

Objetivos

Unidad 1: Análisis de Algoritmos

- OE1.1. Calcular la complejidad temporal de algoritmos iterativos.
- OE1.2. Calcular la complejidad espacial de algoritmos iterativos.
- OE1.3. Caracterizar la entrada de un algoritmo iterativo con el fin de calcular la complejidad para el mejor y peor caso.
- OE1.4. Analizar algoritmos independiente de una implementación concreta (no dependiente del lenguaje de programación).
- OE1.5. Utilizar notación asintótica para describir la complejidad de algoritmos.
- OE1.6. Evaluar varios algoritmos que resuelven el mismo problema en términos de sus complejidades computacionales.
- OE1.7. Comprender la importancia del Modelo RAM en el proceso de análisis de algoritmos.

Unidad 2: Diseño y Construcción de Estructuras de Datos

- OE2.1. Proponer y justificar un diseño para implementar una estructura de datos, siguiendo una metodología y considerando la flexibilidad en los tipos de datos y la complejidad temporal de las operaciones.
- OE2.2. Implementar estructuras de datos extensibles y generales utilizando interfaces, herencia y tipos de datos genéricos.
- OE2.3. Escribir el invariante de una clase e implementar los métodos necesarios para su verificación utilizando los elementos apropiados del lenguaje.
- OE2.4. Diseñar, adaptar y utilizar estructuras de datos de acceso directo por llave, las cuales están basadas en la capacidad de las funciones de hashing para localizar una posición física a partir de una llave lógica.
- OE2.5. Utilizar estructuras lineales FIFO, LIFO y diccionarios como parte de la solución de un problema.
- OE2.7. Diseñar y construir las pruebas unitarias de cada una de las estructuras de datos lineales implementadas.

Enunciado

Cierto excéntrico millonario extranjero decidió invertir y emprender en la ciudad de Cali. Su visión es empezar una tienda de videojuegos que presta sus servicios de una manera innovadora.

Este es el funcionamiento que tiene pensado el nuevo negocio de videojuegos: El cliente antes de ingresar al establecimiento, debe revisar los catálogos digitales, estos se encuentran en la página web del aplicación móvil. En estos catálogos se pueden revisar todos los títulos la tienda, reseñas y hasta críticas de videojuegos consultados. Al finalizar la virtual con el catálogo, el cliente añade a su lista juegos de su interés. Una vez lista genera un código que el cliente deberá tener en cuenta al ingresar al establecimiento, ya que en la primera sección del establecimiento, deberá código en una tablet que se les proporcionará a la entrada. Esta tablet cargará la cliente. Cabe resaltar que cada juego disponible en la tienda tiene un código que



negocio o en la disponibles en interacción terminada, la ingresar dicho lista del lo identifica.

