

PRÁCTICA Nº 3: 1 sesión

(S3: 29 de Marzo y 6, 7, 22 y 23 de Abril)

USO DEL TAD PILA. JUEGO DE LAS TORRES DE HANOI

0. OBJETIVOS

El objetivo de esta práctica es la implementación del TAD Pila y su uso en la simulación del juego de las torres de Hanoi con tres torres y un número indeterminado de fichas introducido por el usuario.

1. INTRODUCCIÓN

En el templo de Benarés, a las orillas del Ganges, bajo la cúpula que marca el centro del mundo hay una lámina de cobre con tres agujas de diamante. En una de ellas, Bramah, cuando creó el mundo, puso 64 discos de oro, de manera que el disco más grande estaba sobre la lámina de metal y el resto de discos de mayor a menor uno sobre otro. Día y noche, incesantemente, los monjes cambian de aguja los discos con unas reglas fijas: Sólo pueden mover cada vez un sólo disco; y nunca puede estar un disco más grande sobre uno más pequeño. Dice la leyenda que cuando los sesenta y cuatro discos hayan sido cambiados de aguja, los monjes se convertirán en polvo y el mundo desaparecerá con un gran estallido.

Esta práctica consiste en realizar un juego que simule las torres de Hanoi y que permita decidir al jugador cuál es el movimiento que desea realizar, indicando si es o no posible, y realizándolo en su caso utilizando para ello la estructura de datos Pila.

2. ESPECIFICACIONES

El juego de las tres torres de Hanoi parte de una configuración de tres torres numeradas como 1, 2 y 3, con 'n' (supongamos $n=3$) fichas de tamaño creciente y ordenadas en la primera de ellas como se indica en la siguiente figura:

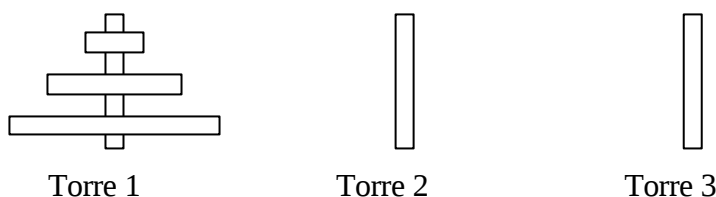


Figura (a)

El objetivo del juego es trasladar los discos de la torre 1 a la torre 3, usando la torre 2 como auxiliar, obteniendo la configuración que se muestra a continuación en la figura (b):

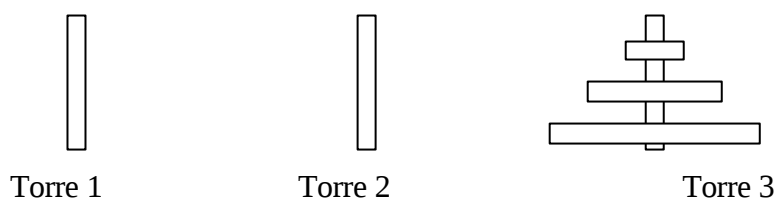


Figura (b)

Para realizar este trasbase se deben cumplir siempre los siguientes requisitos:

i.- Sólo se puede mover una pieza cada vez; y para coger una segunda pieza debemos de haber dejado primero la anterior en alguna torre.

ii.- Además una pieza sólo puede apilarse encima de una más grande.

Como se desprende de las restricciones anteriores cada torre del juego se comporta como un elemento del tipo pila.

Para la programación de este juego se debe implementar el tipo pila en forma de clase y realizar un programa que simule el juego utilizando dicha clase.

El juego que se implementará solicitará al usuario el número de discos con el que se va a jugar y mostrará por pantalla el estado inicial del juego (todas las piezas colocadas en la torre 1 y las torres 2 y 3 vacías).

A partir de ahí, pedirá sucesivamente pares de números indicando la torre origen desde la que coger la pieza y la torre destino a la que el usuario quiere realizar el movimiento. El programa analizará si la jugada es factible. Si el resultado del análisis es positivo moverá la ficha de una torre a otra. Si no lo es indicará que es una jugada imposible, indicando el por qué y pedirá un nuevo movimiento.

Después de cada jugada se mostrará el contenido de las tres torres.

