

MEMORIA PRÁCTICA 2

1. Una aplicación habitual del TAD PILA es para evaluar expresiones posfijo. Describe en detalle qué TADs habría que definir/modificar para poder adaptar la pila de enteros (P3_E1) de modo que tengamos una que permita evaluar expresiones posfijo.

Habría que modificar el TAD PILA, incluyendo una función la cual aplique un algoritmo que te permita evaluar expresiones posfijo. También habría que implementar un TAD ITEM el cual pueda manejar información con números y símbolos, por lo tanto su estructura se implementaría con un único campo de tipo void*. Implementaría Funciones las cuales te permitan leer los símbolos de la cadena, funciones que te permitan distinguir entre operandos y operadores y funciones que detecten el símbolo de fin de cadena.

2. Explicad las decisiones de diseño y alternativas que se han considerado durante los distintos ejercicios de la práctica. En particular, explicad las claves de la implementación y de las decisiones de diseño de los ejercicios P2_E2, P2_E3 y P2_E4.

p2_e2: En este ejercicio, hemos implementado dos TADS. Uno de ellos nos permite trabajar con información de tipo entero. Hemos creado las funciones principales para crear, liberar, obtener, dar, copiar, comparar e imprimir la información de estos elementos. En el programa principal de este ejercicio, combinaremos el TAD Pila con lo implementado anteriormente. Utilizamos la Pila para realizar operaciones entre elementos y para comprobar que estas operaciones se realizan de forma correcta. Esto nos sirve para observar si el nuevo TAD que hemos implementado funciona correctamente.

p2_e3: La principal diferencia entre este ejercicio y el anterior es la forma de implementar el TAD Pila. En este apartado, se nos pide implementar una pila genérica. Esto quiere decir que mediante el uso de punteros a funciones podremos utilizar la Pila para el tipo de elemento que queramos. Lo que cambia es que al inicializar la pila, tenemos 3 tipos de punteros a función(copia, imprimir e ini) y tenemos que pasarle estos 3 argumentos. Los punteros a función nos permiten utilizar esta pila para cualquier tipo de elemento, lo cual es mucho más útil.

p2_e4: Para este ejercicio hemos implementado una función nueva en el TAD Grafo. Esta función nos permitirá recorrer un grafo desde un nodo origen hasta otro destino. Lo que queremos comprobar en este apartado es si se puede encontrar un camino entre dos nodos dados mediante el uso de este algoritmo. La principal idea es empezar en un nodo inicial del grafo e ir comprobando las conexiones que hay desde el nodo que partes e ir avanzando hasta el nodo que queremos. Se ha definido un campo “etiqueta” que nos sirve para ir marcando los nodos los cuales hemos ido recorriendo y comprobando que no es el nodo destino. Gracias a las conexiones entre nodos, podremos finalmente comprobar si existe o no un camino entre los nodos elegidos. En el programa principal, tendremos 3 argumentos los cuales tendremos que introducir por pantalla. Uno es el fichero que contiene el grafo, el segundo es el nodo origen y el tercero el nodo destino. Abriremos el fichero que pasamos como 1er argumento y la utilizaremos la función que nos lee el grafo de un archivo. Imprimimos el grafo y si existe un nodo destino(camino encontrado), imprimimos los dos nodos.

3. CONCLUSIONES FINALES

En general, los primeros ejercicios de la práctica han sido sencillos de realizar. Nos ha ido costando un poco más ya que algunas veces arrastrábamos errores previos, pero una vez arreglados hemos sabido afrontarlo. Desde nuestro punto de vista, la práctica nos ayuda a visualizar de una forma más gráfica las operaciones que se van realizando. Beneficia más que el papel ya que podemos observar lo que está ocurriendo en cada momento y podemos arreglar los errores que tengamos más fácilmente. Creemos que ha reforzado bastante nuestros conocimientos sobre la teoría, sobretudo en la utilización de la pila y la gestión de memoria. Donde más dificultades nos hemos encontrado ha sido en el ejercicio 4. Pensamos que ha sido complicado implementar este algoritmo con el pseudocódigo que se nos ha proporcionado y nos ha quedado algún error por pulir en este aspecto.