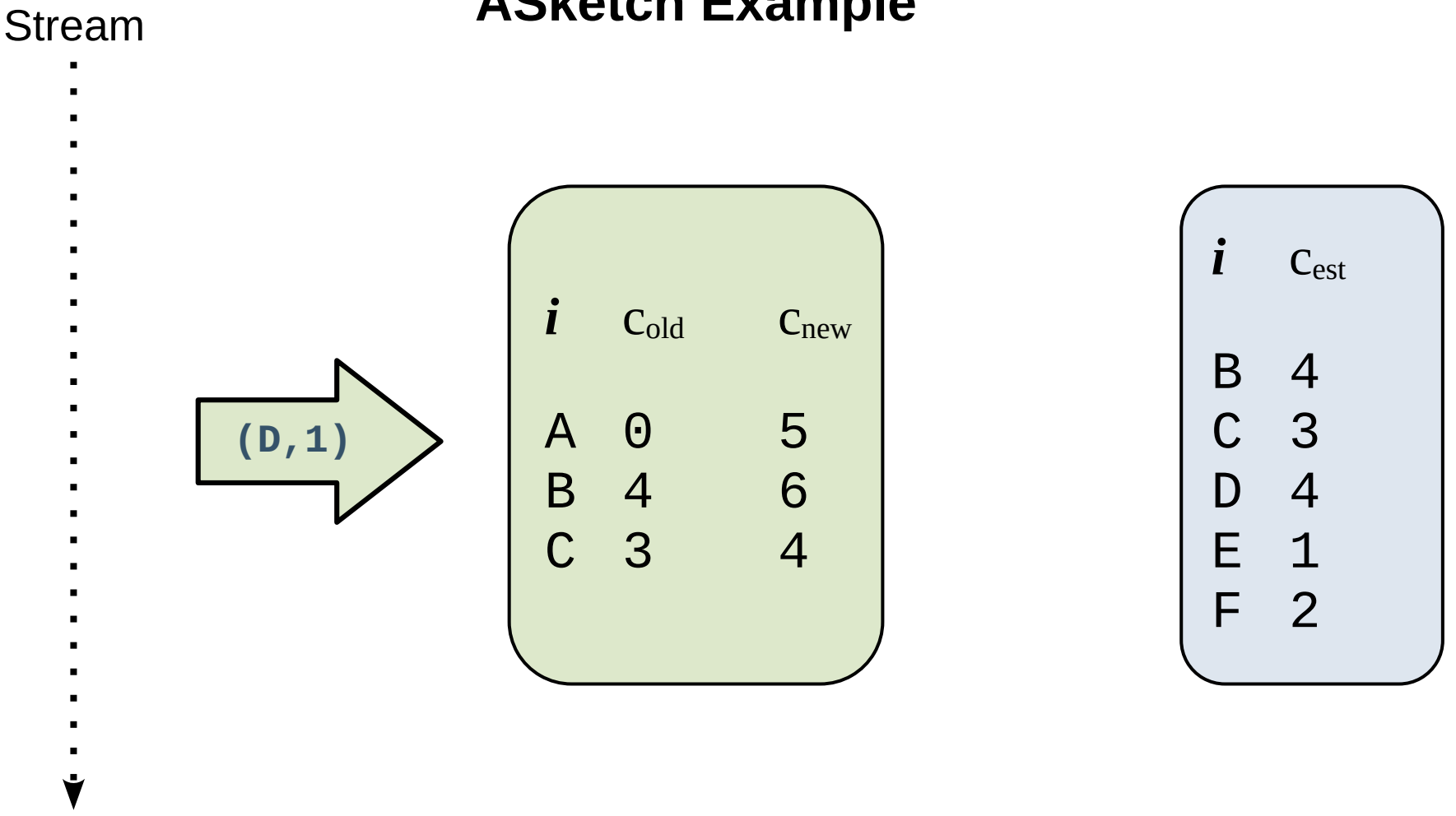


(B,1)

