

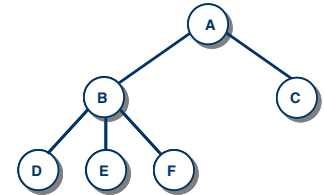
GUIÓN DE PRÁCTICAS 2

APLICACIÓN DE ESTRATEGIAS DE BÚSQUEDA NO INFORMADA AL PROBLEMA N-PUZLE

En la práctica 1 se ha realizado la implementación en C de la formulación del problema del N puzzle y se han obtenido los archivos `puzzle.c` y `puzzle.h`

Este código fuente se integrará en un proyecto junto con el código de las estrategias de búsqueda no informada para obtener la resolución completa del problema. Los archivos que se os facilitan son:

`busquedaAlum.c`, `busquedaAlum.h`, `listaia.c`, `listaia.h`, `nodo.h`



Nodo.h

1. Al utilizar las distintas estrategias de búsqueda, los estados de cada problema se incluyen dentro de una estructura de datos que serán los nodos. Identifica esta estructura de datos en la implementación dada y determina para qué sirve cada uno de los campos. Razona por qué no se usa directamente el tipo `tEstado`.

Búsqueda.h

2. Dado el código en C del algoritmo de búsqueda:
Crea un proyecto nuevo que incluya todos los archivos facilitados en `búsqueda.zip`, y los archivos correspondientes a la formulación del puzzle de la práctica 1.
 - a. Prueba el programa llamando a la función `búsqueda` desde un `main.c`
 - b. Crea el árbol de búsqueda que se derive de la estrategia implementada para el siguiente estado inicial (añade las instrucciones de visualización para estudiar la ejecución paso a paso):

	1
3	2

Estado inicial

1	2
	3

Estado final

- c. Prueba el programa con distintos estados iniciales y determina la estrategia que se encuentra implementada, Anchura o Profundidad.
- d. Haz los cambios necesarios para obtener la estrategia alternativa: Anchura o Profundidad.
- e. Estudia detenidamente el código para
 - i. Añadir un contador del número de estados visitados y número máximo de nodos en la lista de abiertos
 - ii. Añadir el control de los estados repetidos: dónde incluir el control, qué función se usaría, qué estructuras de datos, etc.

GUIÓN DE PRÁCTICAS 2

Ejercicios de ampliación

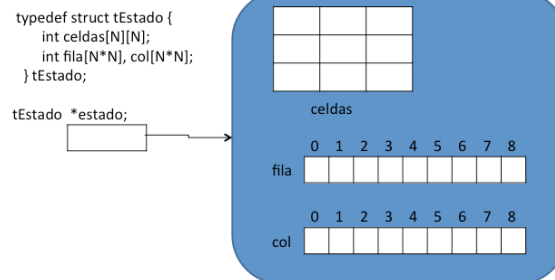
1. Dado el pseudocódigo de la búsqueda genérica vista en el tema 2, realiza las modificaciones necesarias para obtener el algoritmo de Búsqueda en Anchura y en Profundidad.

Pseudocódigo de la Búsqueda Genérica (transparencias Tema 2 de teoría)

```
solución o fallo función Búsqueda (problema, estrategia)
inicio
  Abiertos ← {Nodo Inicial}
  Vacía ← Falso
  Objetivo ← Falso
  mientras (No Vacía) y (No Objetivo)
    Actual ← selecciona nodo de Abiertos
    si EsObjetivo(Actual) entonces
      Objetivo ← Verdadero
    si_no
      expande el nodo y
      añade los nuevos nodos generados a la lista de Abiertos
      de acuerdo a estrategia
    fin_si
  Vacía ← EstaVacía(Abiertos)
fin_mientras
si Objetivo entonces devolver Camino a la Solución
si_no devolver Fallo
fin_función
```

2. Las listas Abiertos, Sucesores y Cerrados son las principales listas de nodos que se utilizan. Identifica su tipo de datos dentro del código y realiza un esquema gráfico que te ayude a entender cómo son y qué contienen.

Ejemplo con el tipo de datos tEstado:



3. ¿Qué ventajas y desventajas tiene el uso de estructuras de datos con memoria estática para resolver este problema?