El juego terminará cuando las torres 1 y 2 estén vacías y todas las fichas se encuentren en la configuración adecuada vista en la figura (b) mostrando el número de jugadas realizadas y el número mínimo de jugadas ($2^n - 1$) en el que se podría haber realizado.

3. REALIZACIÓN DE LA PRÁCTICA

Implementación de la clase y comprobación de su correcto funcionamiento.

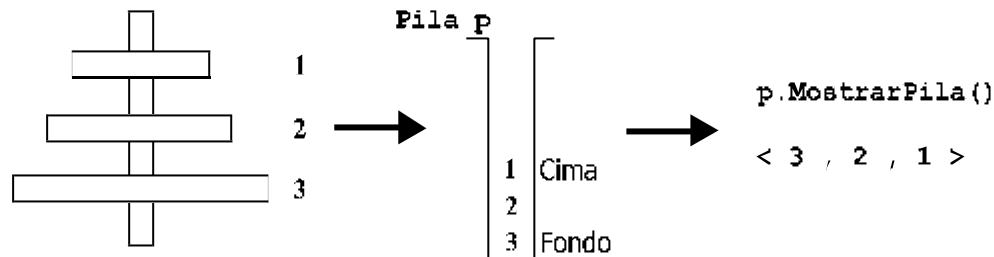
En esta tercera práctica, se debe implementar una clase en C++ que permita representar el TAD pila, asumiendo la interfaz estándar vista en teoría, y una implementación estática de la pila:

```
class Pila
{
public:
    // Los valores que contendrá la pila van a ser enteros
    typedef int Valor;

    Pila (void);                // Constructor por defecto
    bool Apilar (Valor);
    bool Desapilar (void);
    bool CimaPila (Valor &);
    bool PilaVacía (void);
    // Metodo para visualizar el contenido completo de la pila
    void MostrarPila (void);
private:
};
```

Los discos se representarán mediante la utilización de enteros. Los discos más grandes utilizarán valores enteros mayores y los discos más pequeños utilizarán enteros menores.

El método 'MostrarPila' mostrará por pantalla el contenido de la pila desde la parte más profunda (fondo de la pila) a la derecha, hasta la cima de la pila a la izquierda:



El correcto funcionamiento de la clase lo comprobaremos con el programa de prueba 'ProbarPilas.cpp' proporcionado por el profesor de prácticas.

Programación del juego de las tres torres de Hanoi.

Se realizará un programa completo llamado **##Hanoi.cpp** donde se utilice la implementación anterior de la clase Pila para realizar un juego de las torres de Hanoi, en el que el usuario decidirá en cada momento el movimiento de los discos que desea realizar.

El programa empezará pidiendo el número de discos con el que el usuario desea empezar la partida, y a continuación pedirá pares de números que indican de qué torre a qué torre se desea realizar el movimiento de los discos. El programa comprobará si el movimiento se puede realizar e indicará al usuario si es posible o no realizarlo. Si el movimiento es legal, se realizarán los cambios en las torres indicadas y se mostrará el estado actual de las tres torres. Si el movimiento es ilegal se lo indicará al usuario e indicará porque es un movimiento ilegal ('la torre no contiene discos', 'imposible colocar un disco menor sobre otro mayor', ...).

El programa acabará cuando todos los discos de la primera torre hayan pasado a la tercera torre, o cuando el usuario introduzca un valor de la torre imposible (distinto de uno, dos o tres).

Al finalizar el juego, el programa indicará el número de movimientos realizados por el usuario y el mínimo número de movimientos necesarios para realizar el cambio ($2^n - 1$).

4. ENTREGA DE PROGRAMAS

Al comienzo de la siguiente sesión de prácticas se entregarán al profesor tres ficheros:

- 1) Ficheros de la clase **Pila** (**##Pila.cpp** y **##Pila.h**)
- 2) Programa principal del juego (**##hanoi.cpp**).

Nota Muy Importante

Antes de poder empezar a realizar cualquiera de las prácticas **es necesario** presentar las hojas de especificación de programas (documentación de programas) con las tareas que se van a realizar en la práctica, explicando brevemente como se van a solucionarse los problemas que se plantean.

ENTREGA DE PROGRAMAS:

Al comenzar la sesión 4 de prácticas (5, 20, 21, 29 y 30 de abril)