

Course: Data Structures (CSE CS203A, 114-1)

Quiz II: Array, Linked List, Stack and Queue

October 21, 2025, 16:30~17:00

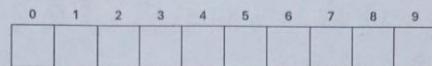
Student ID: 1131411

Student Name: 魏允鴻

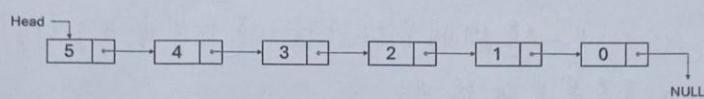
86

Data Structures: Visualization

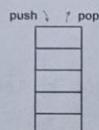
(1) Array



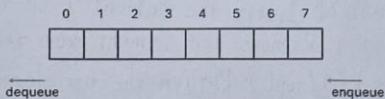
(2) Linked List



(3) Stack



(4) Queue



Q1: (30 pts; 10 pts for each) Describe the mechanism of the function

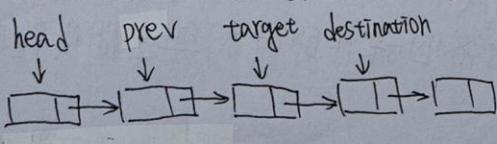
MoveTo(node *head, node *target, node*destination)

A1: Write a short paragraph explaining how the **MoveTo** function works (you may answer in English or Mandarin).

① Are there any additional variables required? If so, explain why they are necessary.

~~MoveTo()這個函式在做的事就是將 target 指向的節點移至 destination 指向的節點之後，如果可以創立一個新變數: node *prev 去指向 target 的前一個節點，這樣能幫助我們完成這項任務，我們先將 prev 指向與 head 相同節點，再利用 while 迴圈從頭開始尋找 prev 的正確位置。~~

② Draw a visualization of the singly linked list to support your explanation.



~~prev->next = target->next;
target->next = destination->next;
destination->next = target;~~

③ Is there any variation of a linked list (e.g., doubly linked list or circular linked list) that can simplify or

~~improve this operation? 使用 doubly linked list (雙向鏈結串列) 可以簡化操作，因為每個節點都有 prev 指標，可以直接取得前一個節點，不必從頭搜尋哪個才是 target 的前一個節點。~~

Q2: (40 pts, 10 pts for each) Definition of Data Structures

Define the following data structures and list their fundamental operations.

-4

A2:

① Definition of "Stack"

Stack (堆疊) 是一種線性資料結構，遵循後進先出 (LIFO) 原則，最後放入的元素會先被取出。

② Definition of "Queue"

Queue (佇列) 是一種線性資料結構，遵循先進先出 (FIFO) 原則，最早放入的元素最先被移出。

③ Preliminary operations of "Stack"

1. push(x): Insert the element X on the top of the stack.

2. pop(): Remove the element from the top of the stack.

3. peek() / top(): Return the top element. 4. isEmpty(): Check if the stack is empty.

④ Preliminary operations of "Queues"

1. enqueue(x): Insert element at the rear of queue.

2. dequeue(): Remove element from the front of queue.

3. front() / peek(): Return the first element without removing it.

4. isEmpty(): Check if the queue is empty.

Q3: (30 pts) AI Copilot Application

Choose up to two data structures from the visualization list above.

Compose a single prompt (within 300 words) that you would use with an AI Copilot to explore or learn advanced concepts related to your chosen data structures.

A3:

我想學習 Stack 和 Queue 的進階概念，請用 C++ 或 Python 示範他們的實作方式、真實應用 (例如 Stack 用於運算式求值，Queue 用於任務排程)，並展示更進階的版本，例如：Circular Queue、Priority Queue 和 Linked Stack，同時分析各個操作的時間與空間複雜度。

-10