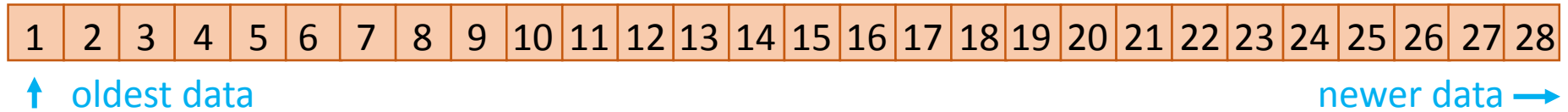# GNU Radio History: Example

Let's say that `history()` is `3`. The block reads in the following items from the upstream buffer:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |

↑ oldest data   newer data ➡

The scheduler allows the data input buffer to be processed piecewise in a series of calls to the work function. Let's say the first 3 work calls give `noutput_items` values of `10`, `8` and `10`, respectively.

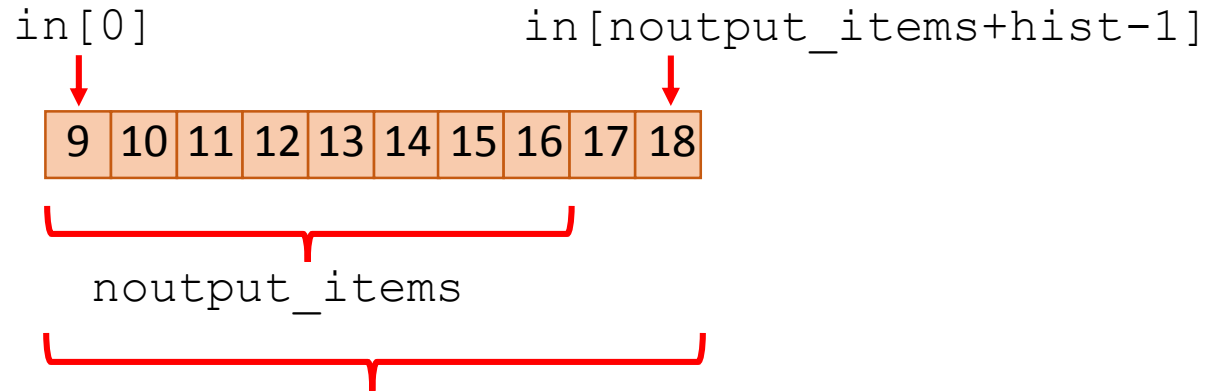## 1st work call

`noutput_items = 10`

`in[0]`   `in[noutput_items+hist-1]`

| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
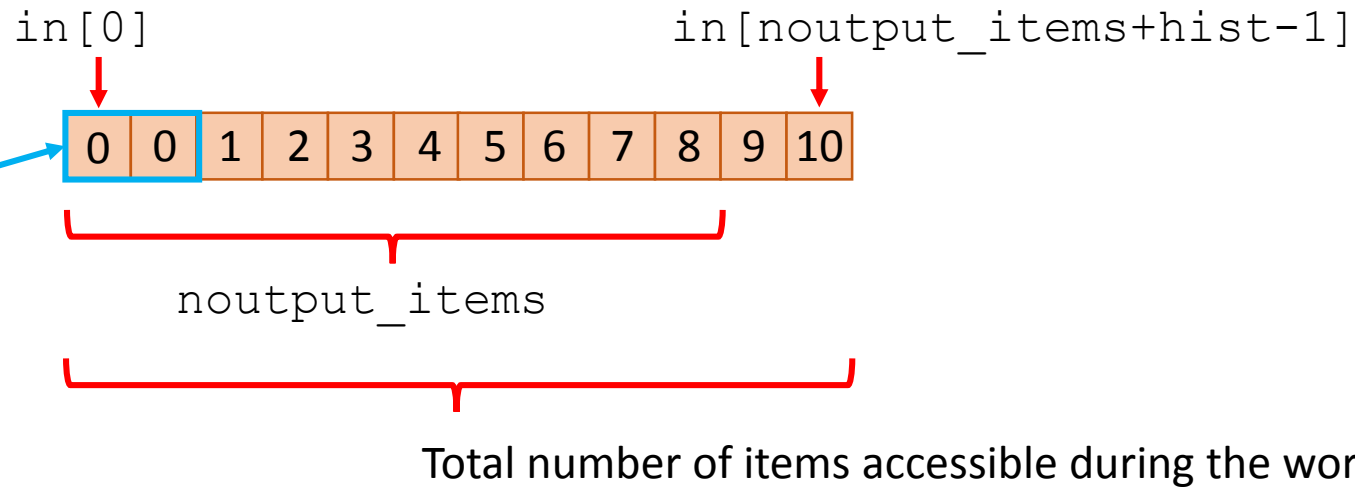
setting history > 1 prepends `history()-1` zeros to the available items in the first work call

`noutput_items`

Total number of items accessible during the work call

GNU Radio History: Example

2nd work call

noutput_items = 8

in[0]   in[noutput_items+hist-1]

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |

noutput_items

Total number of items accessible during the work call

3rd work call

noutput_items = 10

in[0]   in[noutput_items+hist-1]

| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |

noutput_items
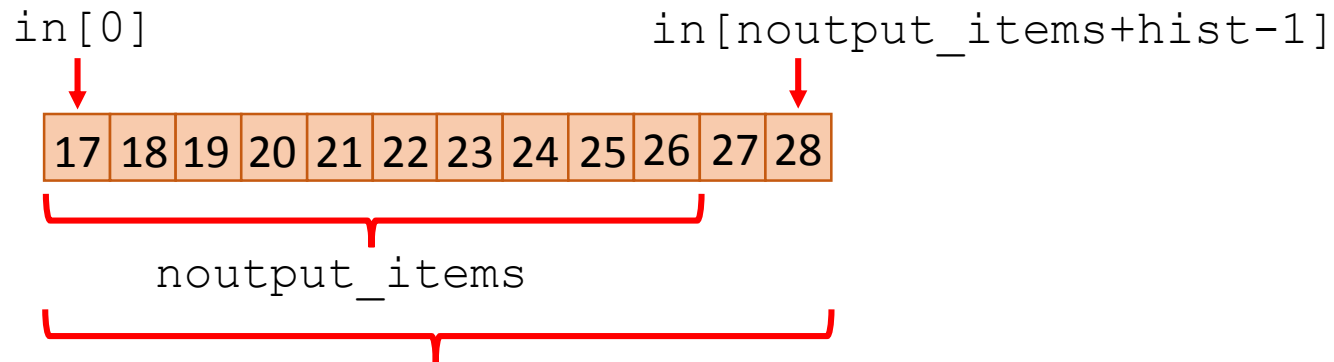
Total number of items accessible during the work call

Summary

Setting history to n increases the number of items available to the work function in any given work call by $n-1$. This is done by giving $n-1$ zero-value items at the beginning of the first work call.

History of length n means:

- The $n-1$ items *beyond* `noutput_items` can be processed in the work call
- Block latency of $n-1$ of items
- Tags can be set in the first `noutput_items` giving the appearance of tagging backward in time