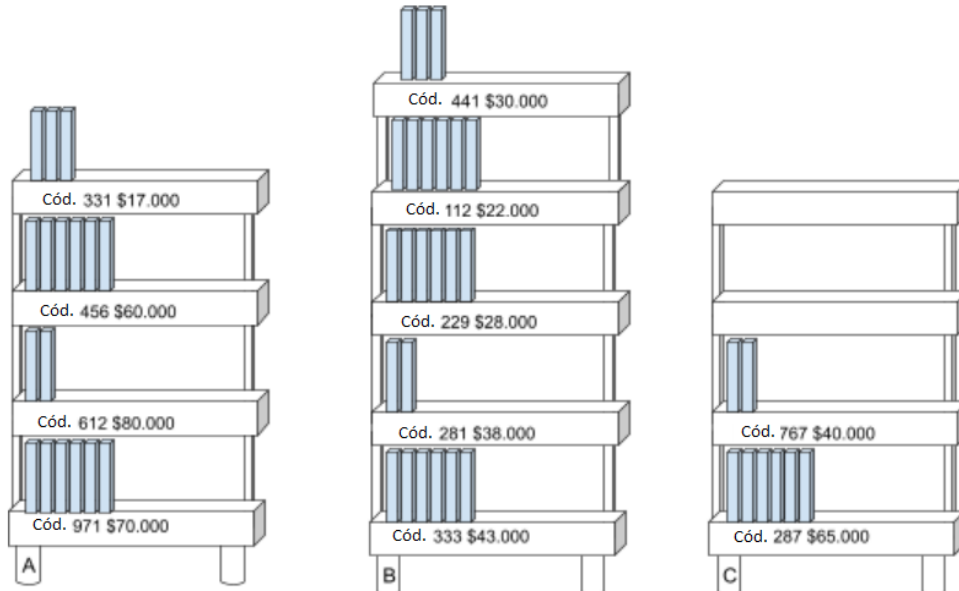
Ya teniendo en sus manos dicha lista, el usuario prosigue su camino a la sección 2 donde se realiza la búsqueda. Este es un lugar provisto con un conjunto de estaciones donde se puede conectar la tablet provista. Estas estaciones **permiten realizar consultas muy rápidas**. La lista ingresa los nombres de los videojuegos a la estación y ésta le indica los bloques o estanterías en donde debe buscar el juego de su interés. Para agilizar el proceso de recolección de los ejemplares, el equipo **ordena el listado de los juegos** acorde con la ubicación de las estanterías de tal manera que el comprador siga la mejor ruta (si el juego se encuentra agotado, su código no aparecerá en la lista ordenada final). En esta etapa el cliente puede utilizar 2 algoritmos distintos de ordenamiento para cumplir dicha tarea.

Al concluir esta fase, el comprador prosigue su tránsito a la sección 3 de las copias físicas de los juegos donde recogerá aquellos presentes en su lista en el orden suministrado en la etapa anterior. Para facilitar esta labor a cada cliente, se le asigna un cesto automatizado que le sigue para ir ubicando los juegos que va encontrando, uno encima del otro, siguiendo el orden suministrado en la etapa previa.

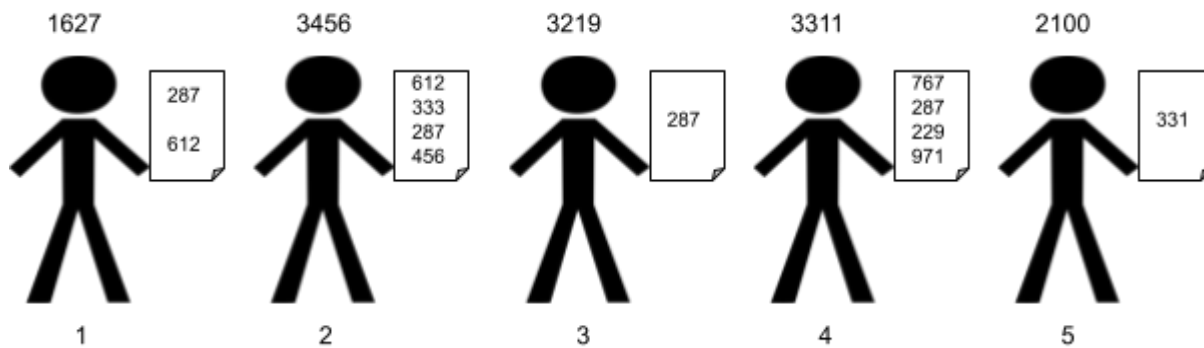
Con su cesto lleno, al cliente sólo le resta pasar al punto final, la sección 4 de pago. Teniendo en cuenta el tiempo en el que ha entrado a la tienda, más lo que se ha tomado recogiendo los juegos de cada estantería, se va estableciendo el orden con el que se ingresa a la fila de cajas. En este caso, la estrategia es la siguiente, todos hacen una única cola aunque haya varios cajeros o puntos de atención. Cuando uno o más puntos de atención estén disponibles, pasan tantos clientes como puntos disponibles haya, para ser atendidos uno a uno en cada uno de ellos. Ya que los clientes pasan provenientes de la única fila, se puede afirmar que todos son atendidos en el mismo orden de llegada. El orden de salida sí puede variar, pues éste dependerá del tiempo que se demore la atención de cada uno en el punto (cantidad de elementos que se van a comprar). Hay que resaltar que como los juegos vienen en un cesto, el último juego añadido será el primero facturado y empacado.

