



INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

ANÁLISIS DE ALGORITMOS Y ESTRUCTURAS DE DATOS

Enunciado Laboratorio nro. 2

Profesor: Cristián Sepúlveda S.

13 de diciembre de 2024

Tabla de contenidos

1. Introducción	2
2. Resultados de aprendizaje (RdeA)	2
3. Instrucciones	2
4. Evaluación	4
5. Fecha de Entrega	4
6. Problema propuesto	5
6.1. Descripción problema	5

1. Introducción

La teoría de colas estudia mediante métodos formales las líneas de espera. Principalmente se modelan colas para poder predecir su longitud y el tiempo de espera. Esta teoría es considerada una rama de la investigación de operaciones, debido a que los resultados son utilizados generalmente por los administradores con el fin de tomar decisiones sobre los recursos necesarios para proporcionar un servicio. La teoría de colas presenta aplicaciones en telecomunicaciones, transporte, informática, y particularmente en el modelado de problemas de producción, en el diseño de fábricas, tiendas, oficinas y hospitales, así como en la gestión de proyectos.

2. Resultados de aprendizaje (RdeA)

- Formular algoritmos para problemas computacionales.
- Calcular y analizar la eficiencia de algoritmos computacionales.
- Desarrollar la capacidad de comunicarse efectivamente en español de forma escrita.
- Demostrar capacidad crítica en el análisis de resultados.

3. Instrucciones

1. El trabajo se realizará en grupos de a lo más 2 personas.
2. Describir en pseudocódigo los algoritmos propuestos para resolver el problema, utilizando como estructura de datos lineales **solamente listas, pilas y colas**. La utilización de cualquier otra estructura de datos lineal para resolver el problema, por ejemplo arreglos, implicará la **calificación mínima** en el laboratorio.
3. Calcular la complejidad de tiempo para cada algoritmo que maneje listas, pilas o colas.
4. Implementar los algoritmos propuestos utilizando el lenguaje de programación C. Las estructuras de datos lineales utilizadas deben ser implementadas mediante listas enlazadas.
5. Evaluar la implementación con distintas entradas.
6. Generar un reporte con las siguientes secciones:

- **Introducción**

Se proporciona el contexto y la motivación para el trabajo. Se explica brevemente la teoría relevante con suficiente detalle como para introducir leyes, ecuaciones o teoremas relevantes. Se indica claramente el/los objetivo/s.

- **Método**

Se describen el equipo, los materiales y los procedimientos utilizados en los experimentos. Se describen procesamientos o cálculos realizados sobre los datos utilizados.

- **Algoritmos propuestos**

Se describen en pseudocódigo, de forma ordenada, los algoritmos utilizados para resolver el problema.

- **Cálculo de complejidad**

Se describen los cálculos realizados para obtener la complejidad de tiempo y el orden de complejidad de los algoritmos propuestos.

- **Conclusiones**

Se recuerda al lector qué problema se estaba investigando. Se resumen los hallazgos en relación con el problema/hipótesis. Se identifican brevemente las implicaciones generales de los principales hallazgos.

- **Apéndice**

Se agrega información que ayuda a los lectores a comprender el proceso de investigación.

- **Referencias** (opcional)

Se enumeran los detalles de todas las publicaciones citadas en el texto, permitiendo a los lectores localizar las fuentes de forma rápida.

7. Consideraciones:

- La utilización de cualquier estructura de datos lineal distinta listas, pilas y colas para resolver el problema, por ejemplo arreglos, implicará la **calificación mínima** en el laboratorio.
- No existe restricción para el número máximo de páginas del informe.
- En el apéndice **se debe** incluir un manual de usuario explicando la correcta ejecución de la implementación entregada, especificando el formato tanto de los parámetros de entrada como de las salidas. Se espera además una descripción de las funcionalidades del programa y que se listen posibles errores. No incluir un manual de usuario implica **calificación mínima**.

ma en el laboratorio.

