NYU SHANGHAI 上海纽约大学

CSCI-SHU 210 Data Structures 100 Points

Homework Assignment 4 - Dynamic Array

Problem 1 – Ring Buffer - 25 Points

You are tasked with implementing a ring buffer, which is a data structure that uses a fixed-size array to manage a queue of elements efficiently. The buffer should support the following operations:

1. **enqueue(item)**: Adds an item to the end of the buffer. If the buffer is full, the oldest item should be removed to make space for the new item.

2. **dequeue()**: Removes and returns the oldest item from the buffer. If the buffer is empty, return None.

3. **is_empty()**: Returns True if the buffer is empty, otherwise False.

4. **is_full()**: Returns True if the buffer is full, otherwise False.

5. **resize()**: Resizes the buffer to double its capacity if its size is greater than half or reduces the buffer size if its current length is smaller or equal to one-quarter of its capacity. The minimum buffer capacity is 5.

You should implement the ring buffer using the UserDefinedDynamicArray class. Your implementation has to extend the UserDefinedDynamicArray class. You can not modify the UserDefinedDynamicArray class.

NYU SHANGHAI 上海 纽约大学

Problem 2 – Intersection and Union- 25 Points

Part A 7.5 Points

Given two strictly increasing lists, L1 and L2, of length n, write a Python function to compute L1 ∩ L2 using only basic Python operations.

Requirements

- You cannot use Numpy or import other Python libraries.
- You cannot cast L1 and L2 to sets or use Python sets.
- You must use basic Python functions and lists.
- You don't have to sort your results.
- Your function has to be O(n) time complex

Part B 7.5 Points

Given two strictly increasing lists, L1 and L2, of length n, write a Python function to compute L1 ∪ L2 using only basic Python operations.

Requirements

- You cannot use Numpy or import other Python libraries.
- You cannot cast L1 and L2 to sets or use Python sets.
- You have to use basic Python functions and lists.
- You don't have to sort your results.
- Your function has to be O(n) time complex

Part C 10 Points

Solve the problem of parts A and B using the dynamic array class from Recitation 4. Introduce the function $intersect(self,\ b)$ and $union(self,\ b)$ as new member functions in the dynamic array class.

Requirements

- You cannot use Numpy or import other Python libraries.
- You cannot cast b to a set or use Python sets.
- You have to use basic Python functions and lists.
- You don't have to sort your results.
- You cannot use Python lists
- Your function has to be O(n) time complex

NYU SHANGHAI 上海纽约大学

Problem 3 – Mirror and Rotate Matrices - 50 Points

Part A - 15 Points

Implement the Python function $mirror(M)$ which mirrors the provided nxn matrix M vertically.

Example $mirror(M)$

Input: [[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]]

Output: [[5, 4, 3, 2, 1],
[10, 9, 8, 7, 6],
[15, 14, 13, 12, 11],
[20, 19, 18, 17, 16],
[25, 24, 23, 22, 21]]

Part B - 15 Points

Implement the Python function $rotator(M,\ a,\ d)$ which rotates the provided nxn matrix M by a given angle a in the specified direction d.

Important

- $a\ \in \{\ 0,\ 90\ , 180\}$
- $d\ \in \{clockwise,\ anticlockwise\}$

Requirements

- You cannot change the input parameters.
- You have to return a new matrix.

Example $rotator(M,\ 90,\ "anticlockwise")$

Input: [[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]]

Output: [[5, 10, 15, 20, 25],
[4, 9, 14, 19, 24],
[3, 8, 13, 18, 23],
[2, 7, 12, 17, 22],
[1, 6, 11, 16, 21]]

Part C - 20 Points

Implement a Python function to solve the problem of *Part B* in place.

Requirements

- Your solution has to be completed in place. Compute and return your result in M.
- You are only allowed to use a constant amount of extra space.