A continuación se muestra un ejemplo:

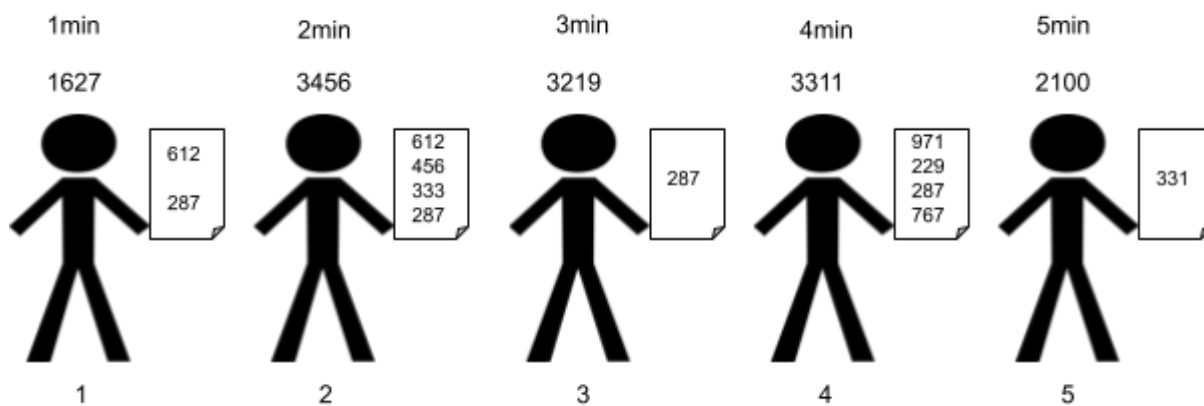
Estanterías:



5 clientes al salir de la sección 1:

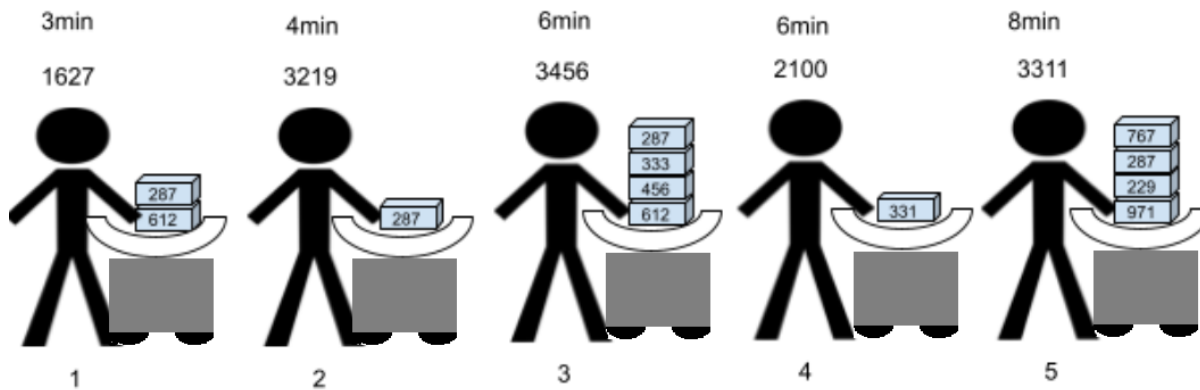


5 clientes al salir de la sección 2:



