DS 210 Discussion 5 - Mini Project 1

Mini Project 1: Build your own Vector!

https://rust4ds.github.io/ds210-sp26-a1/projects/project01_vec.html

The goal of today's discussion is to get you through *Part 1, Step 0 & 1.*

1 Some logistics

Only group of **size 2** is allowed, no single or larger than 2!

Both of the members in the group must be **in the same discussion section**!

- So you can work together and we will check your project in groups during later discussion time.

2 Part 1: SlowVec

Carefully check the course website for details of steps! These below are only some supplementary/hints.

2.1 Step 0: Set up your GitHub

Our code repo again: https://github.com/rust4ds/ds210-sp26-a1-code.

2.2 Collaboration

1. **Student 1**:
   1. Fork/sync your codes to our code repo.
   2. Add **student 2** as collaborator to you fork with write (by default for personal account).
   3. Clone the fork.
   4. You **must** implement `push()` function and
   5. Push your solution to a branch called `std1`.
2. **Student 2**:
   1. Be a collaborator in **student 1**'s fork.
   2. Clone **student 1**'s fork.
   3. You **must** implement `remove()` function and
   4. Push your solution to a remote branch called `std2`.
      - You can create a local branch named `std2`: `git switch -c std2`
   5. **After both of you have finished,** merge the branch `std1` to `std2`.
      - You can do that by first `git fetch --all` all remote branches, `git merge origin/std1` to your local `std2` branch, and do `git push -u origin std2` again to push merged branch to `std2`.
   6. For submission:
      - Create a branch named `submission1` from `std2` (before deadline).
      - Submit the link of github repo to gradescope with **student 1** as collaborator.

2.3 Step 1

First, run the main function inside that file using VSCode or using the following command:

```
cd project_1_vec/slow_vec
cargo run --bin main
```

Note: If you open the `slow_vec` folder in VSCode as instructed, then you do not need to `cd`. Just run the `cargo run --bin main`

You should see the following outputs before implementation:

```
slow_vec_basics------
SlowVec(Fixed[10, 20, 30])
Element at 0 is 10
Element at 1 is 20
Element at 2 is 30
---------------------

fixed_sized_array----
Fixed[10, 20]
10
10
10
---------------------

slow_vec_push--------

thread 'main' (21444) panicked at C:\Users\Maro\Documents\ds210-sp26-a1-code\project_1_vector\slow_vec\src\lib.rs:63:9:
not yet implemented: Student 1 should implement this
note: run with `RUST_BACKTRACE=1` environment variable to display a backtrace
error: process didn't exit successfully: `C:\Users\Maro\Documents\ds210-sp26-a1-code\project_1_vector\target\debug\main.exe` (exit code: 101)
```

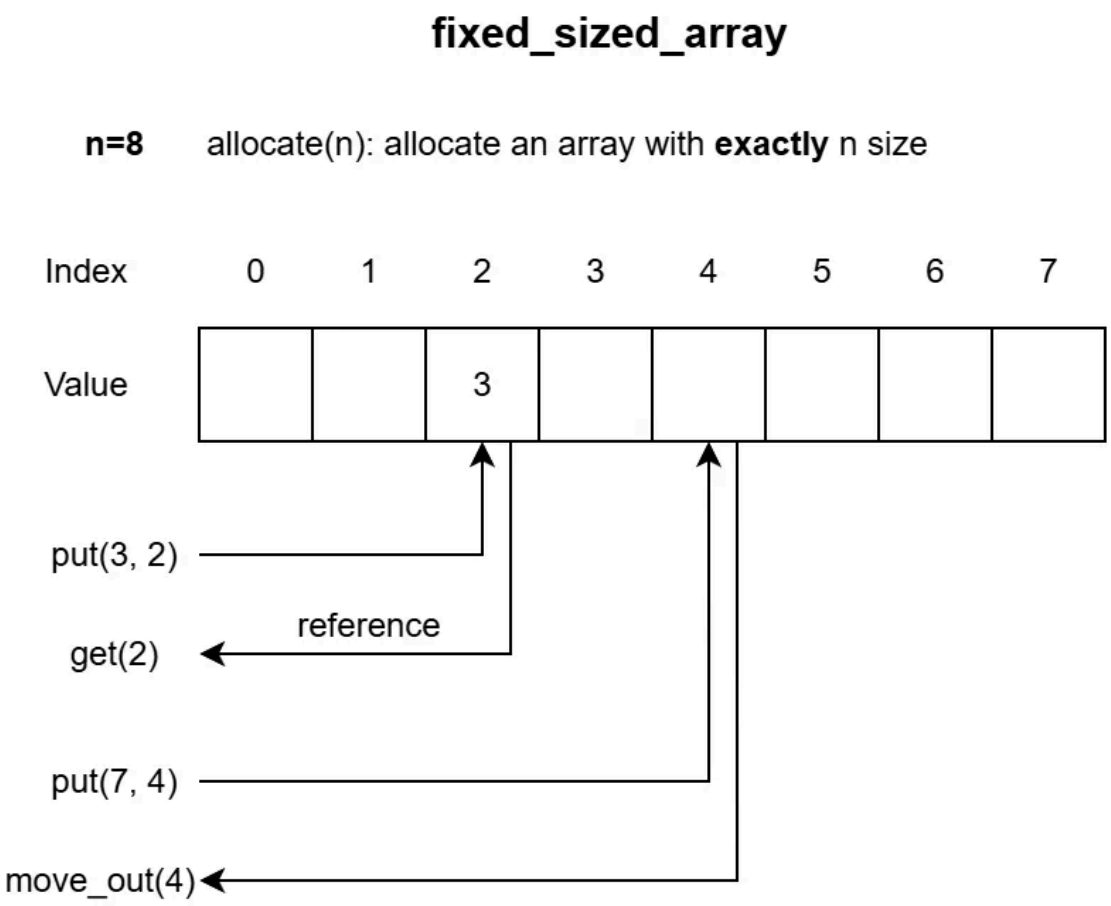
2.3.1 Implementation

- For `push(t)`: copy the values from old `fixed_sized_array` **with an additional new `t`**.
- For `remove(i)`: copy the values from old `fixed_sized_array` **without the element positioned `i`**.

Notes:

- Using a for-loop is enough - it's "Slow"-Vec!
- Think carefully of using `get()` or `move_out()` - do you need a reference or value?
  - Our SlowVec does not specify `clone()` traits so you cannot use `.clone()`. Please don't modify the definition of classes either.

3 Figures


fixed_sized_array
n=8
allocate(n): allocate an array with **exactly** n size
Index
0
1
2
3
4
5
6
7
Value
3
put(3, 2)
reference
get(2)
put(7, 4)
move_out(4)