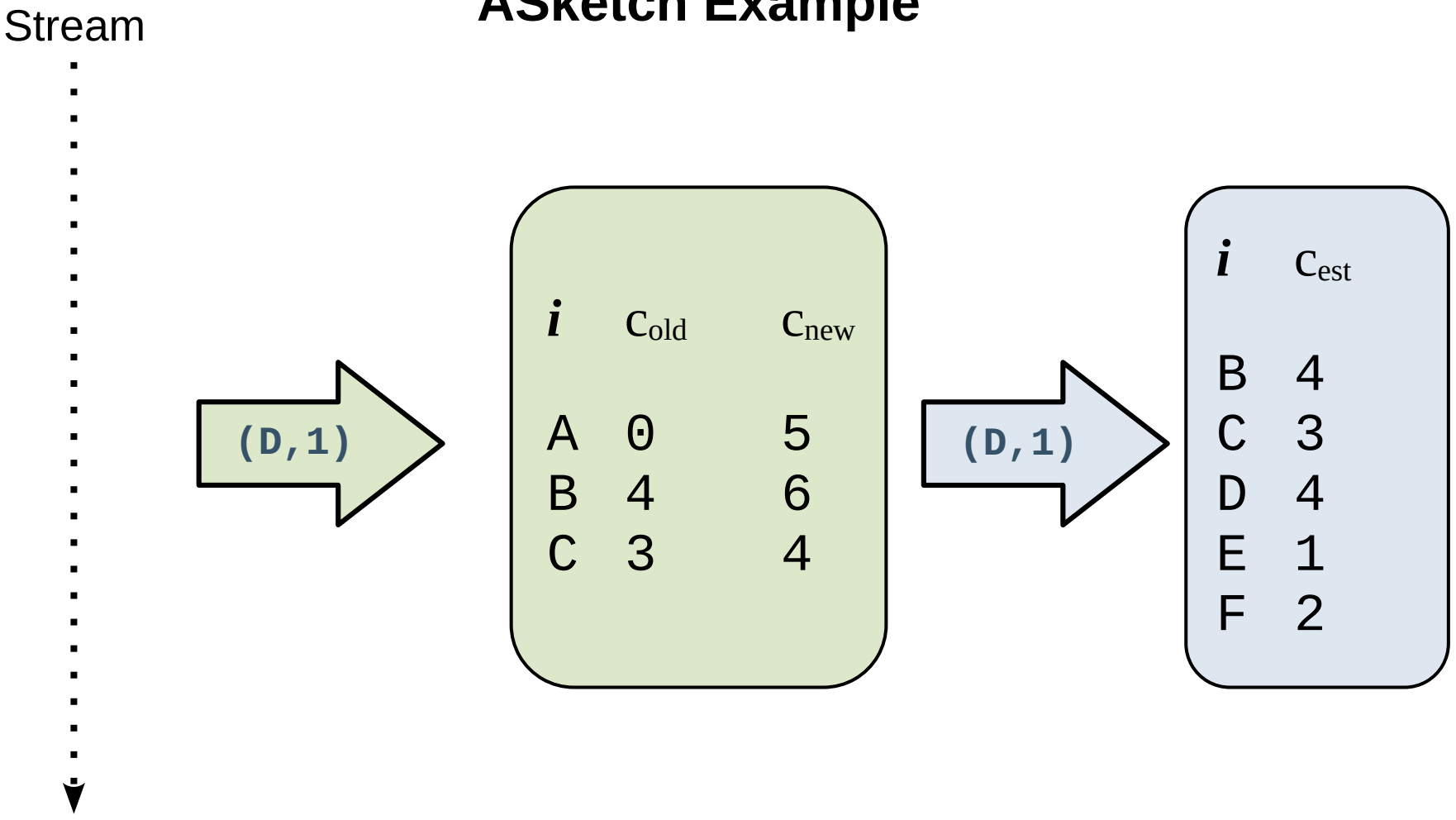
i	C_{old}	C_{new}
A	0	5
B	4	**6**
C	3	4

i	C_{est}
B	4
C	3
D	4
E	1
F	2

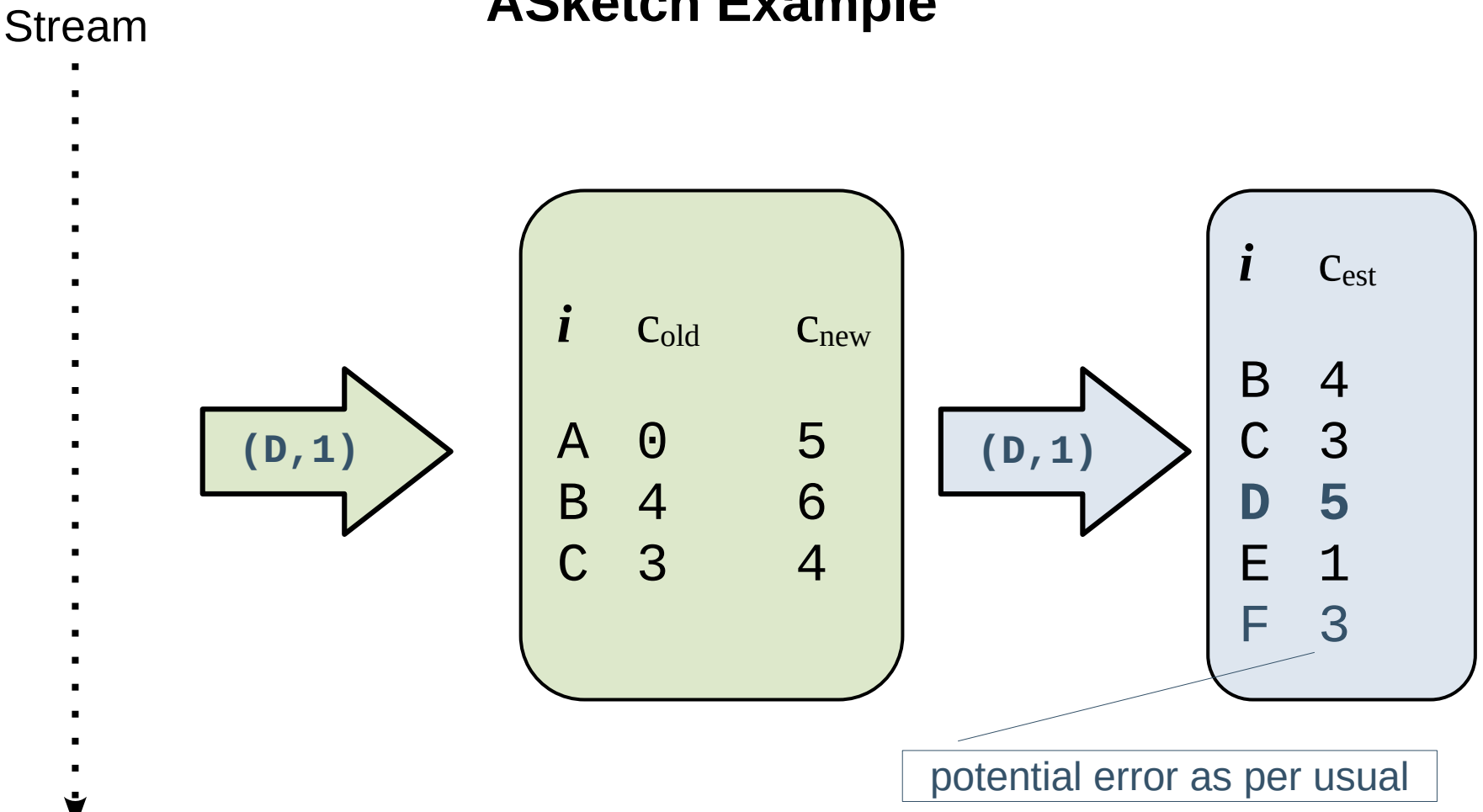
ASketch Example



ASketch Example