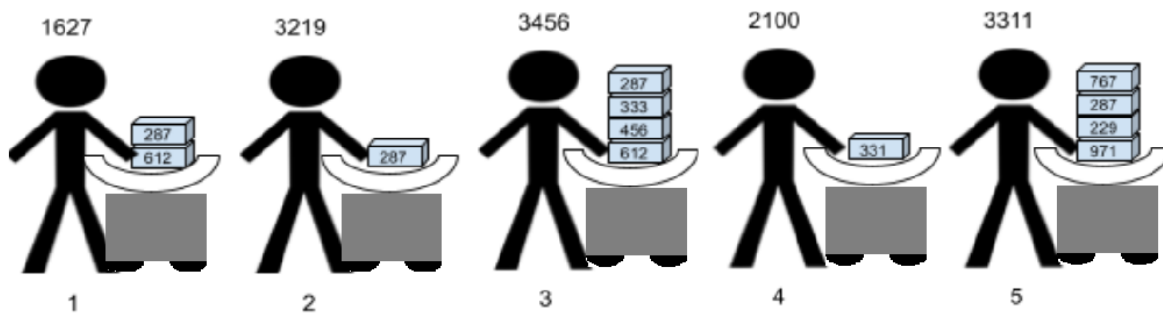
Cada cliente sale en el orden suministrado con una unidad de tiempo de diferencia entre ellos (ej. un minuto).

