Ruby master - Feature #16987

Enumerator::Lazy vs Array methods

06/26/2020 02:57 PM - zverok (Victor Shepelev)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Enumerations are designed to be greedy (immediately executed on each method call within a chain) by default. Sometimes, that is not useful for practical purposes (e.g. 2 mln strings array, drop comments, split into fields, find the first ten whose field 2 is equal to some value). So one needs to either do everything in one each block, or use Enumerable#lazy. There are three problems with the latter:

1. It is much less known,
2. It is said to be almost always slower than non-lazy, and is therefore not recommended,
3. It lacks some methods that are often necessary in processing large data chunks.

I want to discuss (3) here. Enumerator::Lazy would better, but actually doesn't, have methods such as: #flatten, #product, and #compact. They are all methods of Array, not Enumerable. In fact,

1. They probably should belong to Enumerable (none of them requires anything besides #each to function),
2. They are definitely useful for lazily processing large sequences.

Related issues:

Related to Ruby master - Feature #17312: New methods in Enumerable and Enumer...	**Open**

History

#1 - 06/26/2020 05:18 PM - sawa (Tsuyoshi Sawada)

- Description updated

#2 - 06/26/2020 05:21 PM - sawa (Tsuyoshi Sawada)

- Description updated

#3 - 07/15/2020 04:56 PM - midnight (Sarun R)

I used Lazy all the time. There is nothing to be done here about its popularity.
FWIW, People knew about it, but choose not to rely on it because they want to support old versions of Ruby.
Hence, it is not very popular in open-source settings.

Regardless of what should be implemented, for now, you can use

```
lazy.flat_map(&:itself)
```

as #flatten, and

```
lazy.select(&:itself)
```

as #compact.

Only #product is the tricky one that requires multiple operations, but it is not used very often anyway.

What I missed most is #scan.
https://ramdajs.com/docs/#scan
It is basically a #reduce that yield at every iteration.

#4 - 11/20/2020 08:32 AM - matz (Yukihiro Matsumoto)

- Related to Feature #17312: New methods in Enumerable and Enumerator::Lazy: flatten, product, compact added