

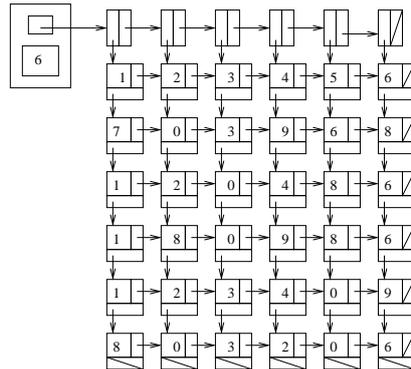


Alumno: []

Grupo: []

1. Sea una matriz cuadrada representada mediante listas dinámicas:

(1 2 3 4 5 6
7 0 3 9 6 8
1 2 0 4 8 6
1 8 0 9 8 6
1 2 3 4 0 9
8 0 3 2 0 6)



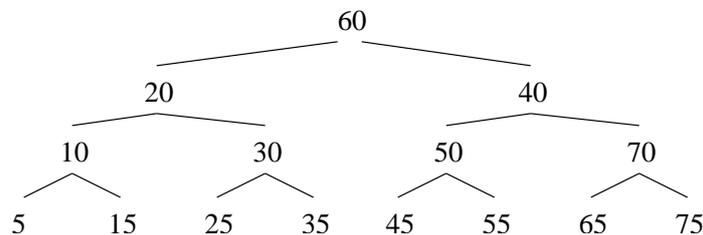
Definir la estructura adecuada para ese tipo de datos e implementar las siguientes operaciones de transformación de matrices:

x.x pts.

- a) Invertir sobre el eje de las columnas: obtener una matriz de la misma dimensión donde la primera columna de la original sea la última, la última la primera y las demás columnas también tengan el orden invertido
b) Invertir sobre el eje de las filas: la idea es similar a la del apartado anterior, pero relativo a filas.
c) Traspuesta: obtener una matriz de la misma dimensión cambiando filas por columnas.
d) Adjunta a una posición: obtener la matriz adjunta a una posición (fila, columna). La adjunta se obtiene eliminando la fila y la columna que contienen esa posición.

2. Codificar una función recursiva que tome como parámetros un array con una serie ordenada de elementos y el número de elementos válidos del array y genere un árbol binario de búsqueda completo. Por ejemplo, si el contenido del array es 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, el árbol resultante es

x.x pts.



3. Construir la clase array asociativo (CArrAsoc). Un array asociativo no utiliza números enteros como índice sino cadenas de caracteres. Además de esta diferencia, a los arrays asociativos se les puede añadir o quitar elementos. Siempre se accede a ellos a través del índice que aquí se denomina clave. De esta forma pueden verse como una tabla o diccionario flexible. La implementación habrá de hacerse sobre la clase Lista posicional estudiada en clase. La interfaz sería:

```
INTERFAZ CLASE CArrAsoc
  Tipo TD
  CADENA S
METODOS
  Asignar(E S Clave, E TD v)
  // si ya existia Clave,
  // cambia su valor, si no, la crea
  Eliminar(E S Clave, E TD v)
  // elimina el par
  N NElem()
  // total de elementos en el array
  S ClaveI(E N i)
  // devuelve el valor de la clave
  // i (i debe ser <= NElem)
FIN CArrAsoc
```

```
INTERFAZ CLASE CListaPos
TIPOS
  TBaseL ...
METODOS
  Insertar(E N pos, E TBaseL x)
  Reescribir(E N pos, E TBaseL x)
  Eliminar(E N pos)
  TBaseL Consultar(E N pos)
  N Longitud()
FIN CListaPos
```