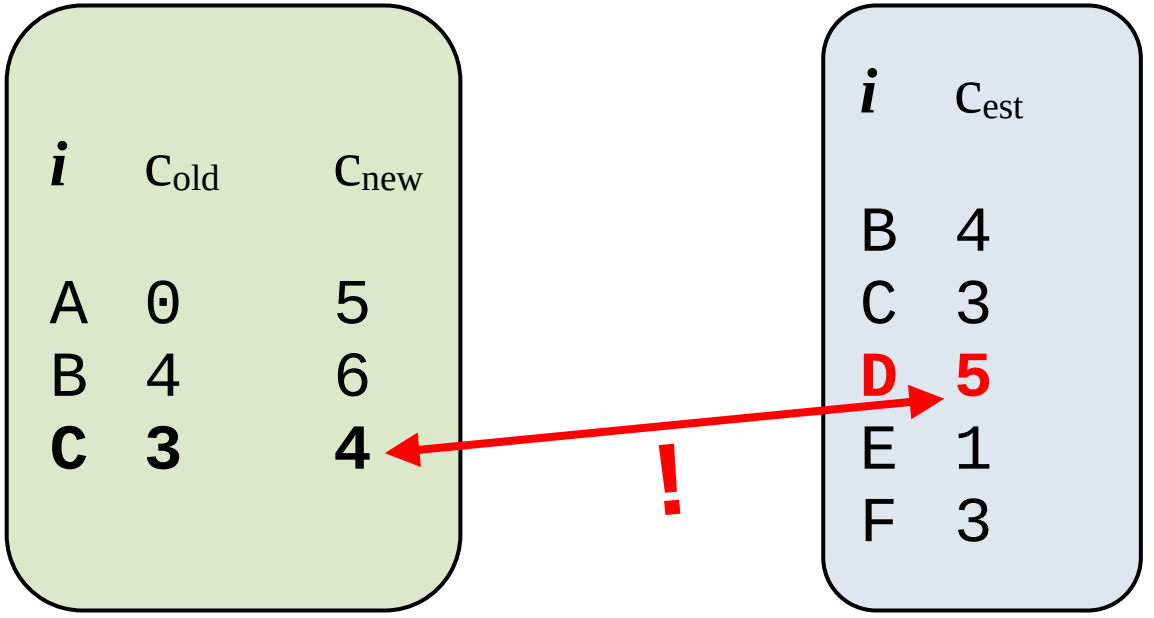


ASketch Example



ASketch Example

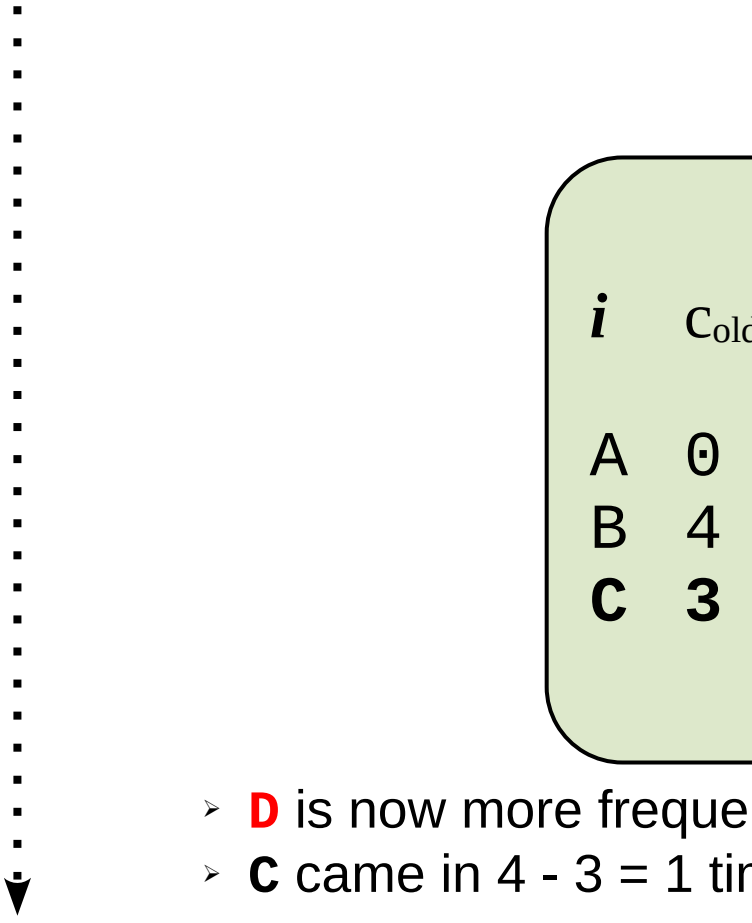
Stream

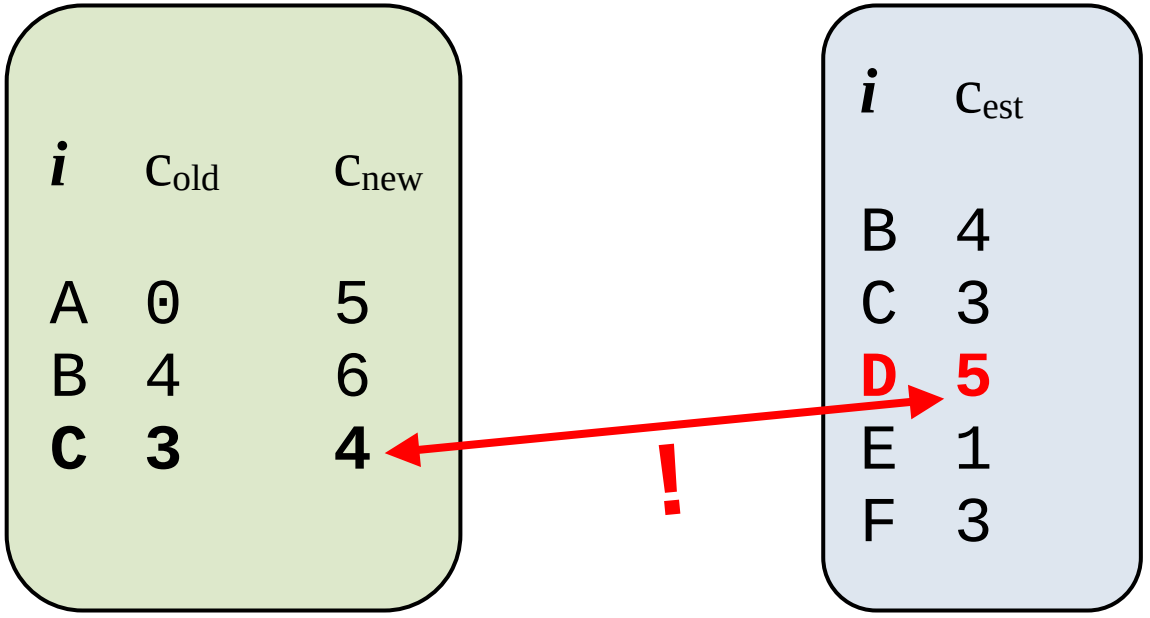


- **D** is now more frequent than current heavy hitter **C**

