Names: Jules Maurice Uwizeyimana

Dept: Business Information Technology

Reg number: 224014947

Date: 23/09/2025

SOLUTIONS - DATA STRUCTURES & ALGORITHIMS Exercise II

PART I – STACK

A.Basics

Q1. How does the MTN MoMo app example show the LIFO nature of stacks?
Answer:
In a stack, the Last Item that is Pushed (added) is the First to be Popped (removed). In the MoMo app, when you fill in payment details step by step and press the back button, the very last step you entered is the one removed first. This clearly demonstrates the LIFO (Last-In-First-Out) behavior of stacks.

Q2. Why is the back action in UR Canvas similar to popping from a stack?
Answer:
When you navigate modules and press “back,” the most recent page or step disappears. This is the same as a stack operation called Pop, which removes the item on top (the last added). Therefore, pressing back is like popping the latest entry from the stack.

B. Applications

Q3. How could a stack enable the undo function when correcting mistakes?
Answer:
Every action you perform is pushed into a stack. When you make a mistake, the undo feature simply pops the last action from the stack. This way, the system goes back to the earlier correct state. Hence, a stack makes undo operations possible.

Q4. How can stacks ensure forms are correctly balanced?
Answer:
Stacks are used to check balance by pushing every opening bracket (or field) and popping when a matching closing bracket is found. If, at the end, the stack is empty, then everything is properly matched. This logic ensures that registration forms or input fields are correctly balanced without mismatches.

C. Logical

Q5. After the given push and pop sequence, which task is next on top of the stack?
Answer:
Operations: Push("CBE notes"), Push("Math revision"), Push("Debate"), Pop(), Push("Group assignment"). After Pop(), "Debate" is removed. The remaining stack has
[CBE notes, Math revision]. Then "Group assignment" is pushed, making it the new top.
Therefore, the top of the stack is "Group assignment."

Q6. Which answers remain in the stack after undoing 3 recent actions?
Answer:
When a student undoes three actions, the system pops three items from the top of the stack. The answers that were entered earlier (before the last three) remain. This shows that undo removes the latest actions but keeps the previous ones intact.

D. Advanced Thinking

Q7. How does a stack enable retracing process in RwandAir booking forms?
Answer:
Each booking step is pushed onto a stack. When the passenger wants to go back, the steps are popped one by one in reverse order. This ensures retracing is smooth and follows the same order as actions were taken.

Q8. Show how a stack algorithm reverses the proverb "Umwana ni umutware." Answer:
Steps:

- Push each word → ["Umwana", "ni", "umutware"].
- Pop words → first "umutware", then "ni", then "Umwana." The reversed sentence becomes: "umutware ni Umwana."

Q9. Why does a stack suit DFS better than a queue?
Answer:
DFS (Depth First Search) goes deep along one branch before backtracking. A stack naturally supports this because it always takes the last path added and explores it fully. In contrast, a queue would explore level by level, which is BFS, not DFS.

Q10. Suggest a feature using stacks for transaction navigation in BK Mobile app.
Answer:
A stack can allow users to move backward (Pop) and forward (Push again) through transaction history. This would work like undo/redo in text editors, helping users navigate their past financial actions.

PART II —QUEUE

A. Basics

Q1. How does a restaurant line show FIFO behavior?
Answer:
Customers are served in the exact order they arrive. The first person to enter the line is the first person to be served. This matches the queue principle of FIFO (First-In-First-Out).

Q2. Why is a YouTube playlist like a dequeue operation?
Answer:
In a playlist, the first video at the front plays first and then gets removed. The next video moves forward. This is exactly what happens in a queue when we dequeue the front item.

B. Applications

Q3. How is waiting to pay taxes at RRA a real-life queue?
Answer:
People line up to pay taxes one by one. The first to join the line is the first to be served, while the last to join waits until others have finished. This is a direct example of a queue in daily life.

Q4. How do queues improve customer service at MTN/Airtel centers?
Answer:
By serving people in order, queues avoid confusion and unfairness. Each customer knows their position and gets served when their turn arrives, making service more organized and efficient.

C. Logical

Q5. Who is at the front after the given operations in Equity Bank?
Answer:
Sequence: Enqueue(Alice), Enqueue(Eric), Enqueue(Chantal), Dequeue(), Enqueue(Jean). Start: [Alice, Eric, Chantal]. Dequeue removes Alice → [Eric, Chantal]. Enqueue Jean → [Eric, Chantal, Jean]. Front = Eric.

Q6. How does a queue ensure fairness in pension applications (RSSB)?
Answer:
Applications are handled in arrival order. No one can jump the line, and everyone gets served fairly based on who came first. This prevents bias and ensures fairness.

D. Advanced Thinking

Q7. Explain how each queue type maps to Rwandan life.
Answer:

- Linear queue: People lining up at a wedding buffet, moving forward until served. - Circular queue: Buses at Nyabugogo loop back after dropping passengers, creating a cycle.
- Deque: Boarding a bus from both front and back doors → people can enter/exit from either end.

Q8. How can queues model restaurant orders?
Answer:
When customers order food, each order is enqueued. The kitchen prepares them in the same order, and when food is ready, it is dequeued and served.

Q9. Why is CHUK hospital example a priority queue, not normal queue?
Answer:
In emergencies, patients are treated immediately regardless of arrival time. This "jumping the line" feature is unique to a priority queue, unlike a normal FIFO queue.

Q10. How would queues fairly match drivers and students in a moto/e-bike app?
Answer:
Passengers are enqueued when they request a ride. Drivers are also queued by availability. The system dequeues from both queues and pairs the first available driver with the first waiting passenger, ensuring fairness.