Informe Tarea 0 - IIC2133 Estructuras de Datos y Algoritmos

Matias Briones - 15621340

29 de marzo de 2022

1. Estructura de datos usada

La tarea fue realizada con las siguientes estructuras para facilitar su desarrollo:

- Restorant: cumple la función de guardar en ella el id de Local, la cantidad maxima de mesas en dicho local, y por ultimo guarda las direcciones a las mesas que este local posee. Dentro de esta estructura podemos encontrar la funcion que inicializa la estructura y la funcion que libera la memoria asociada a esta estructura.
- Mesa: Cumple la funcion de guardar en ella la informacion pertinente a las mesas de un determinado local. Contiene el id de la mesa, su capacidad, disponibilidad, y finalmente guarda las dirreciones a los asientos relacionados a esta misma estructura. Contiene las funciones que la inicilizan, funcio que verifica la disponibilidad, funciones que modifican la disponibilidad dependiendo de la salida o entrada de un Customer, y por ultimo la funcion encargada de liberar la memoria solicitada de la estructura.
- Customer: Cumple con el proposito de almacenar la informacion de id del cliente, y los items(ordenes) que este solicite. Cuenta con la funcion inicializadora de la estruxctura, y las funciones para liberar la memoria asociada.
- Menu: guarda la informacion relacinada a los menu, como id del item (plato) y precio. Cuenta con las funciones para inicializar la estructura y liberar la memoria asociada.
- List y Node: Ambas estructuras son utilizadas para el manejo de los items (ordenes) solicitdas por los clientes. List guarda la cantidad de ordenes y el costo total de ordenes, ambos atributos se guardan con el fin de evitar recorrer el arreglo para entregar esa informacion. Tambien, por concepto de lista ligada posee punteros a un Node que representa el inicio de la cadena y un Node que representa el final de la cadena. List posee como funciones aquellas necesarias paera agragar un nuevo nodo, eliminar el ultimo de la cadena, imprimir en conmsola (para debug), y para liberar la memoria asociada.

  Node por su parte guarda el precio y id de un item (orden), y a la par que con List, por concepto de lista ligada, tiene punteros que diregen a un nodo anterior y nodo siguiente en la cadena.

Las estructuras Restorant, Mesa y Customer se guardan una dentro de otra, haciendo más comprensible como llegar a cada dato que quiero intervenir. Ademas al estar ordenados en cascada, al liberar la memoria en Restorante este libera en secuencia Mesa y Customer, facilitando la liberación.

Menu se diseño para poder guardarse en un array, pero como se necesitaba que cada elemento del array contenga más de un solo valor, se opto por crear un array de estructuras Menu.

La utilizacion de List y Node fue escogida para poder crear listas mutables por medio de listas ligadas. Dado que no se tenia claro cuantos items podian comprar los Customer y añadiendo que podian eliminarlos o añadirles de otros clientes, se necesitaba cierta flexibilidad en cuanto a la estructura para que pueda guardar todos oos que sean necesario. No era posible utilizar un array debido a que estos son inmutables.

2. Complejidad de los eventos

2.1. OPEN TABLE

Complejidad O(n^2) debido a que itera en un for para para crear un array donde almacenar los locales, para luego iterar sobre el array para iniciar las mesas.

2.2. MENU-ITEM

Complejidad O(n) debido a que itera en un for realizando n iteraciones (siendo n el numero de items).

2.3. CUSTOMER

Complejidad O(n) debido a que itera en un for al cual se le pasan n elementos (estos vienen siendo la capacidad del local).

2.4. TABLE-STATUS

Complejidad O(n) debido a que itera en un for al cual se le pasan n elementos (estos vienen siendo la capacidad del local).

2.5. ORDER- CREATE

Complejidad O(n) debido a que itera en un for al cual se le pasan n elementos (estos vienen siendo la capacidad del local).