Fila para pagar luego de recoger los juegos

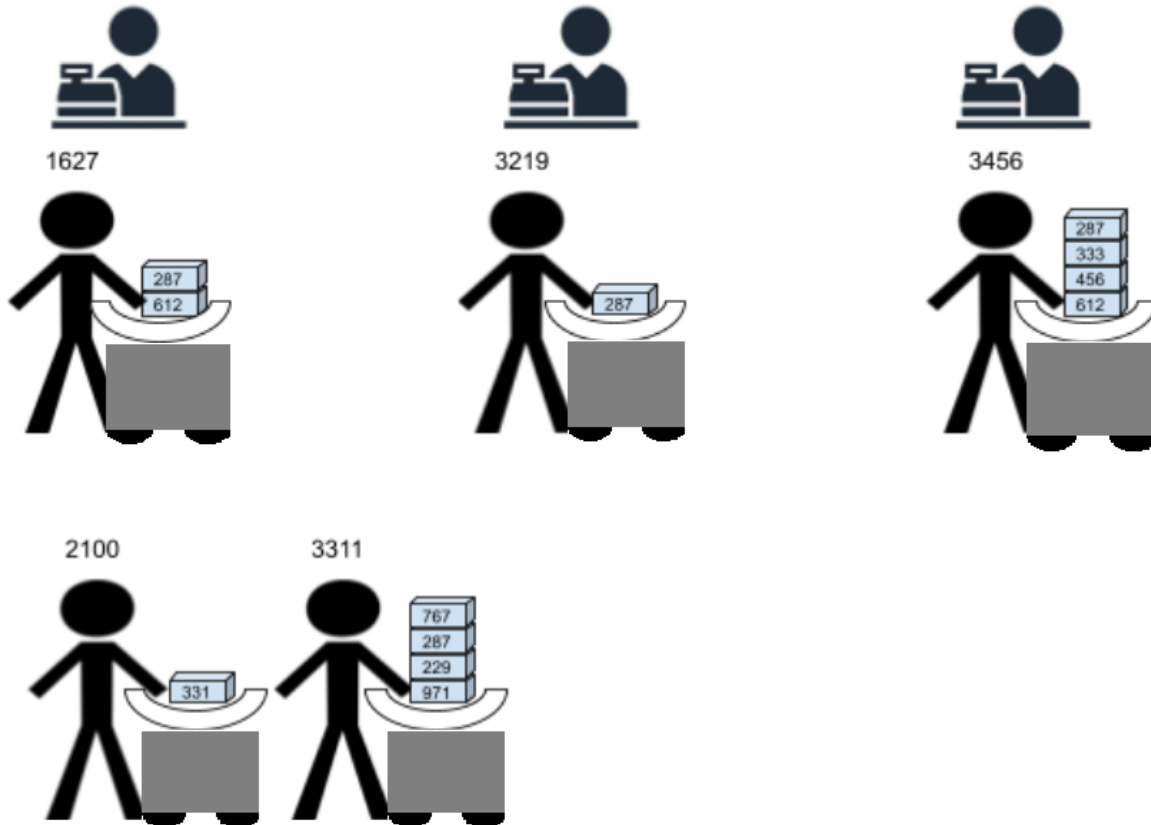


Cada cliente se demora 1 unidad de tiempo en recoger un juego, por tal, el tiempo total al salir de la sección 3 será el valor previo proveniente de la sección 2 más la cantidad que se toma en recoger los ejemplares. Ya a partir de ese tiempo, sí se establece el orden de la cola de pago. Si dos usuarios se toman el mismo tiempo, sale primero aquel que estaba antes a la salida de la sección previa.

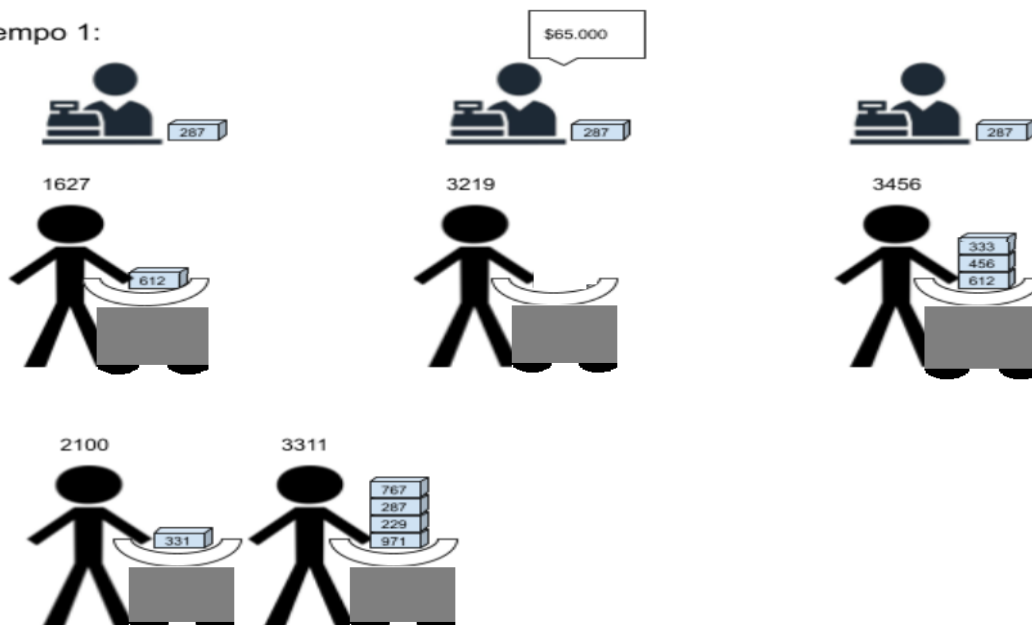
Proceso de pago teniendo 3 cajeros:



Tiempo 0:



Tiempo 1:



Tiempo 2: \$145.000



1627



2100



3456



3311



Tiempo 3:



3311



\$17.000



2100



3456



Tiempo 4:



Tiempo 5:



Tiempo 6:





Nota: El orden en que se empaacan los juegos de los clientes se da teniendo en cuenta el montón que tiene el cajero a su lado. Aquel que esté en el tope obviamente será el primero en empaacarse y así sucesivamente.

Usted debe desarrollar un programa (con interfaz gráfica) que le permita a los caleños conocer un poco acerca de cómo funcionaría esta nueva tienda en la ciudad. Su programa ha de simular el proceso de compra de videojuegos desde la salida de la sección 1. Para ello debe recibir los siguientes datos básicos: catálogo de juegos (código del juego, cantidad de ejemplares, estantería donde está ubicado, precio del juego), cantidad de cajeros a utilizarse durante la jornada, serie de códigos o cédulas que representan a los clientes (en el orden en que entraron a la tienda) y lista de juegos por comprador (sus códigos). Como resultado, su aplicación debe informar el orden de salida de los clientes, el valor de cada compra y el orden en que quedaron empacados sus juegos.