ASketch Example

Stream

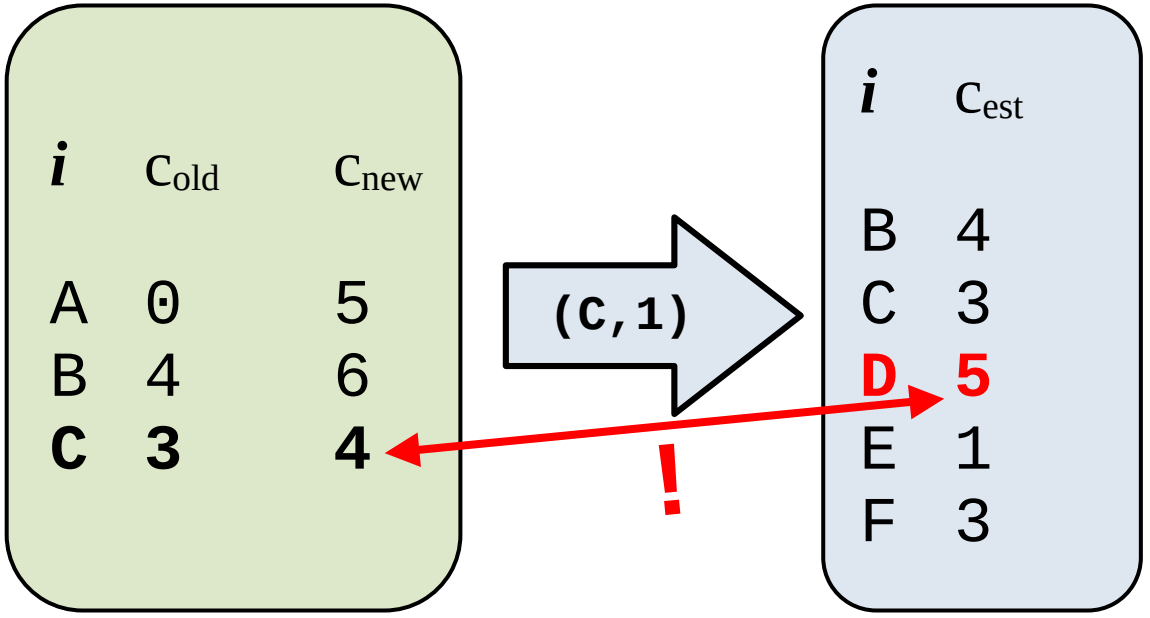




- **D** is now more frequent than current heavy hitter **C**
- **C** came in 4 - 3 = 1 times since last fetched from the sketch

ASketch Example

Stream



- **D** is now more frequent than current heavy hitter **C**
- **C** came in 4 - 3 = 1 times since last fetched from the sketch
- Update the sketch accordingly

