I frequently find myself needing to determine if two arrays intersect but not actually caring about the intersection, so I write code like:

```
(a1 & a2).any?
```

It would be nice to have an `intersect?` convenience method on `Array` to perform this query.

ruby#1972: Add Array#intersect?

**Related issues:**
- Related to Ruby master - Feature #16928: Array#include_all? & Array#include_any? [Open]
- Has duplicate Ruby master - Feature #15976: Add Array#overlap? for whether th... [Closed]

**Associated revisions**
- Revision b6bb4623 - 04/16/2021 07:12 AM - nobu (Nobuyoshi Nakada)
  NEWS for [Feature #15198] [ci skip]

**History**

#1 - 10/04/2018 02:28 AM - shyouhei (Shyouhei Urabe)

- Interesting. Can you share a bit more detail about your "if two arrays intersect but not actually caring about the intersection" use case? For instance if you have open sourced such code, a URL for it helps us a lot.
- If you have zero interest to the intersection itself I think you can avoid creating the temporary array.

#2 - 10/05/2018 02:01 PM - c4am95 (Travis Hunter)

The most recent example I encountered was authorizing a user in a Rails endpoint. Each user has a list of abilities, and each endpoint has a list of abilities that is authorized to perform the action. We just need to check if there is an intersection between the two lists.

I updated the PR to avoid creating the intermediate array.

#3 - 10/06/2018 05:30 PM - shevegen (Robert A. Heiler)

Demonstrated use cases helps the core team & matz assess on the usefulness of a proposed change, which can help in accepting issue requests ultimately in the long run (if the use case is considered sufficiently useful). :)

#4 - 10/07/2018 11:19 PM - c4am95 (Travis Hunter)

I threw together an example in a gist which is very similar to the use case I described. I have also run into numerous other situations where this functionality would have been useful.

I added some comments in the gist but I'll cross-post them here for reference:

The current behavior creates an intermediate array for both intersection tests when the resulting array is clearly not needed. With the desired behavior, we could avoid creating the intermediate array and produce a faster best case runtime.

#5 - 10/08/2018 05:10 PM - c4am95 (Travis Hunter)

It also seems like it comes up fairly commonly on stackoverflow/blogs:
Has duplicate Feature #15976: Add Array#overlap? for whether the intersection of 2 arrays is non empty? added

It might make sense to use ary1.to_set.intersect?(ary2). That way it makes explicit the fact that ary1 must be converted to a set. But Set#intersect? would have to support any Enumerable.

I tried setting that up here: https://github.com/ruby/ruby/pull/2598

I ignored infinite ranges for now

- Related to Feature #16928: Array#include_all? & Array#include_any? added

Matz/Marc-Andre: anything I can do to help here as far as documentation or providing more use cases? I/my colleagues would love to see this make it into Ruby.

I am positive on the feature as has valid use cases, it can be optimized in C and as a bonus adds no cognitive load.

I believe we just need a final "yay" from Matz. I'll add it to the next meeting's agenda.

Accepted.

Matz.

We discussed this in today's developer meeting. Here are some remarks:

- It takes one argument for now.
- The implementation will internally use a hash to match the behavior of Array#&.

Status changed from Open to Closed

Applied in changeset gilb6bb4623ebcf1335cf0a81d279cde24a9e97bce.

NEWS for Feature #15198 [ci skip]