4. Evaluación

■ Informe

Entregable: informe en formato PDF. El nombre del documento debe tener el siguiente formato:

T2_coordinación_nombre_apellido.pdf.

Ponderación: 70 %. Pauta en página principal de laboratorio.

Entrega: Buzón de entrega en Campus Virtual.

■ Implementación

Entregable: archivo(s) con la implementación de la solución en lenguaje C. El nombre del programa principal sigue el formato:

T2_coordinación_nombre_apellido.c

El resto de los archivos que se entregue, siguen nomenclatura personal. Se agrupan en un archivo único archivo comprimido de nombre:

T2_coordinación_nombre_apellido.zip.

Ponderación: 30 %. Pauta en página principal de laboratorio.

Entrega: Buzón de entrega en Campus Virtual.

5. Fecha de Entrega

- **Fecha de entrega:** viernes 20 de diciembre hasta las 19:00 horas.

6. Problema propuesto

6.1. Descripción problema

El transporte marítimo de mercancías ha sido ampliamente utilizado a lo largo de la historia. Con el advenimiento de la aviación, ha disminuido la importancia de los viajes por mar para pasajeros, aunque sigue siendo popular para viajes cortos y cruceros de placer. El transporte por agua es más barato que el transporte por aire o tierra, pero significativamente más lento en distancias extensas. Actualmente, el transporte marítimo representa aproximadamente el 80 % del comercio internacional, según la UNCTAD en 2020.

Maersk Line es la compañía de transporte de contenedores más grande del mundo, tanto por tamaño de flota como por capacidad de carga, y cuenta con un equipo dedicado de más de 80,000 operarios en 130 países. Es mejor conocida por su cobertura global. Además de sus principales rutas comerciales entre Asia y Europa y la ruta Transatlántica, Maersk Line también ofrece una amplia cobertura entre América del Sur, Europa y África. La compañía opera las 24 horas del día, los 7 días de la semana, con equipos dedicados a las operaciones de buques. Estos equipos están en el corazón de las operaciones de contenedores.

Debido al tamaño de la compañía, esta se ve enfrentada a diversos problemas que afectan la preparación de la carga, desde problemas con el peso de los contenedores hasta el mal funcionamiento de las unidades de refrigeración.

Uno de los problemas más importantes de la operación diaria de la compañía, es el de agilizar los procesos de embarque de cargas. Con el fin de mejorar los tiempos del proceso, los ingenieros e ingenieras estudian los procesos de embarque mediante simulaciones.

La simulación de una cola de embarque consta de una grúa encargada de elevar-descender las cargas. Las cargas llegan a la cola en intervalos regulares de T unidades de tiempo. Al momento de llegar, las cargas son identificadas por un número mayor que cero que representa el orden de llegada a la cola de embarque $(c_1, c_2, c_3, \dots, c_n)$. Además, se registra el instante de llegada, siendo este un valor mayor o igual que cero y menor o igual que el valor del tiempo de duración de la simulación $(0, X, 2X, 3X, \dots)$. En cada unidad de tiempo, se debe verificar si es necesario embarcar las cargas. Para ello, se debe esperar una orden que indique que los barcos están dispuestos en un cierto orden en el puerto y que se puede iniciar un periodo de embarque. Al recibir la orden, se deben embarcar todas las cargas que estén esperando en ese momento, asumiendo un tiempo de embarque de una unidad de tiempo por cada carga.

En caso de que, durante un período de embarque, arribe una nueva carga a la cola,

esta no será embarcada, y deberá esperar en la cola hasta que se active el siguiente período de embarque.

Se le solicita a usted la construcción de un programa en lenguaje C que simule el comportamiento de una cola de embarque. Para su programa considere:

- Se debe construir una función nombrada **embarcar**, con la siguiente sintaxis:

```
embarcar(num probabilidad): bool
```

la función debe retornar, con cierta probabilidad de entrada, un valor booleano que determine si se embarcan (1) o no (0) las cargas de la cola.

