

Taller IA: Pilas y Colas

Inversión recursiva de una cola (1.25 puntos)

Escribe una función recursiva que tome una cola como entrada y la reorganice para que quede en orden inverso. Pista: usa `dequeue()` para eliminar el primer elemento, invierte recursivamente la cola, y luego usa `enqueue()` para agregar el primer elemento de nuevo. Además, debes explicar los conceptos subyacentes al problema (recursión y estructuras de cola) y proporcionar un pseudocódigo que ilustre la solución.

multiPop(int k) en una pila en C++ (1.25 puntos)

Añade un método `std::vector<Item> multiPop(int k)` a una clase Pila (Stack) que saque `k` elementos de la pila y los devuelva como un `std::vector<Item>`. Explica los conceptos relacionados con pilas y manejo de múltiples elementos, y proporciona un pseudocódigo que explique el proceso paso a paso.

Implementar una pila usando una cola (1.25 puntos)

Implementa una pila usando una sola cola. Pista: para eliminar un elemento, pasa todos los elementos de la cola uno por uno hasta el final, excepto el último, que se debe eliminar y devolver. Debes explicar cómo se puede simular el comportamiento de una pila utilizando una cola, y acompañar tu solución con el pseudocódigo correspondiente.

Sistemas de etiquetas binarios (1.25 puntos)

Escribe un programa que lea una cadena binaria desde la línea de comandos y aplique el siguiente sistema de etiquetas (00, 1101): si el primer bit es 0, elimina los primeros tres bits y añade 00 al final; si el primer bit es 1, elimina los primeros tres bits y añade 1101 al final. Repite mientras la cadena tenga al menos tres bits. Determina si las siguientes entradas se detendrán o entrarán en un bucle infinito: 10010, 100100100100100100. Usa una cola. Explica los conceptos detrás del sistema de etiquetas y cómo se puede modelar el problema con una cola. Incluye el pseudocódigo de tu solución.

Evaluación

La evaluación de este taller se realizará en base la siguiente escala:

1. **Muy superior (4.7-5.0):** El programa o función compila y se ejecuta correctamente, realizando todas las operaciones solicitadas con la eficiencia y precisión requeridas. El código está bien estructurado, es eficiente y está completamente documentado. Además, se explican

claramente los conceptos subyacentes al problema y se proporciona un pseudocódigo preciso y detallado.

2. **Superior (4.2-4.6):** El programa o función compila y se ejecuta correctamente, pero presenta ligeras desviaciones en la eficiencia o en la organización del código. Las operaciones son correctas, pero pueden ser optimizadas o mejor documentadas. Los conceptos subyacentes al problema son explicados de manera adecuada y se incluye un pseudocódigo funcional pero con detalles menores que podrían mejorar.
3. **Satisfactorio (3.6-4.1):** El programa o función compila y se ejecuta, pero contiene errores menores que afectan parcialmente la precisión o la eficiencia de las operaciones. La estructura del código y la documentación pueden mejorarse. La explicación de los conceptos es correcta, pero carece de detalles, y el pseudocódigo puede estar incompleto o no completamente claro.
4. **Suficiente (3.0-3.5):** El programa o función compila, pero tiene problemas importantes en la ejecución o en la precisión de las operaciones. El código es difícil de seguir, y la documentación es insuficiente o incorrecta. Los conceptos subyacentes al problema están poco claros o incompletos, y el pseudocódigo presenta errores importantes.
5. **Insuficiente (0.0-2.9):** El programa o función no compila o no puede ejecutarse debido a errores graves. Las operaciones no se realizan correctamente. El código no está bien documentado ni organizado. No se explican los conceptos subyacentes de manera adecuada y no se proporciona pseudocódigo o este es incorrecto.

Escala de Evaluación

La escala de evaluación se construyó de acuerdo con el Artículo 79 del Decreto Rectoral No. 1751 del 31 de octubre de 2022.