Ing. de Sistemas e Informática
Estructuras de Datos y Algoritmos
Jorge Mario Londoño P.

Ejercicios 2

Estructuras de datos básicas

1. Se quiere implementar una de las estructuras básicas (Bag, Queue, Stack) con tipos genéricos para garantizar la integridad de los datos en la estructura, sin embargo, el ADT a almacenar tiene múltiples implementaciones (e.g. Punto2DCartesiano y Punto2DPolar). Cómo sería posible lograr esto? Indicar los cambios necesarios para implementar el tipo genérico.

2. Considerar que tipo de estructura sería aplicable a cada uno de los problemas siguientes, justificando brevemente:

- En un help-desk se reciben peticiones de soporte. Las peticiones deben atenderse en el orden en que fueron recibidas.
- Se quiere guardar el historia de notas, para sacar promedios por estudiante, por grupo, por curso.
- Una bodega recibe cajas (con un número de stock) y las guarda en forma vertical. Así mismo, cuando se despachan, cada camión se carga con las cajas empezando desde la parte superior.

3. En muchos lenguajes se utilizan estructuras anidadas por paréntesis. Hacer un algoritmo que lee un archivo de texto plano y determine si todas las estructuras anidadas se cierran apropiadamente.
Por ejemplo

```
[[]({}())]{([])}
```

es correcta y

```
{[)(]}  ,  {([)}
```

son incorrectas.

2. Idear un método peek() para retornar el elemento en el tope de la pila sin eliminarlo. Proponer su implementación para la pila basada en arreglo y para la basada en lista enlazada.

3. La estructura Bag analizada en clase no tiene forma de eliminar elementos.

- Dar una implementación de la operación remove(Item) que remueve un item de la bolsa en caso de estar presente. Si no existe no hace nada. Asumir una implementación de Bag basada en arreglo.
- Dar una implementación de la operación remove(Item) para cuando la bolsa está implementada por medio de una lista enlazada.

4. Escribir un procedimiento que tome como entrada una lista enlazada y devuelva una nueva lista con los elementos en orden inverso.

5. La lista doblemente enlazada ofrece la posibilidad de recorrer los elementos en ambos sentidos. Proponer una implementación de la estructura PilaCola que implementa al mismo tiempo las funcionalidades de la Pila y de la Cola.

6.* Escribir un procedimiento que tome como entrada una lista enlazada y retorne una nueva lista con los elementos en orden aleatorio.

7. Ejercicios de codificación de operaciones con listas simples: Ver ejercicios.

8. ADT Juego de Naipes: Un juego de naipes se compone de un conjunto de ADTs Carta. Al inicializar, el juego de naipes contiene todas las cartas del juego. El Juego de Naipes requiere las operaciones barajar y removerCarta. La primera reorganiza de forma aleatoria las cartas en el juego y la segunda remueve una carta (ya no queda disponible) y devuelve al cliente la carta removida. Dar una definición, implementación y pruebas unitarias del Juego de Naipes.

9. El browser necesita mantener el historial de sitios visitados para la operación de los botones “Back” y “Forward”. Diseñar e implementar un ADT `Historia de Navegación` que en su API maneje las siguientes operaciones:
- back: devuelve el URL de la página previamente visitada.
- forward: devuelve el URL de la página donde estaba justo antes del back.
- visit: registra el URL cada que se visita una página (por ejemplo al hacer click en un link).
Observar que en el historial de back y forward se pueden tener muchos URLs.