- El programa deberá recibir como entrada el tiempo de duración de la simulación, el valor del intervalo de llegada de las cargas y la probabilidad (valor entre 0 y 1) de que la función `embarcar()` retorne un valor verdadero (1).
- El programa deberá mostrar el estado de la cola en cada unidad de tiempo y una vez finalizada la simulación, entregar los tiempos en los que se activó la orden aleatoria de embarque (retorno verdadero de la función `embarcar`), el tiempo en el que fue embarcada cada carga, el tiempo de permanencia en la cola de cada carga embarcada, el tiempo promedio de permanencia de las cargas embarcadas, el número de cargas embarcadas y el número de cargas no embarcadas.
- La solución propuesta debe utilizar al menos una estructura de datos tipo cola para representar la cola de embarque.
- Todas las estructuras lineales utilizadas (listas, pilas y colas) deben ser implementadas sobre listas enlazadas.
- Cada carga debe tener asociada una estructura que mantenga al menos el tiempo de llegada y el tiempo de atención.
- Cualquier supuesto que realice para su solución, regístrela en la sección algoritmos propuestos del informe.

Entrada de ejemplo:

Tiempo de simulación: 45 unidades de tiempo

Intervalo de llegada: 10 unidades de tiempo

Probabilidad de la función `embarcar()`: 0.1

Salida de ejemplo:

Suponiendo que la función `embarcar()` retorna verdadero en las unidades de tiempo 15 y 39 (retornando falso en las restantes unidades de tiempo). La secuencia de estados de la cola de embarque desde $T = 0$ hasta $T = 45$ es:

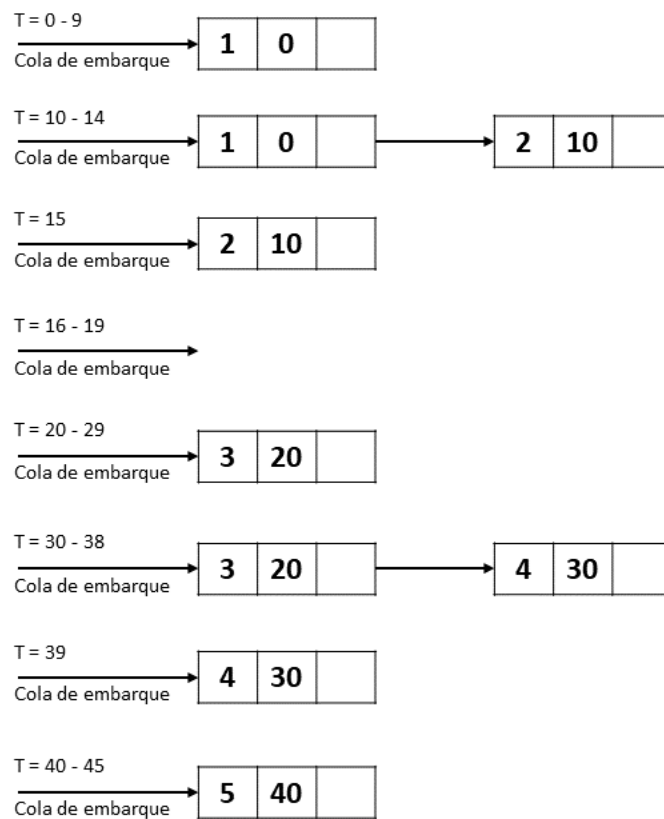


Figura nro. 1: secuencia de estados de la cola de embarque.

Resumen de la simulación:

- Tiempos de activación de períodos de embarque: 15 y 39
- El momento de embarque de cada carga es: c1: 15, c2: 16, c3: 39, c4: 40, c5: -1 (indica no embarcada)
- El tiempo de permanencia de cada carga es: c1: 15, c2: 6, c3: 19, c4: 10, c5: -1 (indica no embarcada)
- Tiempo promedio de permanencia de las cargas embarcadas: 12,5 unidades de tiempo
- Número de cargas embarcadas: 4
- Número de cargas no embarcadas: 1