2.6. BILL-CREATE

Complejidad O(n^2) debido a que debe iterar sobre dos for. Primero debe recorrer una lista de largo n (la capacidad del local), para luego iterar sobre cada item que posee (estos son nodos de una lista ligada).

2.7. CHANGE-SEATS

Complejidad O(n^2) debido a que itera en dos for. Ambos for, uno despues del otro, se les entrega una cantidad n de elementos (siendo n la capacidad de cada mesa).

2.8. PERROU-MUERTO

Complejidad O(n^2) debido a que debe iterar sobre dos for. Primero debe recorrer una lista de largo n (la capacidad del local), para luego iterar sobre cada item (platos que solicito) que posee (estos son nodos de una lista ligada).

3. Ventajas y desventajas de listas ligadas y array

Ventajas Listas Ligada:

1. La principal ventaja de estas estructuras son su flexibilidad al momento de añadir elementos o eliminarlos, este tipo de estructuras no estan limitadas por el numero que elementos que pueden tener luego de ser inicializadas, y pueden crecer tanto como la memoria lo permita.

2. No es necesario que los nodos al ser guardados en la memoria estos permanezcan juntos como ocurre en los array.

3. No necesito conocer el largo de la lista ligada pàra recorrerla, ya que se indicará el termino de la lista cuando el ultimo nodo de esta presente el puntero siguiente como vacio. Esto nos evita problemas como en los array donde podemos pasarnos de la memoria asignada.

Ventajas Array

1. Conocimiento de la posicion de los elementos que lo componen. Si es necesario buscar el i-esimo dato del array es facil encontrarlo.
2. La velocidad de busqueda en un array es O(1), lo que la hace eficiente en comparación a otras estructuras
3. No hay desperdicios de memoria debido a que al solicitar espacio parea almacenar estas estructuras son todos aledaños.

Desventajas Listas Ligadas

1. Dificultad de recorrer la lista. Al no conocer la ubicacion de los elementos que la componen, buscar el i-esimo elemento repercute en hacer una busca por toda la lista.
2. Mayor complejidad de estructurar. Definir una estructura de lista ligada es más complejo que un array, siendo que este esta implementado por C, mientras que las listas deben crearse para ser utilizadas. Ademas todas las funciones para utilizarlas como recorrer, mostrar, y eliminar (entre otras), suele ser más complejo de diseñar.
3. La listas ligadas suelen consumir más espacio en memoria debido a que deben guardar punteros.

Desventajas Array

1. Poca flexibilidad para aumentar la cantidad de elementos de este luego de ser inicializado.
2. Solo puede almacenar un tipo de datos, es homogenea.
3. El compilador no sabe de que largo es el array, por lo que si le doy una direecion que no pertenece al array este me retornará datos almacenados en la memoria que podrian interpretarse de cualquier forma, ya que no pertenecen al array.

4. Ventajas de C versus Python

C es un lenguaje compilado, a diferencia de python que es interpretado. C es un lenguaje de bajo nivel, lo que implica que no tiene funcionalidades muy desarrolladas para facilitar el trabajo al programador como python, que es de alto nivel. Python para poder facilitar el trabajo del programar realiza muchas funcionalidades por detras que el usuario no percibe, esto a costa de perder la posibilidad de re-diseñar funciones ya establecidas para personalizar el trabajo, y a costa de un costo computacional de velocidad y memoria.

El lenguaje C tiene un mayor control sobre los recursos del computador, lo que hace que sea más eficiente con respecto a Python, por que hay que recordar que Python se encuentra escrito en C, por lo que dificilmente puede supèrar en eficiencia a este. Python como se decia, esta más orientado a simplificar la escritura de codigo.

5. Fuentes

1. https://aprendiendoaprogramar.es/blog/ventajas-y-desventajas-de-array/
2. https://github.com/DCCentral-de-Apuntes/intro-C
3. ¿Por qué Python es más lento que C?
4. ¿Cuál es la diferencia entre una lista y un arreglo programación?