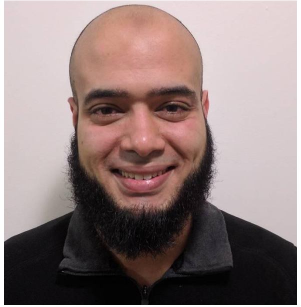
Data Structures

Vector Homework

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Problem #1: Right rotation

- Consider our Vector class. Add the member function: void right_rotate()
- The function shifts every element 1 step towards the right.
  - What about the rightmost element? It goes to the first idx
- Example
- Assume the array content is: 0 1 2 3 4
- After a right rotation it will be: 4 0 1 2 3
  - Notice how, in this case, the '4' has been rotated to the head of the array!
- Ensure you avoid expanding the array's capacity

Problem #2: Left rotation

- Consider our Vector class. Add the member function: void left_rotate()
- The function rotates the whole array 1 step to the left
    - However, in this case, the leftmost element will be 'rotated' around to the back of the array!
- Example
- Assume the array content is: 0 1 2 3 4
- After a left rotation, it will be: 1 2 3 4 0
    - Notice how the 0 has 'rotated' to the tail of the array after applying left_rotate()
- Ensure you avoid expanding the array's capacity

Problem #3: Right rotation with steps

- Implement void right_rotate(int times)
- This one applies the right rotation times time
- Assume array content is: 0 1 2 3 4
- right_rotate(2) ⇒ it will be: 3 4 0 1 2
- The challenge: times can be up to: 2000000000
- Your code should be efficient to some extent

Problem #4: Deleting a position

- Implement method int pop(int idx) in the vector class
- It returns the deleted value
- Remove this element from the array
- Assume array is: 4 0 **1** 2 3
- pop(2)
    - Return value 1
    - New array: 4 0 2 3

Problem #5: Improved search

- Assume our vector is huge and we do many find() calls for almost a few small repetitive values
- One way to improve the code speed is: each time you find the value, you **shift** it one step to the **left**
- Eventually, the values that are queried a lot, will move to the head of array
- Implement method: int find_transposition(int value)
  - It returns the found position, but consider moving it one step to the left
- Example: 10 20 30 40 50. find_transposition(3)
  - New array 10 30 20 40 50
  - Return 1

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."