ASketch Example

Stream

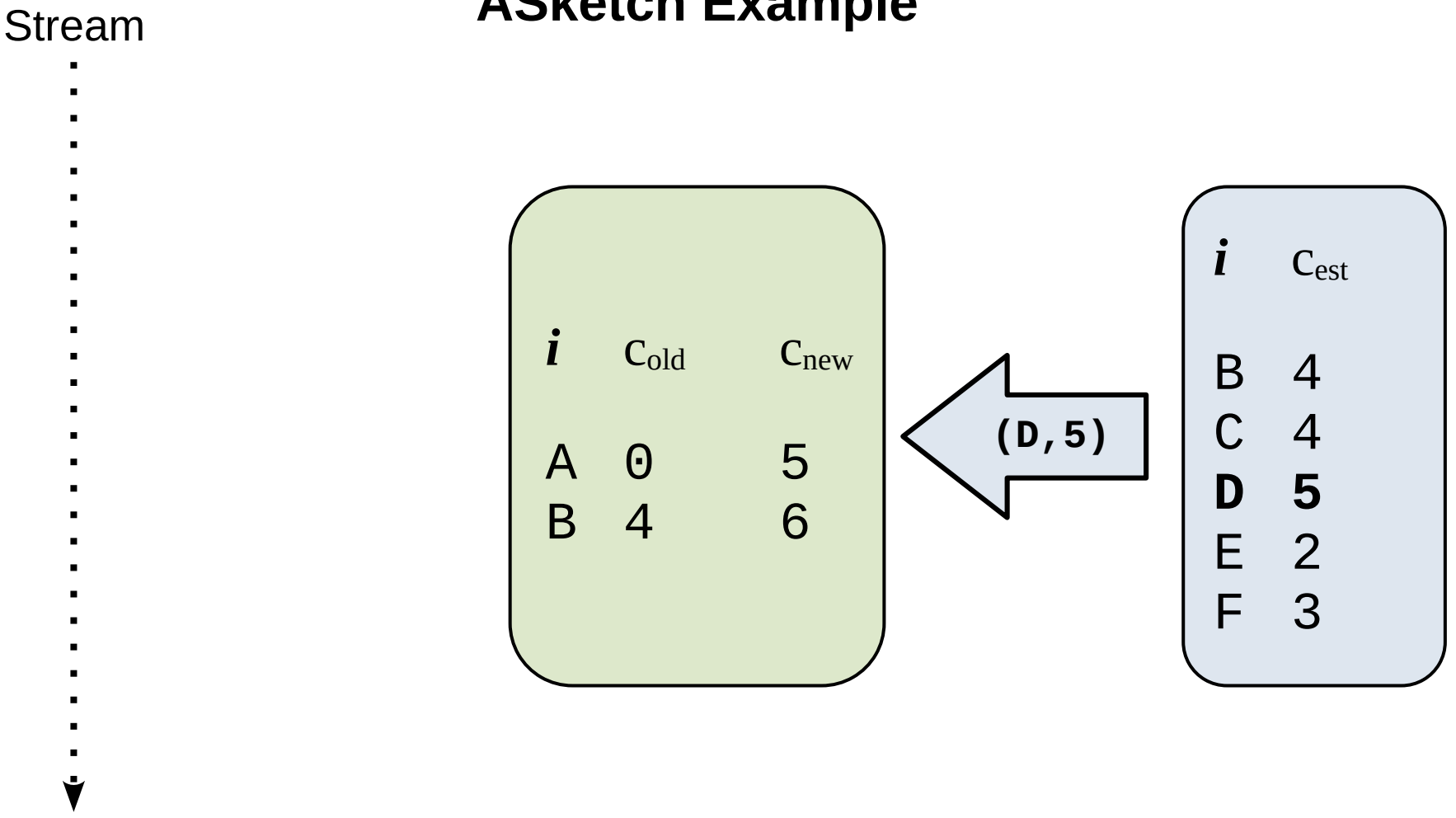
i	C_{old}	C_{new}
A	0	5
B	4	6

(C,1)

i	C_{est}
B	4
C	**4**
D	**5**
E	**2**
F	3

- **D** is now more frequent than current heavy hitter **C**
- **C** came in 4 - 3 = 1 times since last fetched from the sketch
- Update the sketch accordingly
- **This may introduce additional error in the sketch!**

ASketch Example



ASketch Example

Stream

i	C_{old}	C_{new}
A	0	5
B	4	6
D	5	5

i	C_{est}
B	4
C	4
D	5
E	2
F	3

ASketch Example

Stream

i	C_{old}	C_{new}
A	0	5
B	4	6
D	5	5

i	C_{est}
B	4
C	4
D	5
E	2
F	3

- Filter stays small: more frequent items are processed faster

ASketch Example

Stream

i	C_{old}	C_{new}
A	0	5
B	4	6
D	5	5

i	C_{est}
B	4
C	4
D	5
E	2
F	3

- Filter stays small: more frequent items are processed faster
- ASketch is suitable for *pipeline parallelism*