Usted debe utilizar las estructuras de datos (colas, pilas, tablas hash) que más se adecúen a las necesidades de cada etapa dentro del funcionamiento de esta particular librería. Asimismo, debe elegir dos algoritmos de ordenamiento que se le ofrecerán al cliente en la sección 2. Se cuenta con la restricción de que ambos deben tener complejidad temporal $O(n^2)$.

Usted debe utilizar el método de la ingeniería para resolver este problema y dejar evidencia en su informe de los resultados de cada fase. Recuerde revisar el [Resumen del Método de la Ingeniería](#) y el [ejemplo del Método de la Ingeniería aplicado a un problema](#).

Entregables.

1. Informe PSP0. Cada estudiante debe entregar el informe de su desarrollo.
2. Entrega de informe del método de la ingeniería.
3. Análisis de complejidad temporal de cada uno de sus algoritmos de ordenamiento.
4. Análisis de complejidad espacial de cada uno de sus algoritmos de ordenamiento.
5. Especificación de Requerimientos y Diseño.
6. Diseño del TAD para cada estructura de datos requerida.
7. Diseño del diagrama de clases desacoplado y utilizando generics.
8. Diseño de los casos de prueba. Adicionalmente debe explicar cómo se resuelven dos casos, paso a paso (con dibujos, si es necesario), por cada estructura de datos diseñada e implementada.
9. Diseño del diagrama de clases de pruebas unitarias automáticas.
10. Todos los archivos deben estar almacenados en GitHub y debe evidenciarse su uso desde el inicio del proyecto.
11. Implementación de las pruebas unitarias automáticas.

La tarea debe ser desarrollada en grupos de máximo 3 estudiantes.

Si tanto las entradas como salidas fuesen por consola, he aquí una posible especificación del problema de tal manera que pueda generar distintos casos de prueba que le permita corroborar la correctitud de la solución:

Entrada.

La primera línea es la cantidad de casos de prueba.

Luego, la primera línea tiene la cantidad de cajeros disponibles durante el día. La segunda línea presenta la cantidad de estanterías. Para cada estantería está la primera línea con su identificador y un número j de juegos en dicha sección. Después hay j líneas cada una de ellas con un primer número que indica el código de identificación del juego, un segundo que es el precio y el tercero que indica la cantidad de ejemplares. Posteriormente, en la siguiente línea después de terminar con las estanterías, aparece el número c de clientes que ingresan a la tienda durante la jornada. Finalmente aparecen c líneas en donde el primer número indica la cédula del cliente y le siguen los números que identifican los códigos de los juegos a comprar.

Salida.

Por cada caso se imprimen $2c$ líneas, donde c es el número de clientes. En una línea aparece la cédula del cliente seguida del valor total de su compra y en la siguiente la serie de códigos de identificaciones de sus juegos comprados siguiendo el orden en que fueron empacados.

Ejemplo:

Teniendo en cuenta los datos de las ilustraciones.

Entrada	Salida
1	3219 65000
3	287
3	1627 145000
A 4	612 287
331 17000 3	2100 17000
465 60000 6	331
612 80000 2	3456 248000
971 70000 6	612 465 333 287
B 5	3311 203000
441 30000 3	971 229 287 767
112 22000 6	
229 28000 6	
281 38000 2	
333 43000 6	
C 2	
767 40000 2	
287 65000 6	
5	
1627 287 612	
3456 612 333 287 465	
3219 287	
3311 767 287 229 971	
2100 331	

La rúbrica con la que se evaluará esta prueba se encuentra en el listado de notas de tareas y seguimientos en la pestaña TI 1.